HOWTO Terminal Texte pour Linux

David S. Lawyer `mailto:bf347@lafn.org`, traduit par Julien Garnault `mailto:judge@club-internet.fr`
v1.09, 28 Novembre 1999

Ce document explique ce que sont les terminaux en mode texte, comment ils fonctionnent, comment les installer et les configurer, et fournit des informations sur la manière de les réparer. Cela peut être utile même si vous n'avez pas le manuel du terminal. Bien qu'il soit écrit pour de vrais terminaux reliés à un système Linux, certaines informations de ce manuel sont applicables à l'émulation de terminal et peuvent être utiles pour des systèmes différents de Linux.

Contents

1 Introduction

Pour tenter d'installer un terminal rapidement, voyez 3 (installation rapide).

1.1 Copyright, marques déposées, avertissement et crédits

1.1.1 Copyright

1. envoyez le travail dérivé (dans le format le plus adapté, en sgml par exemple) au LDP (Linux Documentation Project) ou un equivalent pour poster sur internet. Si ce n'est pas le LDP, dites au LDP ou il est trouvable. A part pour une traduction, envoyez une copie a l'url precedente du mainteneur donnée dans la derniere version.
2. donniez une licence au travail dans l'esprit de cette licence, ou que vous utilisiez la GPL. Incluez une notice concernant le copyright et au moin un pointeur vers la licence utilisée.
3. donniez un crédit total aux précédent auteurs et contributeurs majeurs.

Si vous considerez faire un travail dérivé autre qu'une traduction, il est requis d'en discuter avec le mainteneur courant.

1.1.2 Avertissement

Bien que je n'ai pas tenté de vous induire sciemment en erreur, il y a sûrement un certain nombre d'erreurs dans ce document. Veuillez me les signaler. Puisque c'est une documentation libre, il devrait être évident que je ne peux pas être tenu légalement responsable des erreurs commises.

1.1.3 Marques déposées

Si certains mots représentent des marques déposées, le contexte devrait indiquer clairement à qui elles appartiennent. Par exemple, "MS Windows NT" implique que "Windows NT" appartient à Microsoft (MS). Mac est fabriqué par Apple Computer. Les marques déposées appartiennent à leurs propriétaires respectifs.

1.1.4 Crédits

La majeure partie de la section "Connexion physique" provient du Howto Serial v. 1.11 de Greg Hankins (avec sa permission). Sa partie : "Comment configurer un terminal connecté à mon PC ?" a été incorporée dans la version 1.00 en des endroits variés. v1.09 a 25 changements (et corrections d'erreurs) suggérés par Alessandro Rubini qui a passé en revu ce HOWTO.

Pour la traduction en français, j'aimerais remercier Xavier Glattard, qui a entrepris le début de la traduction et que j'ai reprise en cours ; ainsi que Guillaume Allègre et Jean-Luc Cassel qui ont fait un énorme travail de relecture.

Pour ma part (Julien Garnault) Je remercie Olivier Tharan qui a fait du très bon travail. Ce qui m'a permis de reprendre ce HOWTO sans trop de problemes.

1.2 Plans pour l'avenir : vous pouvez m'aider

Veuillez me signaler toute erreur dans les faits, les opinions, la logique, l'orthographe, la grammaire, la clarté, les liens, etc. Mais d'abord, si la date est vieille de plus de quelques mois, vérifiez que vous avez la dernière version. Veuillez m'envoyer toutes les informations que vous pensez être pertinentes pour ce document. [NdT : cela s'applique aussi pour la version française !]

À partir de la version 1.00, j'ai tenté pour la première fois d'aider les gens à configurer des terminaux sans recourir à un manuel de terminal. Il en faudrait bien plus à cet égard. Une manière de résoudre ce problème serait que les fabricants de terminaux mettent leurs manuels sur Internet. Je vous suggère de les encourager à le faire. Fournir des informations sur la configuration de la plupart des terminaux dans ce HOWTO est une tâche décourageante. Il y a beaucoup de terminaux différents, mais il y a bien moins de modèles qu'il n'y en avait dans les années 1980, la tâche n'est donc pas totalement impossible à réaliser.

Veuillez m'envoyer tous les manuels de terminaux que vous pourriez avoir en trop, surtout sur les terminaux fabriqués dans les dix dernières années (mais j'accepterai aussi les manuels plus anciens). De plus, vous pourriez écrire quelque chose sur un certain type de terminal pour l'annexe D : Notes par nom de marque.

1.3 Nouvelles versions de ce HOWTO

Les nouvelles versions du HOWTO seront disponibles à la navigation Internet et au téléchargement sur les sites miroirs du LDP (Linux Documentation Project). Voyez `http://metalab.unc.edu/LDP/mirrors.html` pour avoir une liste de ces miroirs. Divers formats sont disponibles. Si vous voulez chercher rapidement la date de la dernière version, allez à `http://metalab.unc.edu/LDP/HOWTO/Text-Terminal-HOWTO.html` . (NdT : le miroir officiel en France est `http://www.freenix.org/unix/linux/HOWTO/Text-Terminal-HOWTO.html` .)

1.4 HOWTOs connexes

- Le HOWTO Serial possède des informations sur les cartes série multiports utilisées à la fois pour les terminaux et les racks de modems. Il possède des informations techniques sur le port série, notamment pour le dépanner.
- Le HOWTO Modem
- Le HOWTO Serial-Programming
- Le HOWTO Xterminal (non maintenu). Il se trouve à `http://metalab.unc.edu/pub/Linux/docs/HOWTO/unmaintained/mini/Xterminal`

1.5 Terminologie utilisée dans ce document

Configuration veut dire la même chose que mise en place (NdT : j'ai utilisé le mot "configuration" tout au long du document). Alors que les commandes Linux possèdent des options (avec le symbole -), les options dans un sens plus vaste sont d'autres types de choix. L'installation au sens large comprend la configuration du matériel et du logiciel. Une affirmation que je pense être vraie (mais qui peut ne pas l'être) se termine par deux points d'interrogation : ?? Si vous êtes sûr de l'affirmation, dites-le moi.

1.6 Qu'est-ce qu'un terminal ?

Un terminal consiste en un écran et un clavier qu'on utilise pour communiquer à distance avec un ordinateur (hôte). On l'utilise comme s'il s'agissait d'un ordinateur personnel mais le terminal est éloigné de l'ordinateur hôte (à l'autre bout de la pièce ou même à l'autre bout du monde). Les programmes s'exécutent sur l'ordinateur hôte mais les résultats s'affichent sur l'écran du terminal. Sa capacité de calcul est relativement faible (sinon ce serait un ordinateur et non un terminal). Cette capacité de calcul est en général limitée à la capacité d'afficher ce qu'on lui envoie (il est possible que ceci comprenne des graphiques plein écran) et la capacité d'envoyer à l'hôte ce qui est tapé au clavier.

Dans les temps éloignés des gros ordinateurs, du milieu des années 1970 au milieu des années 1980, la plupart des gens utilisaient des terminaux pour communiquer avec les ordinateurs. Ils y tapaient des programmes, les faisaient tourner, écrivaient des documents, envoyaient des commandes d'impression, etc. Un câble reliait le terminal à l'ordinateur (souvent de manière indirecte). On l'appelait terminal puisqu'il était situé à une extrémité terminale de ce câble.

Si vous utilisez Linux (sauf pour une utilisation avec X Window) avec un moniteur et un clavier vous savez déjà ce qu'est un terminal parce que vous en utilisez un (ou plus précisément un "terminal virtuel"). Le moniteur (avec le clavier) s'appelle console, mais il émule un terminal. Dans X Window : xterm, rxvt et zterm émulent des terminaux.

Un vrai terminal est différent d'un moniteur parce que c'est un montage électronique différent. Un terminal texte est souvent relié au port série de l'ordinateur par l'intermédiaire d'un long câble. Ainsi, contrairement à un moniteur qui est le plus souvent situé juste à côté de l'ordinateur, un terminal peut se situer à une distance assez grande de son ordinateur hôte. La carte vidéo à l'intérieur d'un ordinateur stocke l'image vidéo qui apparaît sur l'écran du moniteur. Pour un terminal, l'équivalent de cette carte vidéo est construite à l'intérieur même du terminal mais puisque les terminaux texte sont souvent monochromes sans beaucoup de graphiques, les capacités de cette "carte vidéo" sont plutôt faibles. De plus, la plupart des terminaux texte n'ont pas de souris.

Dans la terminologie client-serveur en réseau, on pourrait penser que le terminal est le client et l'ordinateur hôte le serveur. Certains ont appelé le terminal un "client léger". Mais ce n'est pas vraiment un "client" et l'hôte n'est pas vraiment un "serveur". Le seul "service" fourni par l'hôte est de recevoir chaque lettre tapée au clavier et de réagir à ceci comme un ordinateur le ferait. Le terminal ressemble à une fenêtre donnant sur l'ordinateur comme l'est un moniteur (et son clavier). Vous avez sûrement déjà utilisé des terminaux virtuels dans Linux (en pressant Alt(gauche)-F2, etc.). Un vrai terminal est la même chose que lancer un terminal virtuel mais vous le lancez sur l'écran du terminal plutôt que de partager le moniteur. Par rapport à l'utilisation d'un terminal virtuel sur la console (moniteur), ceci permet à une autre personne de s'asseoir devant le terminal réel et d'utiliser l'ordinateur en même temps que d'autres personnes.

2 Types de terminaux

2.1 Terminaux passifs

Il y a plusieurs définitions contradictoires pour un "terminal passif" mais à mesure que le temps passe, de plus en plus de terminaux sont appelés passifs. Ce document couvre principalement les terminaux texte qui n'affichent que du texte à l'écran. On pourrait l'appeler "HOWTO Terminaux Passifs" mais dans certains articles de magazine, tout terminal, quelle que soit son intelligence, même ceux qui présentent une interface graphique complète (GUI), sont appelés passifs. Si tous les terminaux sont "passifs" il n'y a plus de raison d'ajouter le mot "passif" derrière le mot terminal (sauf dans un baratin commercial pour vendre des ordinateurs ou similaires en tant que terminaux "intelligents"). À cause de la signification ambiguë de "terminal passif", ce n'est pas considéré ici comme un type de terminal.

2.2 Terminaux texte

Pour un terminal texte, un flux d'informations à double sens entre l'ordinateur et le terminal se forme sur le câble les reliant tous les deux. Ce flux est constitué d'octets ASCII où chaque octet représente généralement un caractère. Les octets tapés au clavier vont vers l'ordinateur et la plupart des octets venant de l'ordinateur sont affichés sur l'écran du terminal. Des octets spéciaux (ou des séquences d'octets) de l'ordinateur indiquent au terminal où déplacer le curseur, ce qu'il faut effacer, où démarrer et arrêter le soulignement et/ou le clignotement et/ou le gras, etc. Il y a souvent des centaines de commandes spéciales et beaucoup de terminaux peuvent même changer leur police.

La communication utilise des caractères (lettres) encodées avec un tableau de codes correspondant au jeu de caractères en cours d'utilisation. En général, les 128 premiers octets sur les 256 octets possibles utilisent les codes ASCII. Les terminaux pour des systèmes de type Unix sont normalement connectés aux ordinateurs par un câble qui se déroule entre les ports série asynchrones (RS-232-C = EIA-232-D) de l'ordinateur hôte et du terminal. Quelquefois la connexion se fait par modem ou grâce à un serveur de terminaux, etc.

D'autres noms pour les terminaux texte sont "terminal série", "terminal à cellule de caractère", "terminal ASCII/ANSI", "terminal asynchrone", "terminal de données", "terminal vidéo" et "terminal d'affichage vidéo" (VDT). Dans les jours anciens, "unité d'affichage vidéo" (VDU) était utilisé pour les terminaux mais en toute rigueur, ceci exclut le clavier.

Le "mode par blocs" était exclusivement utilisé par les vieux terminaux des minis IBM mais beaucoup de terminaux modernes possèdent aussi cette capacité (qui n'est pas beaucoup utilisée). Les caractères que vous tapez sont retenus temporairement dans la mémoire du terminal (et peuvent parfois être édités grâce à un éditeur résidant dans le terminal). Alors quand la touche envoi (ou autre) est pressée, un bloc de caractères (parfois juste une ligne de caractères) est envoyé tout d'un coup à l'ordinateur. Le mode par blocs (à la fin 1998) n'est pas supporté par Linux. Voyez la section 21.7 (mode par blocs).

2.3 Terminaux graphiques

Jusqu'à un certain degré certains symboles ASCII peuvent fournir du graphisme aux terminaux texte. On peut faire des flèches <— et dessiner des boîtes avec _ et |. Avec des ensembles spéciaux de caractères graphiques, on peut en faire encore plus. Aucun de ceux-ci ne sont vraiment des terminaux graphiques. Cependant, le terme "terminal graphique" est quelquefois donné à tous les terminaux texte puisque le texte est une forme limitée de graphique.

Il y a deux types de base pour l'affichage graphique : rastérisé et vectorisé (rarement utilisé). Les graphiques rastérisés (en mode point par point) affichent des points sur l'écran sur des lignes de balayage horizontal par l'intermédiaire d'un faisceau d'électrons (ou en activant des pixels ou points sur un écran plat). Les affichages à graphisme vectoriel sont généralement destinés à des écrans monochromes qui n'ont pas de points. Ils utilisent une électronique intelligente pour tracer des lignes et des courbes avec un faisceau d'électrons qui peut se déplacer dans n'importe quelle direction (à tout endroit et avec tout angle). Les graphiques vectoriels affichent des lignes de grande qualité sans zigzags mais sont à la fois rares et chers. Les graphiques rastérisés sont utilisés de manière quasi-universelle de nos jours. Pour les PC, les images codées en format graphique vectoriel sont quelquefois utilisées mais sont traduites en format graphique rastérisé pour l'affichage (avec une baisse de qualité pour l'image).

2.3.1 Terminaux graphiques sur ligne série

La plus grande partie de ce document s'applique aussi à ceux-ci. La plupart de ceux-ci peuvent aussi fonctionner comme des terminaux texte. Les protocoles pour de tels terminaux graphiques sont : Tektronix Vector Graphics, ReGIS (DEC), Sixel (DEC) et NAPLPS (North American Presentation Level Protocol Syntax, syntaxe de protocole au niveau présentation d'Amérique du Nord).

2.3.2 Terminaux graphiques rapides (d'autres noms leur sont souvent donnés)

Ceux-ci ne sont pas couverts par ce document. Un terminal qui mérite qu'on l'appelle intelligent est un terminal graphique qui peut afficher rapidement des graphiques plein écran comme un moniteur de PC. Il aura aussi une souris. Les octets qu'on lui envoie représentent souvent des bits pour des images (et souvent des graphiques). Il utilisera souvent une connexion à grande vitesse vers son ordinateur hôte en utilisant de la paire torsadée ou un câble coaxial. Les terminaux X Window font partie de ces modèles. Voyez le lien vers le HOWTO XTerminal référencé en 1.4 (HOWTOs connexes)

Pour afficher une interface graphique MS-Windows il y a plusieurs types d'interfaces et de terminaux : WinTerm en est un, il utilise le logiciel WinFrame de Citrix. Un autre est Hydra de Microsoft (nom de code), basé en partie sur le code de Citrix, aussi connu comme "Serveur de Terminal Windows" qui fonctionne avec la version 4 ou supérieure de Windows NT. Citrix utilise son protocole ICA et a créé un supplément à Hydra nommé pICAsso pour que les terminaux WinFrame (ICA) puissent utiliser le système ICA. Hydra est aussi multi-utilisateurs. Il y a aussi le "Terminal Personnel Multiconsole" de Unbounded Technologies et Tektronix possédait son interface multi-utilisateurs mais supportera désormais Hydra. Un article de magazine en 1997 a appelé Winterm un "terminal passif" mais il est vraiment intelligent. De tels terminaux sont souvent appelés "clients légers", mais certains clients légers sont plus que de simples terminaux puisqu'on peut leur faire exécuter du code Java, etc.

2.4 Network Computers (NC, ordinateurs de réseau)

Ce ne sont ni de vrais ordinateurs ni de vrais terminaux mais quelque chose entre les deux. Un type de Network Computers (NC) est un ordinateur avec un processeur mais pas de disque dur. Il est entièrement graphique et se connecte à un ordinateur serveur. Il est différent d'un terminal puisque le programme qu'il fait tourner s'exécute sur son propre processeur. Du code Java peut lui être envoyé pour être exécuté. IBM appelle ceci une "station réseau" (Network Station). Ils devraient fonctionner sur des réseaux IP et pourraient fonctionner avec un serveur sous Linux. Wintel a fabriqué un "NetPC" qui, à la différence du NC, est presque un ordinateur PC. Cependant, il n'a pas de disque amovible et les utilisateurs ne peuvent pas installer leur propre logiciel ou obtenir des copies de quoi que ce soit.

Bien que les promoteurs des NCs et des terminaux graphiques similaires aient prévu de remplacer des millions de PCs, cela n'est pas encore arrivé. L'une des raisons principales est que les PCs ont vu leurs prix chuter dans les dernières années et sont souvent moins chers que les NCs. Ainsi pour les terminaux, les terminaux texte sont encore prédominants.

2.5 Émulation sur un PC

Puisqu'un PC possède un écran et un clavier (comme un terminal) mais possède aussi plus de puissance de calcul, il est facile d'utiliser une partie de cette puissance de calcul pour que le PC se comporte comme un terminal texte. C'est ce que l'on appelle "l'émulation de terminal". Ils émulent en général des terminaux texte. Voyez 9 (émulation de terminal)

3 Installation rapide

Voici une procédure rapide pour installer un terminal sans passer par une procédure de 12 (mise en place) à la fois pour le terminal et l'ordinateur hôte. Cela ne fonctionnera probablement pas bien s'il se trouve que le terminal a été configuré de manière incompatible avec l'ordinateur. Si vous ne comprenez pas tout ceci vous devrez consulter d'autres parties de ce document pour plus d'informations.

Pour installer un terminal, regardez d'abord dans `/etc/termcap` ou `terminfo.src` pour y trouver une entrée le concernant (voir 15 (terminfo et termcap (détaillé))). Déterminez sur quel port série vous le brancherez et quelle est le nom tty pour ce port (par exemple, `ttyS1`, voyez 5.1 (noms de périphériques)). En tant qu'utilisateur root, éditez `/etc/inittab` et ajoutez une commande getty à côté des autres commandes getty. Le format de la commande getty dépend du programme getty que vous utilisez. `agetty` (simplement appelé `getty` dans la distribution Debian) est le plus simple (pas de fichier de configuration). Voyez le fichier "`info`" ou la page de manuel de getty. Pour les paramètres de getty, utilisez le nom terminfo (ou termcap) de votre terminal, comme vt100. Entrez une vitesse de transmission supportée par le terminal. Si vous mettez la vitesse trop haut vous aurez peut-être besoin d'utiliser le 10 (contrôle de flux).

Connectez alors physiquement le port série principal du terminal au port série choisi de l'ordinateur avec un câble null-modem et allumez le terminal. N'espérez pas que la plupart des câbles tout prêts soient câblés correctement pour gérer le contrôle de flux matériel. Assurez-vous que la vitesse de transmission du terminal est la même que celle que vous avez donnée à getty et que son paramètre "bits de données" est 8. Alors, sur la console de l'ordinateur tapez "init q" pour faire prendre en compte les changements que vous avez faits au fichier inittab. Vous devriez maintenant voir une invite de login sur le terminal. Sinon, appuyez sur la touche retour chariot du terminal. Si cela ne fonctionne pas, continuez de lire ce document et/ou voyez 17 (régler les problèmes).

4 Pourquoi utiliser un terminal ?

4.1 Introduction sur le fait d'utiliser un terminal

Les PC sont de nos jours si puissants qu'un seul PC peut souvent supporter plusieurs personnes à la fois en train de l'utiliser, surtout s'ils exécutent des tâches faibles en charge telles que l'édition de texte, l'entrée de données, etc. Une manière de faire ceci est de relier un certain nombre de terminaux à un seul PC (ordinateur hôte) par l'intermédiaire de modems ou de connexions directes par câble. Pour ce faire, il est mieux d'avoir un système d'exploitation multi-utilisateurs tel que Linux, comma ca chaque utilisateur sur un terminal peut utiliser l'ordinateur independemment. On a appelé cela le "partage de temps" (time sharing) mais ce n'est pas une terminologie correcte de nos jours puisque l'informatique "distribuée" sur un réseau est aussi une sorte de partage de temps. On pourrait mieux le décrire comme de l'informatique "centralisée". Mais l'ordinateur central peut être connecté au reste du monde par un réseau afin que les utilisateurs des terminaux puissent envoyer du courrier électronique, naviguer sur Internet avec le navigateur "lynx", etc. Ce n'est donc pas vraiment "centralisé" non plus.

On a rarement utilisé des terminaux avec des PC parce que les systèmes d'exploitation populaires qui les ont utilisés (Windows, DOS et Mac) n'ont pas été multi-utilisateurs jusqu'en 1998 (c'est le cas pour MS Windows NT) et ne pouvaient pas auparavant tres bien accepter de terminaux. Maintenant que Linux, système d'exploitation multi-utilisateurs, est disponible pour les PC, l'utilisation de terminaux avec des PC devient envisageable. L'inconvénient est que les terminaux texte ne sont pas assez perfectionnés pour supporter le type d'interface graphique utilisateur (GUI) que bien des utilisateurs d'ordinateur attendent de nos jours.

4.2 Une réduction du coût du matériel ?

Quand les ordinateurs (même les PCs) étaient relativement chers, l'utilisation de terminaux tirait les coûts du matériel vers le bas de manière significative. Maintenant, avec les PCs bon marché, les économies sur les coûts sont un problème. Voici ce que j'ai écrit il y a des années quand les PCs étaient plus chers. C'est encore vrai maintenant, mais d'une moindre mesure.

Si plusieurs personnes utilisent le même ordinateur en même temps, il y a une diminution de la quantité de matériel nécessaire pour le même niveau de service. Une façon de faire des économies est due au partage du code. Les fichiers des applications sur les disques durs sont partagés, de même que les bibliothèques partagées en mémoire (même si les gens font tourner des programmes différents, à condition qu'ils utilisent certaines fonctions identiques dans leur code). Une autre façon d'économiser est due à la réduction du pic de charge. Le matériel d'un seul PC peut être inactif la plupart du temps pendant que les gens entrent les informations lentement, réfléchissent, discutent ou s'éloignent de leur bureau. Avoir plusieurs personnes sur le même ordinateur à la fois fait bon usage d'une bonne partie de ce temps d'inactivité qui, sinon, serait gâché.

Ces économies sont substantielles. On peut estimer grossièrement (en utilisant la théorie des statistiques) que pour neuf personnes (huit terminaux et une console), le PC partagé n'a besoin que d'environ trois fois plus de capacité (en mémoire, espace disque, puissance processeur, etc.) qu'un PC unique afin de fournir le même niveau de service par personne. Le coût par utilisateur du matériel de calcul pour un tel système partagé devrait donc être trois fois moindre. Cependant, le coût du système d'affichage (écrans, claviers, électronique vidéo, etc.) est à peu près le même dans les deux cas. Les terminaux induisent cependant un surcoût : l'équipement de l'ordinateur hôte avec des ports série supplémentaires.

Pour faire une comparaison honnête avec les PC, les terminaux devraient avoir les mêmes capacités que les moniteurs de PC. Malheureusement, les terminaux graphiques couleur pour Linux (X Window) avec des communications à grande vitesse est un marché de niche avec des prix élevés, et par conséquent dans ce cas les économies en coût de matériel risquent d'être inexistantes. Pour les terminaux texte, par contre, on fera des économies, surtout si on obtient les terminaux à bas prix.

4.3 Contrôle des logiciels

En informatique centralisée, les logiciels (et les mises à jour des logiciels) ne doivent être installés que sur un seul ordinateur hôte au lieu de plusieurs. La personne qui s'occupe de cet ordinateur peut contrôler les logiciels qui y sont installés. Ceci est avantageux si la personne qui contrôle l'ordinateur hôte fait du bon travail et connaît les besoins et préférences des autres utilisateurs. On peut empêcher les utilisateurs de jouer à des jeux ou de naviguer sur Internet en n'installant pas les logiciels (ou bien en en restreignant l'accès). Que le contrôle centralisé soit désirable ou non dépend de chaque situation.

4.4 Mises à jour du matériel

Avec les terminaux, les mises à jour du matériel de l'ordinateur ne prennent place que sur un ordinateur au lieu de plusieurs. Ceci économise un effort d'installation. Alors que le coût du matériel pour la mise à jour de l'ordinateur hôte sera plus important que pour un PC simple (puisque l'hôte a besoin de plus de puissance de calcul qu'un PC), le coût sera bien moindre que de mettre à jour le materiel d'un nombre de PC qu'on utiliserait à la place des terminaux.

4.5 Autres avantages des terminaux

- L'élimination du bruit des ventilateurs et des disques durs, à condition que les terminaux ne soient pas proches de l'ordinateur.
- Les utilisateurs des terminaux peuvent partager des données et des fichiers et peuvent s'envoyer du courrier électronique. C'est la même chose qu'un réseau local.

4.6 Désavantages majeurs des terminaux

- Les terminaux texte ne disposent pas d'affichage graphique rapide (ou de graphiques haute résolution) bien qu'ils utilisent souvent des jeux de caractères graphiques pour dessiner des boîtes, etc. Ce manque limite les logiciels qu'on peut utiliser dessus.
- Si l'ordinateur hôte s'arrête, plus personne ne peut utiliser les terminaux non plus (sauf si on peut se connecter sur un ordinateur hôte qui attend cela).

4.7 Les terminaux texte sont-ils obsolètes ?

Les terminaux texte représentent une technologie obsolète parce que pour un matériel légèrement plus cher, on pourrait construire un terminal intelligent (avec la même qualité d'affichage). Ceci n'a pas toujours été le cas puisque aux alentours de 1980 la mémoire coûtait des milliers de francs par méga-octet. Maintenant avec de la mémoire et des processeurs à bas prix, on pourrait faire un terminal texte intelligent pour une augmentation du coût du matériel de seulement 10 à 20 %.

Les raisons pour lesquelles les terminaux texte ne sont pas encore obsolètes sont :

- Il n'y a pas d'interface normalisée satisfaisante pour des terminaux graphiques intelligents. Le système MS Hydra fonctionne avec Windows NT, alors que X Window n'est pas aussi efficace qu'il devrait être (et les terminaux X Window sont trop chers).
- Beaucoup de gens n'ont pas besoin de graphiques plein écran.
- Les terminaux texte sont peu coûteux et mettent en théorie plus de temps à devenir obsolètes, mais peuvent cependant donner accès à un ordinateur bien plus récent (et plus puissant).
- Puisque le fonctionnement d'un terminal texte (en opposition à un terminal entièrement graphique) ne consomme pas beaucoup de ressources sur un PC moderne, on peut faire tourner beaucoup de terminaux sur un seul PC de manière efficace.

5 Vue d'ensemble du fonctionnement des terminaux (sous Linux)

Voir aussi la section 7 (quelques détails sur le fonctionnement des terminaux).

5.1 Noms de périphériques

Chaque terminal est relié à un port série sur l'ordinateur hôte (souvent un simple PC). Les ports ont les noms suivants : ttyS0, ttyS1, ttyS2, etc. Ils sont représentés par des fichiers spéciaux dans le répertoire /dev (device : périphérique). /dev/ttyS0 correspond au COM1 sous DOS ou Windows. ttyS1 est le COM2, etc. Voyez 6 (fichiers spéciaux pour les terminaux) pour plus de détails sur ceux-ci et les "périphériques" connexes tels que cua.

5.2 Se logger / se délogger

Quand l'ordinateur hôte démarre il lance le programme getty (voyez le HOWTO Serial 4.1 et 7.2) sur chaque port série qui y possède un terminal (ainsi qu'il est spécifié dans le fichier /etc/inittab). Le programme getty lance le programme "login" pour que les gens puissent se logger. Une invite "login:" apparaît à l'écran. Les gens sur le terminal se loggent (après avoir donné leur mot de passe) et ont alors accès à l'ordinateur. Quand il est temps d'éteindre le terminal, on se délogge en général et on éteint le terminal. Voyez 14.8 (restrictions sur le login) à propos de la restriction sur les logins (avec la permission pour l'utilisateur root de se logger sur un terminal).

5.3 Half/Full Duplex

Si on regarde quelqu'un taper sur un terminal, les lettres tapées apparaissent simultanément sur l'écran. Une personne naïve pourrait penser que ce qu'on tape est envoyé directement du clavier à l'écran avec une copie dirigée vers l'ordinateur (de manière half-duplex, voir le paragraphe suivant). Ce qui se passe généralement est que ce qui est tapé au clavier est envoyé directement à l'ordinateur hôte uniquement qui en retour renvoie au terminal chaque caractère qu'il reçoit (ce qu'on appelle full-duplex). Dans certains cas (comme les mots de passe ou des commandes brutes d'éditeurs) les lettres tapées ne sont pas renvoyées.

Full-duplex veut dire qu'il y a deux liens de communications unidirectionnels. Le full-duplex est la norme de fait sur les terminaux. Le half-duplex est la moitié d'un duplex, ce qui veut dire qu'il n'y a qu'un lien de communication unidirectionnel. Ce lien doit être partagé par les communications allant dans les deux directions et on ne peut utiliser qu'une direction à la fois. Dans ce cas l'ordinateur n'est pas en mesure de répéter les caractères que vous tapez (et que vous lui envoyez) et par conséquent le terminal doit aussi envoyer directement sur son écran chaque caractère que vous tapez. Certains terminaux possèdent un mode de fonctionnement half-duplex qui est rarement utilisé.

5.4 Mémoire du terminal

L'image sur un tube à électrons s'évanouira presque instantanément à moins qu'elle ne soit réaffichée fréquemment sur l'écran par un faisceau d'électrons lancés sur la face du tube. Puisque que le texte envoyé à un terminal doit rester sur l'écran, l'image de l'écran doit être stockée dans les puces mémoire du terminal et le faisceau d'électrons doit balayer l'écran de façon répétée (disons 60 fois par seconde) pour maintenir l'image. Voyez 7.1 (mémoire du terminal) pour plus de détails.

5.5 Commandes pour le terminal

Le terminal est sous le contrôle de l'ordinateur. L'ordinateur envoie au terminal non seulement du texte pour afficher ce dernier sur l'écran mais aussi des commandes que le terminal exécute. Ce sont des 7.3.1 (codes de contrôle) (octets) et des 7.3.2 (séquences d'échappement). Par exemple, le code de contrôle CR (retour chariot) déplace le curseur sur le côté gauche de l'écran. Une certaine séquence d'échappement (plusieurs octets dans lesquels le premier octet est le code de contrôle d'"échappement") peut déplacer le curseur à l'emplacement de l'écran spécifié par les paramètres placés dans la séquence d'échappement.

Les 7.2 (premiers terminaux) n'avaient que peu de telles commandes mais les terminaux modernes en ont des centaines. L'apparence de l'affichage peut être modifiée à certains endroits : fort, faible, souligné, clignotant et vidéo inverse. Un haut-parleur dans un terminal peut émettre un "cliquetis" quand une touche est pressée ou émettre un bip si une erreur s'est produite. Les touches de fonctions peuvent être programmées pour des utilisations spéciales. Des polices variées peuvent exister. On peut faire défiler l'affichage vers le haut ou vers le bas. On peut effacer des parties spécifiques de l'affichage. On peut utiliser divers types de contrôle de flux pour arrêter le flux de données quand les octets sont envoyés au terminal plus rapidement que le terminal ne peut le supporter. Il y en a bien plus, que vous découvrirez en parcourant une notice de terminal très technique ou à travers les liens Internet de la 20.1 (liste des séquences d'échappement).

5.6 Manque de normalisation résolu par Terminfo

Alors que les terminaux faits pour les États-Unis utilisaient tous le même code ASCII pour l'alphabet (sauf les terminaux IBM qui utilisaient EBCDIC), il n'utilisaient malheureusement pas tous les mêmes séquences d'échappement. Ceci s'est produit même après que plusieurs normes ANSI (et ISO) aient été établies puisque ces normes n'ont jamais été suffisamment matures. De plus, les vieux terminaux n'avaient pas les possibilités des nouveaux terminaux. Ceci pouvait poser des problèmes. Par exemple, l'ordinateur pouvait envoyer une séquence d'échappement à un terminal lui disant de séparer l'écran en deux fenêtres de taille spécifiée, sans réaliser que le terminal était incapable de le faire.

Pour surmonter ces problèmes une base de données appelée "termcap" (maintenant "terminfo") a été mise en place. Cette base de données réside dans certains fichiers sur l'ordinateur et en une partie (quelquefois le fichier entier) pour chaque modèle de terminal. Pour chaque modèle (comme le VT100) une liste des capacités est fournie, avec une liste de certaines séquences d'échappement disponibles et ce qu'elles font. Par exemple blink=\E5m veut dire que pour faire clignoter le curseur, il faut envoyer au terminal : Escape 5 m. Voyez la section 15 (termcap et terminfo (détaillés)) pour plus de détails. Les applications peuvent utiliser cette base de données en appelant certaines fonctions de la bibliothèque C. Un grand ensemble de tels programmes (il y en a plus de 200) est appelé "ncurses" et ils sont listés sur la page de manuel de "ncurses".

5.7 L'interface

La variable d'environnement TERM décrit le type de terminal que Linux croit que vous utilisez. Certaines applications l'utilisent pour regarder les capacités dans la base de données terminfo et par conséquent TERM doit être positionnée correctement. Mais que l'ordinateur connaisse les capacités du terminal n'est qu'une partie du chemin vers une interface correcte.

Pour que les octets affluent de l'ordinateur vers le terminal, le terminal doit être configuré pour recevoir les octets à la même vitesse (en bits par seconde) qu'ils sont envoyés du terminal. Si on configure le terminal pour recevoir à 19200 bauds et que l'ordinateur envoie des caractères à 9600 bauds, on ne verra à l'écran que des parasites (ou peut-être rien du tout). On sélectionne la vitesse de transmission pour un terminal (ainsi que bien d'autres possibilités) à partir des menus de "configuration" (set-up) sur le terminal. La plupart des terminaux possèdent un grand nombre d'options dans leurs menus de "configuration" (voyez 13 (configuration du terminal (détaillée))). Le port série de l'ordinateur a aussi des options et ces options doivent être configurées d'une manière compatible (voyez 14 (configuration de l'ordinateur (détaillée))).

5.8 Émulation

La plupart des terminaux de nos jours possèdent plus d'une émulation (personnalité ou "mode de terminal"). Les numéros de modèles des terminaux fabriqués autrefois par DEC (Digital Equipment Corporation, maintenant Compaq) commencent par VT (par exemple, VT100). Bien d'autres terminaux différents des VT100 peuvent être configurés pour émuler un VT100. Wyse est un grand fabricant de terminaux et la plupart de leurs terminaux peuvent émuler des terminaux DEC divers comme les VT100 et les VT220. Par conséquent si vous voulez, disons, utiliser un terminal VT320 vous pouvez soit utiliser un vrai VT320 en personnalité "naturelle" soit utiliser un autre terminal capable d'émuler un VT320. Les personnalités "naturelles" ont en général plus de capacités donc, toutes choses étant égales par ailleurs, c'est la meilleure option à utiliser.

Le type d'émulation le plus courant est d'utiliser un PC comme si c'était un terminal vt100 (ou identique). Les programmes chargés dans la mémoire du PC permettent ceci. Dans Linux (sauf si vous êtes dans X Window) le moniteur du PC (appelé la console) émule un terminal de type "Linux" (proche du vt100). Même certaines fenêtres dans X Window émulent des terminaux. Voyez 9 (émulation de terminal).

5.9 La console

Sur un PC, le moniteur est appelé la console. Elle émule un terminal de type "Linux". On se logge dessus via un terminal virtuel. Voyez 6.5 (la console : /dev/tty?). Elle reçoit des messages du noyau concernant la progression du démarrage (boot) et de l'arrêt de la machine. On peut faire en sorte que les messages allant normalement sur la console aillent sur le terminal. Pour ceci, vous devez patcher et recompiler le noyau à la main, sauf pour les noyaux 2.2 (et plus) pour lesquels il s'agit d'une option de configuration. Voyez 16.12 (faire qu'un terminal devienne la console).

6 Fichiers spéciaux pour les terminaux tels que /dev/tty

"tty" est l'abréviation de "Teletype". Les premiers terminaux étaient des télétypes (comme machine à écrire pilotée à distance). Voyez la sous-section 22.3 (télétypes). On trouve une liste des périphériques Linux (ce qui se trouve dans le répertoire /dev) dans "Linux Allocated Devices" (en français "Périphériques référencés sous Linux"), normalement inclus dans les sources du noyau. Il "décrit" à quoi sert un périphérique en un mot ou deux mais ne vous dit pas comment l'utiliser.

6.1 Terminaux sur port série

L'ordinateur considère chaque port série comme un "périphérique". On l'appelle parfois périphérique terminal puisqu'à un moment les terminaux représentaient une utilisation courante des ports série. Pour chacun de ces ports série, il existe un fichier spécial dans le répertoire /dev (device : périphérique). /dev/ttyS0 est le fichier spécial pour le port série connu sous le nom COM1 dans le monde DOS/Windows. Pour envoyer du texte à un terminal vous pouvez rediriger la sortie standard de certains programmes en ligne de commande vers le fichier spécial approprié. Par exemple en tapant "echo test > /dev/ttyS1" à l'invite de commandes, le mot "test" devrait être envoyé sur le terminal sur ttyS1 (COM2) à condition que vous ayez la permission d'écrire sur /dev/ttyS1. De même, taper "cat mon_fichier > /dev/ttyS0" enverra le contenu du fichier mon_fichier sur COM1 (ttyS0).

En plus de ttyS0 (/dev/ttyS0), ttyS1, ttyS2, etc. (le "S" veut dire port Série) il y a aussi une série de "cua" : cua0, cua1, cua2, etc. cua0 correspond au même port que ttyS0, etc. Le "cu" dans cua veut dire CalloUt (appel sortant). Les séries ttyS sont conformes à Posix alors que l'utilisation de cua peut permettre l'ouverture d'un port dont les lignes de contrôles du modem affirment qu'il n'est pas prêt. À partir du noyau 2.2 cua est obsolète et un message d'avertissement est affiché quand vous essayez de l'utiliser (bien qu'il fonctionne encore). Pendant les dernières années, il n'était présent dans Linux que pour assurer une compatibilité ascendante. Un programmeur peut faire en sorte que ttyS se comporte comme cua, et donc cua n'est pas vraiment utile.

6.2 Pseudo-terminaux

Les pseudo-terminaux n'ont pas de connecteur physique sur l'ordinateur. On les utilise pour émuler un port série. Ils n'ont pas d'adresses E/S (IO) ni d'IRQ. Par exemple, si quelqu'un se connecte grâce à telnet sur votre ordinateur à travers un réseau, il peut se trouver connecté au périphérique /dev/ptyp2 (un port de pseudo-terminal). Dans X Window, le programme d'émulation de terminal, xterm (ou rxvt), utilise des pseudo-terminaux. Les programmes pour les radioamateurs sous Linux les utilisent aussi. Avec certaines applications, il est possible d'attacher deux ou plus de deux pseudo-terminaux sur le même port série physique.

Les pseudo-terminaux vont par deux comme ttyp3 et ptyp3. La série pty... est le terminal maître ou contrôleur et la série tty... est l'esclave. ttyq5 est aussi un pseudo-terminal comme l'est ttysc ("c" est un chiffre hexadécimal). Plus précisément, les pseudo-terminaux maîtres sont /dev/pty[p-s]N et les esclaves correspondants sont /dev/tty[p-s]N où N est un chiffre hexadécimal entre 0 et f. L'esclave /dev/ttyp agit comme un port série pour une application, elle peut y lire et y écrire. Cependant, l'application communique vraiment avec quelque soit le programme qui ecrit et lit au maitre /dev/pty correspondant. Aussi le maitre et l'esclave sont vraiment le meme "port" mais un est utilisé par l'application et l'autre par un programme réseau (ou l'équivalent) qui envoit (et récupère) des données au (et du) "port".

La norme Unix98 n'utilise pas les conventions ci-dessus, mais utilise à la place un "pty maître" qui est /dev/ptmx. Celui-ci peut fournir un pty à la demande. Les détails sur l'utilisation des pseudo-terminaux avec d'autres programmes programmes se trouvent dans quelques pages de manuels, mais un manuel dédié seulement aux pseudo-terminaux est nécéssaire pour Linux. D'autres unix-like ont une telle page nommée "pty - pseudo terminal driver".

6.3 Le terminal contrôlant /dev/tty

/dev/tty est le terminal contrôlant (s'il y en a un) le processus en cours. Pour déterminer uels ttys sont attachés à quels processus, utilisez la commande "ps -a" à l'invite de commandes (la ligne de commandes). Regardez la colonne "tty". Pour le processus shell que vous utilisez, /dev/tty est le terminal que vous utilisez à l'instant. Tapez "tty" à l'invite de commandes pour voir lequel c'est (voyez la page de manuel tty(1)). /dev/tty est quelque chose semblable à un lien vers le nom de périphérique du terminal en cours avec certaines possibilités supplémentaires pour les programmeurs en C : voyez la page de manuel tty(4).

6.4 "Terminaux" /dev/ttyIN

N représente un entier. L'une des utilisations possibles de ces terminaux dans Linux est le paquet du pilote ISDN (RNIS) : isdn4linux. La série ttyIN ressemble à ttySN. Il y a aussi une série cuiN qui ressemble à cuaN. Les séries ttyI et cui émulent des modems et on peut leur envoyer des commandes de modems.

6.5 La console : /dev/ttyN

Dans Linux le moniteur du PC est en général appelé la console et on lui associe plusieurs fichiers spéciaux de périphériques : tty0, tty1, tty2, etc. Quand vous vous loggez vous êtes sur le tty1. Pour aller sur le tty2(sur le meme ecran) appuyez sur Alt-F2, pour tty3 appuyez sur Alt(Gauche)-F3. Ces (tty1, tty2, tty3, etc.) sont appellés "terminaux virtuels". tty0 est juste un alias pour le terminal virtuel courant et c'est celui ou sont envoyés les messages du système. Aussi, les messages seront affichés sur la console (moniteur) sans se coucier de quel terminal virtuel est affiché.

Vous pouvez vous logger sur différents terminaux virtuels et ainsi avoir plusieurs sessions différentes tournant en même temps sur l'ordinateur. Seuls le système et l'utilisateur root peuvent écrire dans /dev/tty0 à laquelle /dev/console est quelquefois liée. Pour plus d'informations sur la console, voyez 9.3 (la console Linux).

6.6 Créer un périphérique avec "mknod"

Le répertoire /dev est installé avec beaucoup de fichiers spéciaux de périphériques. Si vous avez besoin de quelque chose non présent dans ce répertoire, vous pouvez essayer de le créer avec la commande "mknod". Voyez la page de manuel de ttys(4) pour savoir comment le faire pour les ports série. Pour utiliser `mknod` vous devez connaître les numéros de périphériques mineur et majeur. Vous pourriez déduire les numéros dont vous avez besoin en utilisant la commande "`ls -l`" dans le répertoire /dev. Elle affichera les numéros majeur et mineur des fichiers spéciaux existants.

7 Quelques détails sur le fonctionnement des terminaux

Si vous ne connaissez presque rien sur les terminaux, je vous suggère de lire en premier 1 (introduction) et de lire aussi 5 (survol du fonctionnement des terminaux).

7.1 Mémoire du terminal

L'écran du terminal se rafraîchit peut-être 60 fois par seconde à partir d'une image stockée dans la mémoire du terminal. Pour un PC l'image du moniteur est stockée dans la mémoire de la carte vidéo à l'intérieur de l'ordinateur mais pour un terminal, l'équivalent de la carte vidéo est à l'intérieur du terminal. Pour un terminal texte le stockage d'une image utilise peu de mémoire. Au lieu de placer chaque point (pixel) sur l'écran en mémoire, ce qui demande le stockage d'à peu près 250.000 points, on utilise une méthode de stockage bien plus efficace.

Un écran rempli de texte peut être représenté dans la mémoire du terminal par des octets ASCII, un pour chaque caractère à l'écran. Un écran entier ne prend qu'environ 2 K octets ASCII. Pour afficher ces caractères, le terminal doit aussi connaître l'image (la forme) de chacun des presque 100 caractères ASCII imprimables. L'image d'un caractère n'occupant que, disons, 15 octets, on n'a besoin que d'environ 1,5 K de mémoire pour les images de tous les caractères ASCII (la police). Ce texte ASCII et la mémoire de la police sont balayés pour que l'image résultante soit affichée à l'écran environ 60 fois par seconde. C'est une forme de mémoire partagée où l'image unique d'une lettre telle que la lettre e est partagée par toutes les nombreuses lettres e qui apparaissent sur un écran rempli de texte. Une faible demande en mémoire impliquait des coûts réduits pour produire des moniteurs au début des années 1980 quand le coût de la mémoire était plusieurs milliers de fois plus important qu'il ne l'est actuellement (ce qui représentait à l'époque plusieurs dollars par kilo-octet).

7.2 Les premiers terminaux

Les premiers terminaux ressemblaient à des machines à écrire contrôlées à distance qui ne pouvaient "afficher" (imprimer sur du papier) que le flux de caractères que leur envoyait l'ordinateur. Les premiers modèles étaient appelés 22.3 (télétypes). Le nom "tty" n'est que l'abréviation de "Teletype". Les premiers terminaux étaient capables de faire un saut de ligne et un retour chariot comme une machine à écrire et faire tinter une sonnette quand un caractère sonnette (bell) était reçu. À cause du manque de possibilités significatives, ce sont les premiers terminaux à avoir reçu l'appellation "simple" (dumb). Ce genre d'interface terminale (l'utilisation d'un type de terminal appelé "dumb") est quelque fois utilisé de nos jours quand l'ordinateur ne peut pas déterminer avec quelle sorte de terminal il communique.

7.3 Séquences d'échappement et codes de contrôle (introduction)

Les terminaux possèdent beaucoup de possibilités dont certaines sont toujours présentes ; pour les autres, l'ordinateur doit envoyer des commandes au terminal afin de modifier ou activer ces possibilités. Mettre en oeuvre toutes ces possibilités sous le contrôle d'un ordinateur nécessite la mise en place de codes spéciaux pour que l'ordinateur puisse dire au terminal ce qu'il doit faire. Il y a deux grandes sortes de codes : les séquences d'échappement et les code de contrôle (caractères de contrôle). Il y a bien plus de séquences d'échappement que de codes de contrôle.

7.3.1 Codes de contrôle

Les codes de contrôle (ou caractères de contrôle) sont constitués des 32 premiers octets de l'alphabet ASCII. Ils comprennent les codes suivants : retour chariot (curseur à l'extrémité gauche), saut de ligne (curseur une ligne vers le bas), retour en arrière, caractère d'échappement, tabulation et sonnette. Ils ne s'impriment normalement pas sur l'écran. Il y a souvent une commande que vous pouvez envoyer à votre terminal qui fera qu'ils seront affichés quand le terminal les recevra. On l'appelle parfois "Contrôles d'affichage" ou "Moniteur". Si vous faites ceci, l'affichage aura l'air légèrement dans le désordre puisque les séquences d'échappement, qui commencent toutes par le caractère de contrôle ESC (escape, échappement), ne seront plus exécutées. Les mots qui devraient apparaître en haut ou en bas de l'écran apparaîtront à d'autres endroits. Les séquences d'échappement pour repositionner le curseur s'afficheront sur l'écran mais le curseur ne se déplacera pas vers la destination indiquée par la séquence d'échappement.

7.3.2 Séquences d'échappement

Puisque qu'il n'y pas assez de codes de contrôle pour tout faire (et pour une raison ou pour une autre, ils ne sont pas tous utilisés), on utilise de nombreuses séquences d'échappement. Elles sont constituées du caractère de contrôle d'échappement (ESC) suivi d'une séquence de caractères ordinaires. En recevant un caractère d'échappement, le terminal examine les caractères suivants pour interpréter la séquence et exécuter la commande voulue par l'ordinateur. Une fois que la fin d'une séquence valide est reconnue, les caractères suivants s'affichent simplement à l'écran (sauf s'il s'agit de codes de contrôle ou de séquences d'échappement supplémentaires). Certaines séquences d'échappement peuvent prendre des paramètres (ou arguments) comme les coordonnées écran pour déplacer le curseur. Les paramètres font partie de la séquence d'échappement. Une 20.1 (liste des séquences d'échappement) se trouve sur le Web pour certains terminaux, mais c'est un peu brutal.

Une liste des séquences d'échappement pour votre terminal devrait se trouver dans le "manuel du programmeur" de votre terminal. À part pour de très vieux terminaux, il devrait y avoir deux ou trois cents séquences. Si vous n'avez pas de manuel, ce n'est pas facile de les trouver. Certaines séquences sont disponibles sur Internet. Un lien possible est 20.1 (liste des séquences d'échappement). En cherchant une séquence sur Internet (comme ESC[5m) vous pouvez tomber sur une longue liste.

Une autre façon de déterminer certaines séquences est de trouver la section termcap (terminfo) du terminal et de la décoder mentalement. Voyez 15 (terminfo et termcap (détails)) dans ce document et/ou le 15.7 (manuel termcap) sur Internet. Malheureusement, la section termcap (terminfo) d'un terminal ne liste souvent pas toutes les séquences d'échappement disponibles sur le terminal, mais heureusement, les plus importantes sont en général présentes.

7.4 Attributs d'affichage et cookies magiques

Les terminaux possèdent diverses méthodes pour générer des attributs de caractères tels que gras, vidéo inverse, soulignement, etc. L'utilisateur ne devrait pas s'inquiéter de la manière dont c'est géré, sauf que cela peut poser des problèmes pour certains vieux terminaux et il y a quelquefois une option à ce sujet dans le menu de configuration des terminaux plus récents.

La méthode des cookies magiques (magic cookie) est dépassée. C'est la méthode la plus simple (et la pire) pour définir des attributs : l'utilisation d'un octet particulier pour démarrer un attribut et un autre pour indiquer la fin de cet attribut. Par exemple, un octet magique "démarrer le soulignement" est placé juste devant le premier mot à souligner. Ces octets supplémentaires sont placés dans la mémoire de la page écran, comme les octets pour les caractères qui s'affichent normalement. Ceci peut cependant fausser le décompte du nombre de caractères par ligne puisque les caractères de cookies magiques non imprimables sont mélangés avec les autres caractères imprimables. Ceci peut dans certains cas poser des problèmes.

Une méthode plus efficace, qui utilise plus de mémoire, est d'associer un caractère d'attribut (ou un demi-caractère, etc.) à chaque caractère affiché. Cette méthode est utilisée par les cartes vidéo PC (pour le texte) pour les moniteurs PC courants.

8 Possibilités spéciales de certains terminaux

8.1 Couleur

Bien que le terminal monochrome courant ne soit pas un terminal couleur, il peut avoir un affichage en "couleur" fixe autre que blanc comme vert ou ambre. Tous les terminaux possèdent le noir (faisceau d'électrons éteint = luminosité nulle). Un véritable terminal couleur peut modifier la couleur du texte et du fond avec plusieurs couleurs différentes alors qu'un terminal monochrome ne peut modifier que la luminosité d'une couleur donnée.

Cependant, changer la luminosité, etc. offre bien des possibilités. Par exemple, un terminal noir et blanc (monochrome) peut avoir du blanc, du gris et du noir en variant la luminosité. Certains mots peuvent être en noir sur un fond légèrement gris tandis que d'autres sont mis en valeur par du noir sur fond blanc. En plus il y a du blanc sur noir, du soulignement et du clignotement.

La couleur fonctionne comme la couleur sur un moniteur d'ordinateur ou un écran de télévision. Le tube possède trois couleurs de points, chacune étant contrôlée par son propre faisceau d'électrons (il y a trois faisceaux). Le monochrome a par définition une meilleure résolution puisqu'il ne dépend pas de points fixés en permanence à l'écran. Pour les terminaux texte la seule utilisation de la couleur permet de différencier le texte et cet avantage ne vaut pas toujours le coût d'une résolution moins bonne. Le monochrome peut donc être meilleur puisqu'il est aussi moins cher.

8.2 Sessions multiples

Pour les sessions doubles, le terminal possède deux ports série de statut égal. Chaque port est connecté au port série d'un ordinateur différent. Ainsi on peut se logger sur deux ordinateurs différents, chaque session s'affichant dans une fenêtre de l'écran séparé en deux. Autrement, chaque session peut s'exécuter en plein écran et on utilise une touche "rapide" (hot key) pour accéder directement à une session (ou une même touche pour basculer). On pourrait aussi se connecter sur deux ports série différents sur le même ordinateur et se logger deux fois (de manière identique aux "terminaux virtuels" sur la console). Le programme "screen" pourra faire tourner sur n'importe quel terminal (session unique) relié à un ordinateur unique deux "sessions" ou plus.

8.3 Port imprimante/auxiliaire

Beaucoup de terminaux possèdent un connecteur à l'arrière pour un tel port. Il peut s'appeler "Aux" ou "Printer", etc. Certains ports d'imprimantes sont destinés aux imprimantes parallèle alors que d'autres sont destinés aux imprimantes série. Si une imprimante est connectée au port imprimante ou auxiliaire, alors l'appui sur certaines touches imprime l'écran. On peut aussi faire imprimer tout ce qui s'affiche à l'écran. Si le port est un port auxiliaire, on peut le connecter à un autre ordinateur et avoir des sessions doubles comme ci-dessus. Cependant, la mémoire vidéo à l'intérieur du terminal peut ne pas retenir les deux sessions et vous pouvez avoir besoin de rafraîchir l'écran en basculant sur l'autre session. Il n'y aura peut-être pas de touche rapide non plus mais on pourra peut-être programmer une touche de fonction pour accomplir cette tâche. Il existe de nombreuses combinaisons de touches et de séquences d'échappement pour contrôler un tel port. Voyez 20.3 (échappement de l'imprimante).

Il y a un programme appelé `vtprint` qui est fait pour envoyer un travail d'impression à votre terminal en vue de l'imprimer sur une imprimante reliée au terminal. Sa page Web est *http://people.qualcomm.com/garrett/vtprint* . Il est aussi inclus (en 1998) dans la distribution Debian de Linux. `xprt` (aussi dans Debian) semble faire une chose identique, mais seulement pour les terminaux X Window ??

8.4 Pages

Beaucoup de terminaux permettent le stockage de plus d'une page dans leur mémoire vidéo. Quelquefois la taille de la page est la même que celle de l'écran, mais elle est parfois plus grande, ce qui fait que le défilement révèlera les parties invisibles de la page. Quand quelqu'un regarde un écran, il peut y avoir du texte caché sur la même page au-dessus ou en dessous de l'affichage. En plus, s'il y a plus d'une seule page, il peut y avoir du texte caché sur ces autres pages. Une utilisation possible de ces pages concerne les terminaux qui supportent les sessions doubles. Chaque session peut avoir sa propre page et on peut basculer de l'une à l'autre simplement.

Même si vous avez un terminal à page unique avec la taille de la page égale à ce qui est affiché sur l'écran, vous verrez encore d'autres pages d'un fichier (etc.) tant que l'hôte envoie des données au terminal. L'un des avantages de stocker des pages supplémentaires dans la mémoire du terminal est que vous pouvez y accéder instantanément sans attendre une seconde ou presque qu'elles soient transmises depuis l'hôte.

Les pages multiples sont supportées par ncurses. Faites-moi savoir si vous en connaissez. Il y a un programme commercial appelé "Multiscreen" qui les supporte mais il n'est sûrement pas pour Linux ?? Il parait que Multiscreen fait partie de SCO et fait quelque chose dans le genre des terminaux virtuels sur une console PC Linux. Le programme Linux "screen" fait semblant d'avoir des pages multiples mais elles sont stockées sur l'ordinateur et vous ne pouvez avoir qu'une fenêtre-page pour chaque programme en cours.

8.5 Jeux de caractères

Un jeu de caractères est normalement représenté par une liste (ou table ou tableau) de caractères accompagnée du code d'octet assigné à chaque caractère. Les codes d'un octet vont de 0 à 255 (00 à FF en hexadécimal). Dans MS-DOS, les tables de jeux de caractères sont appelées "pages de code". Vous devriez examiner une telle table si vous n'êtes pas familier avec celles-ci. Elles sont parfois incluses dans les manuels d'imprimantes et de terminaux mais peuvent être difficiles à trouver.

L'ASCII est l'un des jeux de caractères les plus courants utilisés sur les terminaux texte. C'est un code sur 7 bits mais qui peut être converti en 8 bits si le premier bit (bit de haut rang) est toujours positionné à 0. D'autres jeux de caractères sont en général disponibles (sauf sur de très vieux terminaux où le seul choix est l'ASCII). La première moitié de la plupart des jeux de caractères sont les caractères ASCII traditionnels et la deuxième moitié (les caractères avec le bit de haut rang mis à 1) appartiennent à une grande variété de jeux de caractères. Les jeux de caractères sont souvent des normes ISO. Pour obtenir des jeux de caractères spécialisés sur un terminal, vous devrez certainement télécharger une police logicielle pour ce jeu de caractères dans la mémoire du terminal.

À côté de l'ASCII, il y a d'autres jeux de caractères courants, tous en 8 bits. CP est l'abréviation des jeux de caractères de page de codes (Code Page) inventés par IBM : CP-437 (ECS DOS), CP-850 (Latin 1 multilingue – pas le même que ISO Latin-1), ISO-8859-1 (Latin-1), ANSI (dérivé de Latin-1). MS Windows utilise ANSI alors qu'Internet utilise souvent Latin-1. Il y a plusieurs jeux de caractères ISO-8859 en plus de Latin-1. Ceux-ci comprennent le grec (-7), l'arabe (-6), l'Europe de l'Est (-2) et le russe (-5). Il y en a bien d'autres. Par exemple, KOI8-R est plus souvent utilisé pour le russe que ISO-8859-5. Unicode est un jeu de caractères très grand dans lequel chaque caractère est représenté sur deux octets au lieu d'un seul.

Voici plus d'informations sur les ensembles de caractères :

- Pages de manuel : ASCII et latin1
- Les HOWTOs pour diverses langues (sûrement écrits dans cette langue spécifique). Voyez "Cyrillic" pour le russe.
- `http://www.cc.columbia.edu/kermit/charsets.html` pour une liste courte des divers noms de jeux de caractères.
- `http://www.pku.edu.cn/on_line/w3html/International/Overview.html` pour des informations sur les jeux de caractères et Internet.
- *Langues, pays et jeux de caractères* `http://www.w3.org/International/O-charset-lang.html`
- *Utiliser plusieurs langues en HTML* `http://vancouver-webpages.com/multilingual/`

Une fois que vous aurez le nom (ou le numéro) du jeu de caractères qui vous intéresse, vous pouvez chercher plus d'informations dessus sur Internet.

8.6 Polices

La plupart des terminaux fabriqués après le milieu des années 1980 peuvent accepter des polices logicielles téléchargées. Ceci veut dire qu'ils peuvent afficher quasiment n'importe quel jeu de caractères à condition que vous trouviez la police logicielle pour celui-ci. Si vous ne pouvez pas trouver la police logicielle nécessaire, vous pouvez toujours créer la vôtre. Un éditeur de polices libre pour effectuer ceci s'appelle BitFontEdit (écrit par l'auteur de ce document) et se trouvait (en 1998) à
Europe : `http:/www.funet.fi/pub/culture/russian/comp/cyril-term/`
Amérique du Nord : `ftp://cs.utk.edu/pub/shuford/BitFontEdit.tar.gz`

8.7 Claviers et touches spéciales

Les claviers des terminaux ont souvent un certain nombre de touches qu'on ne trouve pas sur un clavier de PC. Peu (ou pas du tout) de terminaux actuels auront toutes ces touches et la plupart auront des touches supplémentaires qui ne sont pas listées ici. Certains d'entre eux possèdent un grand nombre de touches à usage spécial comme les terminaux faits pour les caisses enregistreuses. Il y a souvent bien plus de significations pour les touches que ce qui est décrit ici puisque ces touches ont souvent une signification étendue quand on les utilise avec d'autres touches (comme les touches shift et control).

- BREAK (interruption) envoie un bit 0 très long (intervalle = +12 V) de durée de 300 à 700 millisecondes vers l'hôte. L'hôte peut l'interpréter comme une interruption si stty a positionné brkint ou l'ignorer si ignbrk est positionné.
- NO SCROLL (pas de défilement) arrête le défilement de l'écran comme le fait ˆS. Presser la touche à nouveau reprend le défilement. Utilise des signaux de contrôle de flux pour faire ceci.
- REPEAT (répétition) si on la maintient avec une autre touche, force la sortie répétée de cette autre touche même si l'option de répétition automatique est désactivée.
- LINE FEED (saut de ligne) envoie le caractère de saut de ligne ˆJ à l'hôte. Rarement utilisée.
- SET-UP (configuration) permet la configuration manuelle du terminal par l'intermédiaire de menus. Quelquefois désactivée à dessein en plaçant un bloc en dessous pour qu'on ne puisse pas la presser. Parfois il faut appyer sur une autre touche en même temps comme Shift ou Control. Voyez 13.3 (entrer dans le mode de configuration).
- LOCAL déconnecte le terminal de l'hôte. En local, ce qu'on tape va directement à l'écran. Utile pour faire des tests.
- RETURN (retour chariot) est la même chose que la touche "Entrée" sur un PC. Elle envoie en général un retour chariot à l'hôte qui est normalement traduit en un caractère nouvelle ligne par le pilote de périphériques de l'hôte. Sur certains terminaux on peut le configurer pour qu'il envoie autre chose.
- F1, F2, ... ou PF1, PF2, ... sont des touches de fonction qu'on peut généralement programmer pour envoyer une séquence d'octets (caractères). Voyez 13.14 (touches de fonction)

9 Émulation de terminal ; la console

9.1 Émulation de terminal

9.1.1 Introduction à l'émulation de terminal

Puisqu'un PC possède un écran et un clavier (comme un terminal) mais possède aussi bien plus de puissance de calcul, il est facile d'utiliser une partie de cette puissance de calcul pour que le PC se comporte comme un terminal texte. Voici un type d'émulation de terminal. Un autre type d'émulation de terminal est de configurer un vrai terminal pour qu'il émule un terminal de modèle ou de marque différents. Pour ce faire, sélectionnez l'émulation désirée (que le jargon Wyse appelle "personnalité") à partir du menu de paramétrage du terminal. Cette section parle du premier type d'émulation : l'émulation d'un terminal sur un PC.

Les logiciels d'émulation sont disponibles pour MS Windows et sont intégrés dans les versions récentes de MS Windows. La plupart des logiciels Linux ne peuvent émuler que des VT100, VT102 ou VT100/ANSI. Si vous en trouvez d'autres, dites-le moi. Puisque la plupart des PC possèdent des moniteurs couleur mais que les VT100 et VT102 ont été faits pour des terminaux monochromes, l'émulation ajoute en général des possibilités de gestion de couleurs (et un choix de couleurs). Parfois l'émulation n'est pas parfaite à 100 % mais ceci ne crée en général que peu de problèmes. Pour utiliser un Macintosh pour émuler un terminal, voyez le mini-Howto : Mac-Terminal.

9.1.2 N'utilisez pas TERM pour l'émulation

Certaines personnes ont pensé, de manière incorrecte, qu'ils pouvaient faire un émulateur sur la console Linux (le moniteur) en positionnant la variable d'environnement TERM sur le type de terminal qu'elles aimeraient émuler. Ceci ne fonctionne pas. La valeur de TERM ne fait qu'indiquer à l'application le terminal que vous utilisez. Ainsi elle n'a pas besoin de vous le demander directement. Si vous êtes sur le moniteur du PC, c'est un terminal de type "Linux" et vous ne pouvez rien y faire. Vous devez donc positionner TERM à "Linux".

Si vous mettez autre chose, vous racontez des bobards à votre application. En conséquence, elles interpréteront certaines séquences d'échappement provenant de la console de manière incorrecte, ce qui donnera une interface corrompue. Puisque la console Linux se comporte presque comme un terminal vt100, elle pourrait encore fonctionner correctement si vous indiquiez que c'est un vt100 (ou tout autre terminal qui ressemble à un vt100). Cela doit marcher la plupart du temps, mais il suffit d'une fois pour qu'il y ait une erreur quand vous editez ou equivalent.

9.1.3 Programmes de communications (appels téléphoniques)

Un programme d'émulation est souvent couplé à un programme d'appel par modem (comme Minicom, Seyon ou Kermit) pour que l'on puisse (par exemple) appeler par téléphone des bibliothèques publiques pour utiliser leurs catalogues et index (ou même lire des articles de magazine). Seyon ne s'utilise qu'avec X Window et peut émuler des terminaux Tektronix 4014. Des émulateurs existent pour DOS comme `telix` et `procomm` et fonctionnent aussi bien. Les terminaux émulés sont souvent les vieux VT100, VT102 ou ANSI (comme VT100).

9.1.4 Émulation sous X Window

Xterm (obsolète ??) peut fonctionner sous X Window et permet d'émuler un VT102, VT220 ou Tektronix 4014. Il y a aussi une émulation xterm (bien qu'il n'y ait pas de terminal physiques qui s'appelle "xterm"). Si vous n'avez pas besoin de l'émulation Tektronix 4014 (un terminal avec des vecteurs graphiques ; voyez 2.3

(terminaux graphiques)) vous pouvez utiliser `eterm`. Les prédécesseurs de `eterm` sont `rxvt` et `xvt`. `eterm` supporte les pixmaps en fond de fenêtre.

Pour les alphabets différents de Latin, kterm permet l'émulation de terminal Kanji (ou tout autre alphabet non Latin) alors que xcin est fait pour le chinois. Il y a aussi l'émulateur 9term. Il semble que ce soit plus qu'un simple émulateur puisqu'il possède un éditeur intégré et des barres de défilement. Il a été fait pour Plan 9, un système d'exploitation à la Unix de AT&T.

9.1.5 Les vrais terminaux sont mieux

Sauf si vous utilisez X Window avec un grand affichage, il est plus agréable d'utiliser un vrai terminal que de l'émuler. Il coûte en général moins cher, a une meilleur résolution pour le texte et n'a pas de lecteurs de disques qui font des bruits agaçants.

9.2 Tester l'émulation du terminal

Pour la série des terminaux VT il y a un programme de test : `vttest` pour aider à déterminer si un terminal se comporte correctement comme un vt53, vt100, vt102, vt220, vt320, vt420, etc. Il n'y a pas de documentation mais il a des menus et est facile à utiliser. Pour le compiler lancez le script configure et ensuite tapez "make". On peut le charger à : `ftp://ftp.clark/net:/pub/dickey/vttest/` . Un autre site de chargement est : `http://metalab.unc.edu/pub/Linux/utils/console/` .

9.3 La console Linux

La console d'un système PC sous Linux est le moniteur de l'ordinateur. Il émule un terminal de type "Linux". Il n'y a pas moyen (sauf si vous voulez passer des jours à réécrire le code du noyau) pour le faire émuler autre chose. Positionner la variable d'environnement TERM à un type de terminal différent de "Linux" n'aboutira pas à l'émulation de cet autre terminal. Vous n'obtiendrez qu'une interface corrompue puisque vous avez déclaré faussement (par l'intermédiaire de la variable TERM) que votre "terminal" est d'un type différent de ce qu'il est. Voyez la section 9.1.2 (n'utilisez pas TERM pour l'émulation).

L'émulation "Linux" est souple et possède des possibilités qui vont bien au-delà de celles du terminal vt102 qu'il était censé émuler. Celles-ci comprennent la possibilité d'utiliser des fontes personnalisées et de reconfigurer facilement le clavier (sans modifier le code source et recompiler le noyau comme il faut le faire dans le cas d'un terminal réel). Ces possibilités supplémentaires se trouvent dans le logiciel de pilotage de la console et non dans le logiciel d'émulation mais le résultat fait qu'on dirait que ces possibilités font partie de l'émulation.

Beaucoup de commandes existent (voyez le HOWTO Keyboard-and-Console) pour utiliser ces possibilités supplémentaires. Les vrais terminaux, qui n'utilisent ni codes de balayages (scancodes) ni carte VGA, ne peuvent malheureusement pas utiliser la plupart de ces possibilités. On peut recompiler Linux pour qu'un terminal reçoive les messages qui vont normalement sur la console. Voyez 16.12 (transformer un terminal en console).

10 Contrôle de flux (prise de contact)

Le contrôle de flux (= prise de contact (handshaking) = ralentissement) permet d'empêcher un flux d'octets trop rapide de dépasser un terminal, un ordinateur, un modem ou un autre périphérique. Le dépassement est le fait qu'un périphérique ne puisse pas traiter ce qu'il reçoit assez rapidement et ainsi perd des octets et/ou fait d'autres erreurs sérieuses. Ce que fait le contrôle de flux est d'arrêter le flux d'octets jusqu'à ce que le terminal (par exemple) soit prêt à recevoir des octets supplémentaires. Le contrôle de flux envoie un signal pour arrêter le flux dans la direction opposée au flux des données qu'il veut arrêter. Le contrôle de flux doit être lancé à la fois sur le terminal et sur l'ordinateur.

Il y a deux types de contrôle de flux : matériel et logiciel (Xon/Xoff ou DC1/DC3). Le contrôle de flux matériel utilise des fils de signaux dédiés comme RTS/CTS ou DTR/DSR alors que le contrôle de flux logiciel se signale en envoyant les octets de contrôle DC1 ou DC3 dans les fils de données normaux. Pour le contrôle de flux matériel, le câble doit être câblé correctement.

Le flux des octets de données dans le câble entre deux ports série est bidirectionnel, il y a donc deux flux (et deux fils) différents à considérer :

1. le flux de données de l'ordinateur vers le terminal
2. le flux de données du clavier du terminal vers l'ordinateur

10.1 Pourquoi le contrôle de flux est-il nécessaire ?

Vous pouvez vous demander : "Pourquoi ne pas envoyer les données à une vitesse suffisamment petite pour que le périphérique ne soit pas dépassé et que le contrôle de flux ne soit ainsi plus nécessaire ?" Ceci est possible mais c'est en général bien plus lent que d'envoyer les données plus rapidement et d'utiliser le contrôle de flux. Une raison à ceci est qu'on ne peut pas positionner la vitesse du port série à n'importe quelle vitesse comme 14.500, puisqu'un nombre limité de choix est disponible. Le meilleur choix est de sélectionner une vitesse légèrement plus élevée que ce que peut soutenir le périphérique et d'utiliser ensuite le contrôle de flux pour que les choses fonctionnent correctement.

Si on décide de ne pas utiliser le contrôle de flux, la vitesse doit alors être suffisamment basse pour pallier à la pire des situations. Pour un terminal, cela arrive quand on envoie des séquences d'échappement pour effectuer des tâches complexes qui prennent plus de temps qu'à l'accoutumée. Dans le cas d'un modem (avec la compression de données mais pas de contrôle de flux) la vitesse de l'ordinateur au modem doit être suffisamment basse pour que cette même vitesse soit utilisable sur la ligne téléphonique, puisque dans le pire des cas les données sont aléatoires et ne peuvent être compressées. Si on ne pouvait pas utiliser de contrôle de flux, la vitesse (avec la compression de données activée) ne serait pas plus rapide que si on n'utilisait pas de compression du tout.

Les buffers (mémoires tampons) aident à gérer les situations catastrophes de courte durée. Le tampon stocke les octets qui arrivent trop rapidement pour être traités tout d'un coup, et les garde pour les traiter plus tard.

10.2 Remplissage

Une autre manière de gérer une situation "catastrophe" (sans utiliser de contrôle de flux ni de tampon) est d'ajouter un groupe de nulls (octets de valeur zéro) aux séquences d'échappement. Quelquefois on utilise des DEL à la place, à condition qu'ils n'aient pas d'autre fonction. Voyez 13.12.3 (reconnaître DEL).

La séquence d'échappement permet au terminal de commencer à faire quelque chose, et pendant que le terminal est occupé à le faire, il reçoit une poignée de nulls qu'il ignore. Quand il reçoit le dernier null, il a terminé sa tâche et est prêt pour la commande suivante. C'est ce qu'on appelle le remplissage de zéros (null padding). Ces nulls étaient autrefois appelés des "caractères de remplissage". Ces nulls sont ajoutés simplement pour "perdre" du temps, mais ce n'est pas tout à fait perdu puisque le terminal est en général occupé à faire autre chose pendant que les nulls sont reçus. On utilisait beaucoup cette méthode dans le passé avant que le contrôle de flux ne devienne populaire. Pour être efficace, il fallait ajouter le nombre exact de nulls et trouver la bonne valeur est difficile. On le faisait souvent par essais successifs et tâtonnements puisque les manuels de terminaux n'étaient pas de grand secours. Si le contrôle de flux ne fonctionne pas correctement ou n'est pas implémenté, le remplissage est une solution. Certaines options de la commande `stty` concernent le remplissage.

10.3 Débordement d'un port série

On peut se demander comment le débordement est possible sur un port série puisqu'à la fois les ports série d'envoi et de réception servant à la transmission d'octets de données sont paramétrés pour la même vitesse (en bits/s) comme 19200. La raison est que bien que l'électronique du port série récepteur peut gérer la vitesse du flux arrivant, le matériel/logiciel qui prend et traite les octets du port série ne peut pas toujours se débrouiller avec une vitesse de flux élevée.

L'une des causes de ceci est que le tampon matériel du port série est assez petit. Les anciens ports série avaient une taille de tampon matériel d'un octet seulement (à l'intérieur de la puce UART). Si cet unique octet de données reçu dans le tampon n'est pas enlevé (pris) par des instructions CPU avant que l'octet suivant n'arrive, cet octet est perdu (le tampon est débordé). Les UART récents, par exemple la plupart des 16550A, possèdent des tampons de 16 octets (mais peuvent être paramétrés pour émuler un tampon d'un octet) et sont moins susceptibles d'être débordés. On peut le paramétrer pour envoyer une interruption quand le nombre d'octets dans son tampon atteint 1, 4, 8 ou 14 octets. C'est le travail d'une autre puce dans l'ordinateur (généralement la puce principale CPU pour un ordinateur) de retirer ces octets entrants de ce petit tampon matériel et de les traiter (ainsi que d'effectuer d'autres tâches).

Quand le contenu de ce petit tampon matériel de réception atteint la limite spécifiée (un octet pour les vieux UART) une interruption est levée. L'ordinateur interrompt alors ce qu'il était en train de faire et une routine fait une vérification pour déterminer ce qui vient de se passer. Il détermine finalement qu'il doit retirer un octet (ou plusieurs) du tampon du port série. Il prend cet (ces) octet(s) et les met dans un tampon plus grand (un autre tampon pour le port série) que le noyau maintient dans la mémoire principale. Pour le tampon de transmission, le matériel série génère une interruption quand le tampon est vide (ou presque vide) pour dire à la CPU de mettre quelques octets supplémentaires dans ce tampon afin de les envoyer.

Les terminaux possèdent aussi des ports série et des tampons similaires à ceux de l'ordinateur. Puisque le flux de données des octets vers le terminal est en général plus grand que le flux dans la direction opposée du clavier vers l'ordinateur hôte, le terminal a plus de chance de souffrir du débordement. Bien sûr, si vous utilisez un ordinateur comme terminal (par émulation), il est à son tour sujet au débordement.

Les situations risquées où le débordement est très probable sont : 1. quand un autre processus a désactivé les interruptions (pour un ordinateur), 2. quand le tampon du port série dans la mémoire principale (ou dans celle du terminal) est prête à déborder.

10.4 Arrêt de l'envoi

Quand le récepteur est sur le point d'être débordé par les octets entrants, il envoie un signal à l'expéditeur pour arrêter l'envoi. C'est le contrôle de flux et les signaux de contrôle de flux sont toujours envoyés dans la direction opposée au flux de données qu'ils contrôlent (bien que ce ne soit pas dans le même canal ou le même fil). Ce signal peut être soit un caractère de contrôle (^S = DC3 = Xoff) envoyé comme un octet de données ordinaire sur la ligne de données (signalement dans la bande), soit une transition de tension du positif au négatif dans le fil de signal dtr-vers-cts (ou autre ; signalement hors-bande). L'utilisation de Xoff est appelée "contrôle de flux logiciel" et l'utilisation du saut de tension dans un fil de signal dédié (à l'intérieur du câble) est appelée contrôle de flux matériel.

10.5 Blocage du clavier

Avec les terminaux, le cas le plus commun "d'arret d'envoi", est quand le terminal ne peut pas suivre avec les caractères qui lui sont envoyés et qui en conclut par un "arret" du PC. Un autre cas, est quand quelqu'un presse control-S. Un autre cas un peu moin commun, est l'opposé, quand le PC ne peut plus suivre votre vitesse de frappe et dit au terminal d'arrêter l'envoi. Le terminal "bloque" le clavier et un message ou une lumière devrait vous informer que le clavier est bloqué. Tout ce que vous tapez sur un clavier bloqué est ignoré.

Le terme "bloqué" est aussi quelque fois utilisé pous les cas où l'on dit à l'ordinateur d'arrêter d'envoyer à un terminal. Le clavier n'est pas bloqué, afin que tout ce que vous tapez soit envoyé à l'ordinateur, mais puisque l'ordinateur ne peut rien vous renvoyer, les caractères que vous tapez ne s'affichent pas sur l'écran et il peut sembler que le clavier est bloqué. Le défilement est bloqué (scroll lock) mais le clavier n'y est pas.

10.6 Reprendre l'envoi

Quand le récepteur a rattrapé son retard dans le traitement et est prêt à recevoir plus d'octets de données il envoie un signal à l'envoyeur. Pour le contrôle de flux logiciel ce signal est le caractère de contrôle ˆQ = DC1 = Xon qui est envoyé sur la ligne de données normale. Pour le contrôle de flux matériel la tension dans une ligne de signal passe de négative (niée) à positive (affirmée). Si on dit à un terminal de reprendre la transmission le clavier est alors débloqué et prêt à être utilisé.

10.7 Contrôle de flux matériel (RTS/CTS, etc.)

Certains terminaux anciens n'offrent pas de contrôle de flux matériel alors que d'autres offraient un assortiment varié de broches diverses sur le port série pour le faire. Pour une liste des differentes borches, aller voir 11.2.2 (Brochage standard d'un cable null-modem). La broche la plus en vogue actuellement semble être la broche DTR (ou les broches DTR et DSR ensemble).

10.7.1 Contrôle de flux RTS/CTS, DTR et DTR/DSR

Les PC Linux utilisent RTS/CTS mais le contrôle de flux DTR/DSR (utilisé par certains terminaux) se comporte de la même manière. Le contrôle de flux DTR (dans une seule direction et aussi utilisé par certains terminaux) n'est que la partie DTR du contrôle de flux DTR/DSR.

RTS/CTS utilise les broches RTS et CTS sur le connecteur série (EIA-232). RTS veut dire "Request To Send" (demande d'envoyer). Quand cette broche reste en position haute (tension positive) sur le récepteur cela veut dire : continuez de m'envoyer des données. Si RTS passe en position basse (la tension devient négative), cela nie "demande d'envoyer", ce qui veut dire : arrêtez d'envoyer. Quand le récepteur est prêt à recevoir plus de données, il relance RTS, demandant à l'autre côté de reprendre l'envoi. Pour les ordinateurs et les terminaux (tous les deux des équipements terminaux) la broche RTS envoie le signal de contrôle de flux à la broche CTS (Clear To Send, prêt à envoyer) de l'autre côté du câble. C'est-à-dire que la broche RTS à un bout du câble est reliée à la broche CTS à l'autre bout du câble.

Pour un modem (équipement de connexion) le principe est différent puisque la broche RTS du modem reçoit le signal et sa broche CTS l'envoie. Alors que ceci peut sembler déroutant, il y a des raisons historiques correctes pour l'expliquer, raisons qui sont trop compliquées pour en discuter ici.

Les terminaux disposent en général du contrôle de flux DTR ou DTR/DSR. Le contrôle de flux DTR est le même que le contrôle de flux DTR/DSR mais il est unidirectionnel et la broche DSR n'est pas utilisée. En ce qui concerne le contrôle de flux DTR/DSR sur un terminal, le signal DTR est comme le signal envoyé de la broche RTS, et la broche DSR est simplement comme la broche CTS.

10.7.2 Etablir une connexion avec le contrôle de flux DTR ou DTR/DSR

Certains terminaux n'utilisent que le contrôle de flux DTR. C'est un contrôle de flux unidirectionnel uniquement pour empêcher le terminal d'être dépassé. Il ne protège pas l'ordinateur de quelqu'un qui tape trop vite pour que l'ordinateur puisse gérer la situation. Dans un câble null modem classique la broche DTR du terminal est reliée à la broche DSR de l'ordinateur. Linux, par contre, ne supporte pas le contrôle de flux DTR/DSR (bien que des pilotes pour des cartes multiports peuvent supporter le contrôle de flux DTR/DSR). Un moyen de contourner ce problème est simplement de relier la broche DTR à la broche CTS sur l'ordinateur et d'activer le contrôle de flux RTS/CTS (stty crtscts). Le fait que ce soit unidirectionnel ne changera rien tant que l'hôte n'est pas dépassé par votre vitesse de frappe et ne lâche RTS en une vaine tentative pour bloquer votre clavier. Voyez 10.5 (blocage du clavier). Pour obtenir le contrôle de flux DTR/DSR (si votre terminal supporte ce type de contrôle de flux bidirectionnel) vous faites ce qui est décrit ci-dessus. Mais vous connectez aussi la broche DSR sur le terminal à la broche RTS sur l'ordinateur. Vous êtes alors protégé si vous tapez trop rapidement.

10.7.3 L'ancienne prise de contact RTS/CTS est différente

Ce qui est déroutant est que l'utilisation d'origine de RTS veut dire à peu près le contraire de l'explication précédente ci-dessus. La signification d'origine est : je demande à vous envoyer (I Request To Send to you). Cette requête était destinée à être envoyée d'un terminal (ou d'un ordinateur) vers un modem qui, s'il décidait d'accorder la requête, renvoyait un CTS affirmatif à partir de sa broche CTS vers la broche CTS de l'ordinateur : vous êtes autorisé à m'envoyer (You are Cleared To Send to me). Notez qu'au contraire du contrôle de flux RTS/CTS bidirectionnel du modem, ceci ne protège le flux que dans une direction : de l'ordinateur (ou du terminal) vers le modem.

Pour de vieux terminaux, RTS peut avoir cette signification et devient positif quand le terminal doit envoyer des données. L'utilisation ci-dessus est une forme de contrôle de flux puisque si le modem veut que l'ordinateur arrête d'envoyer il lâche CTS (connecté au CTS de l'ordinateur) et l'ordinateur arrête d'envoyer.

10.7.4 Canal inversé

Les vieux terminaux à sortie papier peuvent avoir une broche de canal inversé (comme la broche 19) qui se comporte comme la broche RTS dans le contrôle de flux RTS/CTS. Cette broche passera aussi en négatif s'il n'y a plus de papier ou de ruban. Il est souvent possible de relier cette broche à la broche CTS de l'ordinateur hôte. Il peut y avoir un petit interrupteur pour positionner la polarité de ce signal.

10.8 Est-ce que le contrôle de flux matériel est fait par le matériel ?

Certains pensent que le contrôle de flux matériel est fait par le matériel mais (sauf si vous utilisez une carte série intelligente avec plusieurs ports série) c'est en réalité votre système d'exploitation qui s'en charge. Les puces UART et le matériel associé ne connaissent en général rien du contrôle de flux matériel. Quand un signal de contrôle de flux matériel est reçu, le fil du signal inverse la polarité. Ce changement d'état est enregistré dans un registre de port série qui est vérifié par le pilote série avant de mettre les octets dans les tampons materiels de 16 octets. Si le fil du control de flux dit "stop", il n'y a plus d'octets ajoutés et le flux sortant des lignes séries s'arrete.

Il y'a un autre moyen qui aurait pu être implementer depuis que la polarité s'inverse, le materiel aurait pu être configuré pour envoyer un signal éléctrique d'interruption au processeur. Alors le processeur arreterait ce qu'il était en train de faire, se brancherait a un sous-programme de service du pilote série, verifierait les registres dans lesquels le port serie a laissé des traces pour trouver ce qui s'est passé, et fais un rapport, pour ne pas redémarrer le flux apres que le sous programme de service soit quitté. Cela doit être un peu plus efficace, mais il semble que Linux n'agisse pas comme ca. A mon avis.

Noter qu'avec l'une ou l'autre des methodes, le flux d'octets s'arrette quasiment instantanement. Cependant tous les octets (jusqu'à 16) qui étaient déjà dans le tampon de transmission matériel du port série seront encore transmis. Utiliser un control de flux logiciel requiert que chaque octet arrivant soit verifié pour voir si c'est un octet "eteint". Ces octets sont retardés en passant à travers le tampon de réception de 16 octets. Si l'octet "éteint" était le premier octet dans ce tampon, il pourrait y avoir une attente le temps que 15 octets soient reçus. Alors les 16 octets lus seraient obtenus et l'octet "éteint" trouvé. Cette attente supplementaire n'arrive pas avec un control de flux materiel.

10.9 Obsolète ?? Contrôle de flux ETX/ACK ou ENQ/ACK

Ceci est aussi du contrôle de flux matériel et nécessite un pilote de périphérique qui sait le traiter. Les octets sont envoyés par paquets (grâce au port série asynchrone), chaque paquet étant terminé par un caractère de contrôle ETX (End of Text, fin de texte). Quand le terminal reçoit un ETX il attend jusqu'à ce qu'il soit prêt à recevoir le paquet suivant et retourne alors un ACK (Acknowledge, acquittement). Quand l'ordinateur reçoit le ACK, il envoie le paquet suivant. Et ainsi de suite. Ceci n'est pas supporté par Linux ?? Certains terminaux HP utilisent la même méthode mais utilisent ENQ au lieux de ETX.

11 Connexion physique

Les cartes multiports permettent de relier plusieurs terminaux (ou modems) à un ordinateur. On peut relier un terminal à son ordinateur hôte soit par une connexion directe par câble, soit par un modem, ou encore par l'intermédiaire d'un serveur de terminaux.

11.1 Cartes d'entrées/sorties multiports (adaptateurs)

On peut acheter des cartes série supplémentaires qui possèdent plusieurs ports série, et qu'on appelle "cartes multiports". Ces cartes ne sont pas détaillées dans ce HOWTO mais on en parle beaucoup dans le HOWTO Serial. Une société qui fait des prix en dessous de la moyenne est *ByteRunner* `http://www.byterunner.com/cgi-bin/goto.cgi?FILE=iocards.html` .

11.2 Connexion directe par câble

La manière la plus simple de relier un terminal à un ordinateur hôte est par l'intermédiaire d'une connexion directe vers un port série de l'ordinateur. La plupart des PC possèdent deux ports série, mais la souris en utilise en général un. Vous pouvez aussi utiliser quelques infos dans cette section pour connecter un ordinateur a un autre(par le port série). Pour le port EIA-232, vous avez besoin d'un câble null modem qui croise les fils de transmission et de réception. Si vous voulez faire du contrôle de flux matériel, vous utiliserez sûrement la broche DTR (ou les broches DTR et DSR ensemble).

Assurez-vous que vous avez le bon type de câble. Un câble null modem acheté dans un magasin d'ordinateurs peut être bon (s'il est assez long), mais ne fonctionnera probablement pas bien pour le contrôle de flux matériel. Ce genre de câble peut être étiqueté comme un câble pour imprimante série. Voyez ceci pour vous aider à déterminer si vous devez 11.6.1 (acheter ou fabriquer) votre propre câble. Assurez-vous que vous utilisez votre port série, la prise mâle DB25 ou DB9, et non pas votre port parallèle (prise DB25 femelle ou Centronics).

11.2.1 Schéma de brochage des câbles null modem (3, 4 ou 5 conducteurs)

Ces 3 diagrammes sont pour les vrais terminaux textes. Mais vous pouvez les utiliser pour connecter 2 PC si vous remplacer RTS par DTR et CTS par DSR. (N'utilisez pas 4 conducteurs pour PC à PC). Pour les terminaus, si vous n'avez que le contrôle de flux DTR (unidirectionnel), vous pouvez éliminer le fil RTS-vers-DSR. Si vous n'avez pas de contrôle de flux matériel, vous pouvez alors aussi éliminer le fil CTS-vers-DTR. Alors, si vous avez deux paires torsadées, vous pouvez utiliser deux fils pour la masse du signal comme indiqué à 11.2.6 (une astuce qui utilise du câble en paire torsadée). Pour un connecteur DB25 sur votre PC, vous avez besoin de :

```
 PC DB25 mâle                             Terminal DB25
   TxD   Transmit Data       2 --> 3       RxD   Receive Data
   RxD   Receive Data        3 <-- 2       TxD   Transmit Data
   SG    Signal Ground       7 --- 7       SG    Signal Ground
   CTS   Clear To Send       5 <-- 20      DTR   Data Terminal Ready
   RTS   Request To Send     4 --> 6       DSR   Data Set Ready
```

Si vous avez un connecteur DB9 sur votre PC, essayez le schéma suivant :

```
      PC DB9                                     Terminal DB25
   RxD   Receive Data        2 <-- 2       TxD   Transmit Data
   TxD   Transmit Data       3 --> 3       RxD   Receive Data
   SG    Signal Ground       5 --- 7       SG    Signal Ground
   CTS   Clear To Send       8 <-- 20      DTR   Data Terminal Ready
   RTS   Request To Send     7 --> 6       DSR   Data Set Ready   **
```

Si vous avec un connecteur DB9 sur tous vos ports séries et terminal :

```
      PC DB9                                     Terminal DB25
   RxD   Receive Data        2 <-- 3       TxD   Transmit Data
   TxD   Transmit Data       3 --> 2       RxD   Receive Data
   SG    Signal Ground       5 --- 5       SG    Signal Ground
   CTS   Clear To Send       8 <-- 4       DTR   Data Terminal Ready
   RTS   Request To Send     7 --> 6       DSR   Data Set Ready   **
```

Les schémas ci-dessus n'ont pas de lignes de contrôle de modem, soyez donc sûr de donner une option "local" à getty (ce qui est équivalent à "stty clocal"). De plus si vous avez besoin du contrôle de flux matériel il doit être activé du côté de votre ordinateur (utilisez le drapeau -h avec agetty) (équivalent à "stty crtscts").

11.2.2 Brochage d'un câble null modem standard (7 connecteurs)

Les 3 diagrammes suivants montrent des câbles null modem "standards" complets. Si vous en achetez un, il risque d'être câblé de cette manière. Ils fonctionneront pour les terminaux en utilisant le contrôle de flux logiciel (Xon/Xoff), ou sans contrôle de flux. Cependant, ils ne fonctionnent pas pour le contrôle de flux matériel de terminal puisque la plupart des vrais terminaux supportent le contrôle de flux DTR ou DTR/DSR (prise de contact) mais que Linux ne le fait pas.

```
PC DB25 mâle                              Terminal DB25
TxD   Transmit Data          2 --> 3      RxD   Receive Data
```

```
RxD   Receive Data          3 <-- 2     TxD   Transmit Data
RTS   Request To Send       4 --> 5     CTS   Clear To Send
CTS   Clear To Send         5 <-- 4     RTS   Request To Send
DSR   Data Set Ready        6
                            |
DCD   Carrier Detect        8 <-- 20    DTR   Data Terminal Ready
SG    Signal Ground         7 --- 7     SG    Signal Ground
                                  6     DSR   Data Set Ready
                                  |
DTR   Data Terminal Ready  20 --> 8     DCD   Carrier Detect
```

Autrement, voici un câble null modem DB9-DB25 (ne fonctionnera pas avec la prise de contact matérielle des terminaux ; voir ci-dessus) :

```
PC DB9                                  Terminal DB25
RxD   Receive Data          2 <-- 2     TxD   Transmit Data
TxD   Transmit Data         3 --> 3     RxD   Receive Data
                                  6     DSR   Data Set Ready
                                  |
DTR   Data Terminal Ready   4 --> 8     DCD   Carrier Detect
GND   Signal Ground         5 --- 7     GND   Signal Ground
DCD   Carrier Detect        1
                            |
DSR   Data Set Ready        6 <-- 20    DTR   Data Terminal Ready
RTS   Request To Send       7 --> 5     CTS   Clear To Send
CTS   Clear To Send         8 <-- 4     RTS   Request To Send
RI    Ring Indicator        9 (pas nécessaire)
```

(Oui, les broches 2 et 3 *ont vraiment* des significations opposées dans les connecteurs DB9 et DB25 !)

Voici comment connecter 2 DB9 ensemble en null-modem (mais le flux de control DTR ne marchera pas) :

```
PC DB9                                  Terminal DB25
RxD   Receive Data          2 <-- 3     TxD   Transmit Data
TxD   Transmit Data         3 --> 2     RxD   Receive Data
                                  6     DSR   Data Set Ready
                                  |
DTR   Data Terminal Ready   4 --> 1     DCD   Carrier Detect
GND   Signal Ground         5 --- 5     GND   Signal Ground
DCD   Carrier Detect        1
                            |
DSR   Data Set Ready        6 <-- 4     DTR   Data Terminal Ready
RTS   Request To Send       7 --> 8     CTS   Clear To Send
CTS   Clear To Send         8 <-- 7     RTS   Request To Send
RI    Ring Indicator        9 (pas nécessaire)
```

L'utilisation des deux connexions ci-dessus fournit des signaux de contrôle de modem complets et de même nous permettent de positionner "stty -clocal". On doit alors allumer le terminal en premier (pour lancer DTR) avant de pouvoir ouvrir le port de manière normale par getty, etc. Mais il risque d'y avoir des problèmes si vous n'allumez pas le terminal en premier (voyez 17.5 (getty se relance trop rapidement)). Pour cette raison, on devrait utiliser "stty clocal" qui est la valeur par défaut (ignore les lignes de contrôle du modem) et les fils supplémentaires dans ces câbles ne servent alors à rien d'utile.

Dans les jours anciens où il n'était pas si facile d'ignorer les signaux de contrôle du modem, etc., on utilisait "l'astuce" suivante pour les câbles qui n'avaient pas les fils pour le contrôle de modem : du côté ordinateur du connecteur, on reliait RTS et CTS ensemble, et on connectait aussi DSR, DCD et DTR ensemble. De cette manière, quand l'ordinateur avait besoin d'un certain signal de prise de contact pour continuer, il l'obtenait (par erreur) de lui-même.

11.2.3 Limitations de longueur

Un câble de plus de 15 mètres environ (50 pieds) peut ne pas fonctionner correctement à grande vitesse. Des longueurs bien plus grandes fonctionnent parfois correctement, surtout si la vitesse est basse et/ou le câble est de type basse capacitance et/ou l'électronique de la partie réception est très sensible. On dit que sous certaines conditions idéales à 9600 bauds, un câble de 1000 pieds (300 mètres) fonctionne correctement. Une manière de couvrir de longues distances est d'installer un pilote de ligne près de chaque port série afin de convertir des signaux asymétriques en symétriques (et inversement) et d'utiliser ensuite du câblage en paire torsadée. Mais les pilotes de lignes sont chers.

11.2.4 Câbles pour le contrôle de flux matériel

Si vous voulez faire du contrôle de flux matériel (prise de contact) vous aurez certainement besoin de fabriquer votre propre câble (ou de le faire faire). Bien sûr, si les connecteurs à chaque extrémité d'un câble usagé s'enlèvent, vous pouvez le reconnecter. Voyez 11.6.3 (installation de connecteurs DB). Vous devrez déterminer si oui ou non le terminal utilise la broche DTR pour cela, et sinon, quelle(s) broche(s) il utilise. Les menus de configuration peuvent vous donner des indices là-dessus puisqu'il peut y avoir une option pour activer la "prise de contact DTR" (ou le contrôle de flux), ce qui implique bien sûr qu'il utilise la broche DTR. Il peut aussi utiliser la broche DSR. Voyez 10.7 (contrôle de flux matériel) pour une explication détaillée. Des terminaux anciens peuvent ne rien fournir pour faire du contrôle de flux matériel.

11.2.5 Astuces sur les câbles

Un câble "droit" normal ne fonctionnera pas sauf si vous l'utilisez comme câble d'extension couplé soit à un câble null modem, soit à un adaptateur null modem. Assurez-vous que les connecteurs au bout du câble se brancheront dans les connecteurs matériels. On peut utiliser du câble de téléphone qui a au moins 4 conducteurs (et peut-être en paire torsadée). Un câble d'ordinateur spécial de basse capacitance, blindé, est mieux.

11.2.6 Une bidouille qui utilise un câble en paire torsadée

Bien qu'aucun signal EIA-232 ne soit stabilisé pour de la paire torsadée on peut tenter d'utiliser un câble en paire torsadée. Utilisez une paire pour la transmission et l'autre pour la réception. Pour faire ceci connectez le signal de terre à un fil sur chacune de ces deux paires. Une partie seulement du signal de terre passe dans le fil désiré mais ça peut aider. À cause de l'inductance plus faible du circuit en paire torsadée (comparée au courant de retour de masse par un autre chemin), un peu plus de courant de retour (terre) se massera sur le chemin désiré que ce qu'on pourrait attendre des seuls calculs de résistances. Ceci est surtout vrai à des fréquences plus élevées puisque l'impédance d'induction augmente avec la fréquence. La courbe rectangulaire du port série contient des harmoniques de haute fréquence.

11.2.7 Mise à la terre du câble

La broche 1 (d'une prise DB25) devrait être reliée à la masse du châssis (qui est aussi la masse de la terre) mais sur les ports série économiques il peut n'être connecté à rien du tout. Un connecteur 9 broches n'est même pas relié à la masse du châssis. La masse du signal est la broche 7 et est en général reliée à la masse du châssis. Cela veut dire qu'une partie du courant du signal passera dans les fils de masse de l'installation électrique du bâtiment (indésirable). Les blindages de câbles sont sensés n'être mis à la terre qu'à une seule extrémité du câble, mais il peut être mieux de relier les deux extrémités à la terre puisqu'il est mieux d'avoir du courant dans le blindage que dans l'installation électrique du bâtiment ??

11.3 Connexion sur un modem

En utilisant une combinaison terminal-modem (sans ordinateur) on peut se connecter à des BBS. Certains BBS (comme free-nets) permettent un accès à Internet grâce au navigateur texte lynx qui fonctionnera sur les terminaux texte. Ainsi avec un vieux terminal et un modem externe, on peut se connecter à Internet. Si on se connecte à un ordinateur hôte sur lequel on a un compte, on peut parfois y stocker ses fichiers (ou ses téléchargements).

11.3.1 Appeler à l'extérieur à partir d'un terminal

Au lieu de relier un terminal (ou un ordinateur émulant un terminal) directement à un ordinateur hôte en utilisant un câble, il peut être relié à l'hôte par l'intermédiaire d'une ligne téléphonique (ou d'une liaison spécialisée dédiée) avec un modem à chaque bout de la ligne. Le terminal (ou l'ordinateur) appellera en général un ordinateur hôte avec une ligne téléphonique.

La plupart des gens utilisent un PC et un modem pour appeler vers l'extérieur. Le PC peut posséder un terminal relié à un port série et la personne sur le terminal peut appeler par l'intermédiaire du PC. La connexion d'un vrai terminal directement sur le modem externe est plus difficile car le vrai terminal n'est pas très intelligent et ne donne pas autant de retour à l'utilisateur. Pour appeler à l'extérieur, beaucoup de terminaux peuvent stocker un ou plusieurs numéros de téléphone comme des messages qu'on peut leur "paramétrer". Ces numéros sont envoyés au modem en pressant certaines touches de fonction. Beaucoup de modems peuvent aussi stocker des numéros de téléphone. La séquence d'initialisation du modem doit précéder le numéro de téléphone. Quande un autre modem répond à un appel sortant a l'autre bout de la ligne téléphonique, l'ordinateur hote sur ceux modem doit lancer un programme getty pour vous puissiez vous connectez.

11.3.2 On peut appeler un terminal

Il est courant pour un ordinateur faisant tourner Linux de se faire appeler par téléphone. L'appelant obtient une invite de login et se logge. À première vue, il peut sembler étrange qu'un terminal simple (relié à aucun ordinateur) puisse accepter un appel entrant, mais il le peut. Une des raisons possibles de faire ainsi est d'économiser les factures de téléphone quand les tarifs ne sont pas symétriques. Votre terminal doit être configuré pour un appel entrant : mettez le modem derrière votre terminal en mode réponse automatique (auto answer, registre S0 mis à 2 répondra à la deuxième sonnerie). Vous allumez le terminal et le modem avant d'attendre un appel et quand l'appel arrive vous obtenez une invite de login et vous vous loggez.

L'ordinateur hôte qui appelle votre terminal doit faire quelque chose d'inhabituel. Aussitôt que votre modem répond, il doit lancer login (getty). Un hôte peut faire ceci en lançant le programme Linux "callback", parfois appelé "cb". Le callback (ou rappel) c'est l'ordinateur A qui appelle l'ordinateur B, B raccroche et rappelle A. C'est ce que vous voulez si vous utilisez l'ordinateur A pour émuler un terminal. Dans le cas d'un vrai terminal cette tâche peut être trop complexe et par conséquent l'hôte n'utilise que la partie retour du programme callback. Le fichier de configuration de callback doit être configuré correctement sur l'hôte. Callback appelle le terminal et ensuite dit à mgetty de lancer un login sur ce port. Mgetty lui-même (au début 1998) ne sait gérer que les appels entrants mais on travaille sur l'incorporation de possibilités de rappel automatique et le rendre ainsi capable de gérer les appels sortants. Au début de 1999, cela ne semblait pas avoir été réalisé.

11.4 Connexion à un serveur de terminaux

Une utilisation possible est de connecter beaucoup de terminaux (ou des modems) à un réseau à haut débit qui permet la connexion à des ordinateurs hôtes. Bien sûr le serveur de terminaux doit avoir la puissance de calcul et les logiciels pour faire tourner des protocoles réseau, il ressemble donc à un ordinateur. Le serveur de terminaux peut interagir avec l'utilisateur et demander sur quel ordinateur il veut se connecter, etc. ou il peut le connecter sans rien demander. On peut parfois envoyer des travaux à une imprimante à travers un serveur de terminaux.

Un PC de nos jours a suffisamment de puissance de calcul pour agir comme un serveur de terminaux pour des terminaux texte sauf que chaque port série devrait avoir sa propre interruption matérielle. Les PC ne possèdent que peu d'interruptions libres à cette fin et puisqu'elles sont configurées en dur on ne peut pas en créer davantage par logiciel. Une solution est d'utiliser une carte série multiport avancée qui possède son propre système d'interruptions (ou sur les modèles de plus bas prix, partage l'une des interruptions du PC entre un certain nombre de ports). Voyez le HOWTO Serial pour plus d'informations sur de telles cartes. Si un tel PC fait tourner Linux avec getty tournant sur beaucoup de ports série on peut le considérer comme un serveur de terminaux s'il est relié à d'autres PC sur un réseau et si son travail est principalement de transférer les données et de gérer les interruptions du port série tous les 14 octets (ou à peu près). On utilise parfois un logiciel appelé "radius".

De nos jours les vrais serveurs de terminaux servent plus que de simples terminaux. Ils servent aussi des PC qui émulent des terminaux, et sont parfois reliés à des racks de modems reliés à des lignes téléphoniques. Certains contiennent parfois des modems intégrés. Si un terminal (ou un PC en émulant un) est relié directement à un modem, le modem à l'autre bout de la ligne devrait être relié à un serveur de terminaux. Dans certains cas le serveur de terminaux par défaut s'attend à ce que l'appelant utilise des paquets PPP, quelque chose que les vrais terminaux texte ne génèrent pas.

11.5 Types de connecteurs et d'adaptateurs

Un connecteur est plus ou moins relié de façon permanente à l'extrémité d'un câble ou à une unité matérielle. Il y a deux types de connexions de base : 1. DBxx avec des broches et 2. des connecteurs modulaires de type téléphone.

Un adaptateur ressemble à peu près à un connecteur mais il possède deux extrémités. C'est comme un câble qui est si court qu'il ne reste plus de partie câblée du tout – il ne reste que des connecteurs différents sur chaque extrémité. L'adaptateur se branche de chaque côté. Il permet de relier entre eux deux connecteurs incompatibles en s'interposant entre les deux. Quelquefois le but de l'adaptateur est d'interchanger les fils. Évidemment, on peut utiliser un câble spécial (éventuellement fait maison) pour remplacer pour cet adaptateur.

11.5.1 Sexe des connecteurs / adaptateurs

Les connecteurs (ou un côté des adaptateurs) sont soit mâles soit femelles. Les connecteurs qui ont des broches sont mâles et ceux qui ont des "trous" sont femelles. Pour les connecteurs modulaires, ceux qui ont les contacts visibles sont les fiches tandis que ceux qui ont les contacts à l'intérieur (pas facile à voir) sont les prises. Les fiches sont mâles ; les prises sont femelles (NdT : mais en français on dit aussi prise pour un connecteur mâle).

11.5.2 Types d'adaptateurs

Il y a trois type d'adaptateurs de base : les null modem, les changeurs de genre et les adaptateurs de ports. Certains adaptateurs effectuent plus d'une fonction parmi ces trois fonctions.

- adaptateur null modem : croise certaines connexions, comme un câble null modem.
- changeur de genre : change le sexe d'une extrémité d'un câble. Deux connecteurs du même sexe peuvent maintenant être reliés l'un à l'autre.
- adaptateur de port : va d'un type de connecteur à un autre (DB9 vers DB25, etc.).

11.5.3 Connecteurs DB

Pour savoir comment installer un connecteur DB à l'extrémité d'un câble, voyez 11.6.3 (installer des connecteurs DB). Ceux-ci sont disponibles en 9 ou 25 broches. Les spécifications EIA-232 demandent des 25 broches mais puisque la plupart de ces broches ne sont pas utilisées sur les ports série ordinaires, 9 broches sont suffisantes. Voyez 11.2.2 (DB9-DB25) pour la signification des brochages. Les broches sont en général numérotées si vous regardez de suffisamment près ou si vous utilisez une loupe grossissante.

11.5.4 Connecteurs modulaires RJ

Ceux-ci ressemblent à des connecteurs de téléphone modernes mais ne sont parfois pas compatibles avec les connecteurs téléphoniques. Voyez aussi 11.6.4 (installer des connecteurs RJ). Il peut y avoir 6, 8 ou 10 conducteurs. RJ11/14 est une fiche de téléphone à 4-6 conducteurs. Une fiche qui lui ressemble est un connecteur MMJ (6 conducteurs) utilisé sur certains modèles tardifs de terminaux VT (et autres). MMJ possède une languette décalée et n'est pas compatible avec RJ11/14. Cependant, certains connecteurs ont été fabriqués et sont compatibles à la fois avec MMJ et RJ11/14. Le brochage MMJ est : 1-DTR, 2-TXD, 3-TXD GND, 4-RXD GND, 5-RXD, 6-DSR.

Un câble null modem avec des connecteurs MMJ (ou RJ11/14) reliera : 1-6, 2-5 et 3-4. Notez qu'un tel câble supporte le contrôle de flux DTR/DSR qui n'est pas (encore) supporté par Linux. Faire vous-même votre propre câble null modem à 6 conducteurs est très simple si vous comprenez que le câble téléphonique à 4 conducteurs de votre mur à votre téléphone, utilisé dans des centaines de millions d'habitations, est aussi un câble null modem. Trouvez-en un et câblez votre câble de la même façon.

Si vous étalez un tel câble (ou le câble null modem de votre terminal) à plat sur le sol (sans torsion) vous noterez que les deux fiches aux extrémités ont leurs contacts dorés dirigés tous les deux vers le haut (ou tous les deux vers le bas). Bien que ce soit symétrique, c'est aussi null modem si vous y réfléchissez un peu. On pourrait associer un petit nombre de ces câbles avec des coupleurs en ligne et tout fonctionnera correctement car chaque coupleur en ligne est aussi un adaptateur null modem. Deux périphériques null modem en série donnent une connexion directe.

RJ45 et RJ48 sont des fiches téléphoniques à 8 conducteurs (ou 10 conducteurs pour certaines applications informatiques). Ceux à 10 conducteurs sont légèrement plus larges et ne rentreront pas dans des fiches à 8 conducteurs. On les utilise à la fois pour des câbles téléphoniques plats et des câbles à paire torsadée ronds. L'extrémité du connecteur côté câble peut être différent pour les câbles plats et ronds et à la fois RJ45 et RJ48 peuvent avoir 8 ou 10 conducteurs, donc assurez-vous que vous choisissez le bon. RJ48 possède un taquet supplémentaire ce qui fait qu'une fiche RJ48 ne rentrera pas dans une prise RJ45 (mais une fiche

RJ45 rentrera dans une prise RJ48). On les utilise sur certaines cartes série multiports et dans les réseaux. Voici les numéros de broches pour une prise à 8 conducteurs :

```
Fiche                          Prise
(En regardant à                (En regardant la
 l'extrémité d'un câble)        cavité dans un mur)
   .__________.                   .__________.
   | 87654321 |                   | 12345678 |
   |__.    .__|                   |__.    .__|
      |____|                         |____|
```

11.6 Fabriquer ou modifier un câble

11.6.1 Acheter ou fabriquer ?

Vous pouvez essayer d'acheter un câble null modem court. On les étiquette souvent comme des câbles d'imprimantes série (mais les imprimantes série ne sont pas très en vogue de nos jours, ni les câbles associés). Malheureusement, ils ne fonctionneront probablement pas pour le contrôle de flux matériel. Assurez-vous que les connecteurs sur les extrémités du câble conviendront aux connecteurs sur votre ordinateur et votre terminal.

Mais si vous avez besoin de câbles plus longs pour relier des terminaux ou si vous avez besoin du contrôle de flux matériel, comment obtenez-vous les bons câbles ? Les bons câbles longs tout prêts peuvent s'avérer difficiles à trouver (vous pouvez tenter de les chercher sur Internet), surtout si vous voulez utiliser un minimum (disons 4) de conducteurs. Une option est de les faire faire spécialement, ce qui risque d'être relativement cher bien que vous puissiez trouver quelqu'un pour le faire à des prix à peine plus élevés que ceux tout prêts (c'est ce que j'ai fait). Une autre alternative est de fabriquer le vôtre. Ceci peut nécessiter des outils spéciaux. Si vous pouvez en trouver, du câble d'occasion peut s'avérer faire une bonne affaire mais vous devrez probablement recâbler les connecteurs. La plupart des connecteurs livrés avec les câbles courts sont moulés de manière permanente sur le câble et ne peuvent pas être recâblés mais la plupart des câbles faits sur mesure ou à la main possèdent des connecteurs qu'on peut recâbler. L'avantage de fabriquer votre propre câble est que les compétences que vous acquérerez seront utiles si un câble se casse (ou devient mauvais) ou si vous devez fabriquer un autre câble en vitesse.

11.6.2 Numéros de broches

Les numéros des broches devraient être gravées sur la partie plastique du connecteur. Chaque broche devrait avoir un numéro juste à côté. Vous aurez peut-être besoin d'une loupe grossissante pour les lire.

11.6.3 Installer des connecteurs DB sur les extrémités des câbles

Voyez 11.5.3 (connecteurs DB) pour une courte description de ces connecteurs. Malheureusement, la plupart des câbles achetés de nos jours possèdent des connecteurs moulés à chaque extrémité qu'on ne peut pas modifier. Si vous fabriquez des câbles ou en modifiez un existant, vous devrez apprendre à connaître les broches. Il y en a deux sortes : soudées et serties.

Les broches serties nécessitent un outil de sertissage spécial et aussi un outil "d'insertion/extraction". Mais une fois que vous avez les outils, fabriquer et modifier un câble est plus rapide que de les souder. Si vous reliez deux fils à la même broche (nécessaire aussi si vous voulez relier une broche connectée à une autre broche) alors la soudure est plus rapide (pour ces broches). C'est dû au fait que les broches serties ne peuvent prendre qu'un fil à la fois, alors que les broches soudées peuvent accepter plus d'un fil par broche.

Pour insérer des broches serties, poussez-les simplement à la main ou avec l'outil. Enlever une broche avec l'outil est un peu difficile. Ces instructions seront mieux compréhensibles si vous avez à la fois l'outil et les fils devant vous. Vous devez mettre l'embout de l'outil autour de l'arrière de la broche et l'insérer aussi profond que possible dans le trou (environ 1 cm 1/2.). Certains outils ont une marque (comme un tout petit trou) pour indiquer jusqu'où l'insérer. L'embout de l'outil devrait avoir un creux taillé en pointe pour que vous puissiez placer l'embout autour du fil en le mettant là où le creux est plus large que le fil. L'outil peut avoir deux embouts. Celui qui est le plus difficile à mettre autour du fil est aussi celui qui enlève le fil le plus facilement puisqu'il enveloppe presque complètement le fil.

Avec l'embout proprement inséré, tirez à la fois doucement à la fois sur l'outil et sur le fil. Si ça ne vient pas, l'outil n'a sûrement pas été inséré correctement, poussez-le alors plus en avant ou bien tournez-le dans une position différente (ou les deux). Vous auriez peut-être dû utiliser l'autre embout qui entoure mieux la broche. En utilisant cet outil, on peut facilement convertir un câble droit en câble null modem, etc.

Vous pouvez avoir des problèmes en utilisant l'outil d'"insertion/extraction". Si les outils ne s'insèrent pas sur le derrière de la broche, il se peut que la broche n'ait pas été sertie correctement sur le fil et tienne plus du carré que du rond, etc. Si une broche commence à sortir mais ne se redresse pas en entier, la broche peut être tordue. Regardez-la sous une loupe grossissante. Redresser une broche avec une pince à épiler peut endommager le plaquage en or. Parfois une broche coincée peut être poussée au dehors avec le bout d'une lame de tournevis épaisse (ou quelque chose de semblable) mais si vous poussez trop fort vous pouvez agrandir le trou en plastique ou tordre la broche.

N'essayez pas de souder sauf si vous savez ce que vous faites ou avez lu comment le faire.

11.6.4 Installer des connecteurs RJ

Ce sont des connecteurs modulaires de téléphones, dont une sorte est utilisée pour la plupart des téléphones ordinaires. Mais il y a beaucoup de sortes différentes (voyez 11.5.4 (connecteurs modulaires RJ)).

Ils ne sont pas faciles à réutiliser. Vous pourriez retirer les fils, pousser à l'intérieur un outil pointu qui relèverait les contacts dorés et réutiliser le connecteur. Il y a des outils de sertissage spéciaux qu'on utilise pour les installer ; un outil différent pour chaque sorte.

Si vous n'avez pas d'outil de sertissage, l'installation est quand même possible (mais difficile) en utilisant un petit tournevis (et peut-être un marteau). Poussez les fils du câble à l'intérieur et ensuite poussez chaque contact doré vers le bas assez fort avec le petit tournevis qui passera juste entre les stries isolantes entre les contacts. Vous pouvez l'abîmer si vous n'utilisez pas un tournevis avec une tête qui a presque la même épaisseur que les contacts ou si le tournevis glisse du contact alors que vous le poussez vers le bas. Vous pouvez aussi utiliser un petit marteau pour faire contrepoids sur le tournevis (poussez d'abord à la main).

Assurez-vous de ne pas abîmer le "levier d'extraction" sur le connecteur quand vous poussez sur les contacts. Ne le posez pas simplement sur une table en poussant les contacts. Il est préférable de mettre une cale (d'environ 1 mm d'épaisseur) qui se placera parfaitement dans la crevasse entre le levier et le corps. Pour une telle cale vous pouvez utiliser du bristol épais, plusieurs cartes téléphoniques ou du bois. Puisque le dessous du connecteur (que vous mettrez sur la table) n'est pas droit (à cause du "levier d'extraction"), assurez-vous que le dessus de la table est recouvert de quelque chose de mou (comme un morceau de carton) pour aider à supporter le connecteur qui n'est pas droit. Encore mieux : vous pourriez mettre une autre cale d'un millimètre sous les 6 premiers millimètres du connecteur en le supportant juste en dessous de la partie visible des contacts. Un dessus de table mou ne peut pas faire de mal non plus. Une autre méthode (je n'ai jamais fait cela) est de mettre le connecteur dans un étau mais faites attention à ne pas casser le connecteur.

En comparaison de l'utilisation d'un outil de sertissage, l'installation comme indiqué ci-dessus prend beaucoup plus de temps et est plus sujette aux erreurs mais c'est quelquefois plus expéditif et bien moins cher que d'acheter un outil spécial si vous n'avez qu'un ou deux connecteurs à installer.

12 Mise en place (configuration) en général

12.1 Introduction à la configuration

La configuration (mise en place) nécessite à la fois de stocker la configuration dans la mémoire non volatile du terminal et de mettre des commandes dans les fichiers de démarrage (sur votre disque dur) qui se lanceront à chaque fois que l'ordinateur est allumé (ou peut-être aussi à chaque changement de niveau d'exécution). Cette section donne une vue d'ensemble de la configuration et couvre la configuration des principales options de communication à la fois pour le terminal et pour l'ordinateur. Les deux grandes sections suivantes couvrent la configuration du terminal (voyez 13 (mise en place du terminal) et de l'ordinateur (voyez 14 (détails de la mise en place de l'ordinateur (configuration)).

12.2 Vue d'ensemble de la mise en place (configuration) du terminal

Quand un terminal est installé il est nécessaire de configurer le terminal physique en sauvant (dans sa mémoire non volatile qui n'est pas perdue lorsque l'on éteint le terminal) les caractéristiques qu'il aura au démarrage. Vous pouvez avoir de la chance et avoir un terminal qu'on a déjà configuré correctement pour votre installation de sorte que peu ou aucune configuration sur le terminal ne soit nécessaire.

Il y a deux façons simples de configurer un terminal. L'une est de s'asseoir devant le terminal et de parcourir un ensemble de menus de configuration. Une autre est d'envoyer des séquences d'échappement au terminal à partir de l'ordinateur hôte. Avant que vous n'envoyiez quoi que ce soit au terminal (comme les séquences d'échappement ci-dessus), ses options d'12.5 (interface de communication) comme la vitesse de transmission doivent être configurées pour correspondre à celles de l'ordinateur. On ne peut faire ceci qu'en s'asseyant devant le terminal puisqu'il faut configurer la communication correctement avant que l'ordinateur et le terminal ne puissent se "parler". Voyez 13 (configuration du terminal).

12.3 Vue d'ensemble de la mise en place (configuration) de l'ordinateur

À part peut-être l'envoi de séquences d'échappement à partir de l'ordinateur pour configurer le terminal, il faut configurer l'ordinateur lui-même pour gérer le terminal. Si vous avez de la chance tout ce que vous avez à faire est d'ajouter une commande "getty" dans le fichier /etc/inittab pour qu'une invite "login:" soit envoyée au terminal quand l'ordinateur démarre. Voyez 14.1 (getty (dans /etc/inittab)) pour plus de détails.

L'ordinateur communique avec le terminal en utilisant un logiciel de pilote de périphérique (qui fait partie du noyau). Le pilote de périphérique série possède une configuration par défaut et est aussi configuré en partie (parfois en totalité) par le programme getty avant de lancer "login" sur chaque terminal. Cependant, des configurations supplémentaires sont souvent nécessaires en utilisant des programmes appelés "stty" et "setserial". Ces programmes (si besoin est) doivent être lancés à chaque fois que l'ordinateur démarre puisque la configuration est perdue à chaque fois que l'ordinateur s'éteint. Voyez 14 (détails de la mise en place de l'ordinateur (configuration)).

12.4 Beaucoup d'options

Il y a énormément d'options de configuration dans lesquelles vous pourrez piocher. Les options de communication doivent être correctes ou le terminal ne fonctionnera pas du tout. D'autres options peuvent être incorrectes, mais ne causeront pas de problèmes puisque les possibilités qu'elles introduisent ne sont pas utilisées. Par exemple, si vous n'avez pas d'imprimante reliée au terminal, la manière dont sont positionnés les paramètres de configuration de l'imprimante dans le terminal n'a aucun effet. Cette dernière affirmation n'est pas correcte à 100 %. Supposez que vous n'avez pas d'imprimante mais que l'ordinateur envoie (par erreur) au terminal une commande pour rediriger tous les caractères (les données) de l'ordinateur vers l'imprimante uniquement. Alors rien ne s'affichera sur l'écran et votre terminal sera mort. Certains terminaux possèdent une option de configuration pour informer le terminal qu'aucune imprimante n'est présente. Dans ce cas le terminal ignorera toute commande qui redirigerait la sortie vers "l'imprimante" et le problème ci-dessus n'arrivera jamais. Cependant, ceci n'apporte pas beaucoup d'aide puisqu'il y a beaucoup d'autres commandes erronnées qu'on peut envoyer à votre terminal et qui sèmeront vraiment la pagaille. Cela arrivera sûrement si vous envoyez un fichier binaire au terminal par accident.

Dans certains cas un paramètrage incorrect ne causera aucun problème jusqu'à ce qu'il vous arrive de lancer une application peu connue qui attend du paramètre qu'il soit positionné d'une certaine façon. Certaines options ne concernent que l'apparence de l'affichage et le terminal fonctionnera correctement si elles sont mal positionnées mais ne sera pas aussi beau à regarder.

Certaines options ne concernent que le terminal et n'ont pas besoin d'être positionnées sur l'ordinateur. Par exemple : voulez-vous des lettres noires sur un fond clair ? C'est plus agréable à l'oeil qu'un fond noir. Une touche qu'on appuie doit-elle se répéter ? Est-ce que l'écran doit défiler quand une ligne dépasse le bord droit de l'écran ? Est-ce que les touches doivent émettre un cliquetis ?

12.5 Options de l'interface de communication

Certains paramètres (options) de communication concernent à la fois le terminal et l'ordinateur et doivent être positionnés exactement à l'identique sur les deux : vitesse, parité, bits/caractères et contrôle de flux. D'autres options de communication ne sont positionnées que sur le terminal (et uniquement certaines d'entre elles sont essentielles pour établir une communication). D'autres encore comme l'adresse et l'interruption (IRQ) du port physique ttyS ne sont positionnées que sur l'ordinateur en utilisant la commande "setserial". Jusqu'à ce que toutes les options essentielles ci-dessus soient paramétrées de manière compatible on ne pourra pas faire de communication série satisfaisante (et sûrement pas de communication du tout) entre le terminal et l'ordinateur. Pour le terminal, on doit positionner ces options à la main avec des menus sur chaque terminal (ou en utilisant une sorte de cartouche spéciale sur chaque terminal). L'ordinateur hôte est configuré en lançant des commandes à chaque démarrage de l'ordinateur (ou quand les gens se loggent). Quelquefois le programme getty (qu'on trouve dans le fichier /etc/inittab) qui démarre le processus login s'occupera de ceci pour l'ordinateur. Voyez 14.1 (getty (dans /etc/inittab)).

Les paramètres à la fois pour l'ordinateur et le terminal sont :

- 12.5.1 (vitesse (bits/seconde))
- 12.5.2 (parité)
- 12.5.3 (bits par caractère)
- 10 (contrôle de flux)

Certains paramètres essentiels au terminal seul sont :

- 12.5.5 (sélection du port)
- Positionner la communication en full duplex (=FDX sur les terminaux Wyse)

Si le programme 14.1 (getty (dans /etc/inittab)) ne peut pas paramétrer la partie côté ordinateur de la manière dont vous le désirez, vous aurez alors besoin d'utiliser l'une des commandes 14.2 (stty et setserial) (ou les deux).

12.5.1 Vitesse

Elle doit être identique à la fois sur le terminal et l'ordinateur. La vitesse est en bits/seconde (bps ou vitesse en bauds). Utilisez la vitesse la plus grande qui fonctionne sans erreurs. Autoriser le contrôle de flux peut rendre possible des vitesses plus élevées. Il se peut qu'on puisse configurer deux vitesses sur le terminal : transmission et réception, qu'on abrège parfois en T et R. On leur donne en général la même valeur puisque stty dans Linux ne semble pas encore avoir l'option de les positionner différemment. (Il y a une option pour faire ceci avec la commande "stty" mais en réalité il semble qu'elle leur donne la même valeur.) Les vitesses courantes sont 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 ... Les vitesses basses (comme 600) sont faites pour les imprimantes et les terminaux à sortie papier.

12.5.2 La parité et devriez-vous l'utiliser ?

Pour une définition, voyez 21.3 (la parité expliquée). La parité désactivée est souvent la valeur par défaut. Pour activer la parité, vous devez à la fois l'activer et ensuite sélectionner une parité paire ou impaire. Cela ne fait probablement pas de différence si c'est pair ou impair. Pour les terminaux il y a parfois des paramètres de parité pour la transmission et la réception. Vous devriez positionner les deux à la même valeur puisque stty sur l'ordinateur ne permet pas un paramétrage différent. Le port série du PC ne peut pas habituellement supporter differentes parités. Certains terminaux ne sont pas capables de positionner la parité de réception et ignoreront simplement tout bit de parité reçu. Sur certains terminaux plus anciens, si vous utilisez 8 bits de données par octet, la parité ne fonctionnera pas puisqu'il n'y a pas de place au niveau matériel pour le bit de parité supplémentaire.

Est-ce vous devriez utiliser la parité ? Il est bon de disposer de la parité, bien qu'elle ne soit pas strictement nécessaire. Si vous n'avez pas de parité, vous pourrez avoir une lettre incorrecte ici ou là et finirez par corriger des fautes d'orthographe qui n'existent pas vraiment. Cependant, la parité a un coût. D'abord, elle est plus compliquée à configurer puisqu'en général par défaut il n'y a pas de parité. Ensuite, la parité diminuera la vitesse avec laquelle les octets traversent le câble puisqu'il y aura un bit supplémentaire par octet. Ceci peut ou peut ne pas diminuer la vitesse effective.

Par exemple, un terminal à sortie papier est en général limité par la mécanique du processus d'impression. L'augmentation des octets/seconde quand l'ordinateur (sa puce UART) est en train de transmettre ne conduit qu'à plus de signaux "d'arrêt" de contrôle de flux pour permettre au mécanisme d'impression de rattraper le rythme. À cause des attentes supplémentaires dues au contrôle de flux, la vitesse effective n'est pas meilleure sans parité qu'avec. La situation est identique pour certains terminaux : après avoir mis en place la parité il peut y avoir moins d'attente due au contrôle de flux par unité de temps ce qui résulte en plus de bits/seconde (en moyenne). Cependant, à cause des bits de parité ajoutés, le nombre d'octets/seconde (en moyenne) reste le même.

Une possibilité est d'installer les terminaux sans parité. Ensuite si on remarque des erreurs de parité, on peut l'activer plus tard. Pour détecter des erreurs possibles sans parité, regardez des erreurs de frappe que vous ne pensez pas avoir faites. Si vous détectez une telle erreur, rafraichissez l'écran (retransmettez de l'ordinateur). Si l'erreur s'en va, c'est alors sûrement une erreur de parité. Si trop d'erreurs surviennent (comme plus d'une par quelques centaines d'écrans), il faut faire une action corrective comme : activer la parité et/ou réduire la vitesse et/ou utiliser un câble plus court/meilleur. Activer la parité ne réduira pas le nombre d'erreurs mais vous signalera quand une erreur s'est produite.

La politique opposée est d'activer la parité dès le départ. Ensuite si on ne voit jamais d'erreur de parité (des symboles d'erreur sur l'écran), sur une période de temps raisonnable, disons un mois ou deux, on peut la désactiver sans danger.

12.5.3 Bits/caractère

C'est la taille d'un caractère (le nombre de bits de données par caractère, en excluant tout bit de parité). Pour l'ASCII, c'est 7, mais c'est 8 pour les jeux de caractères ISO. Si vous n'allez utiliser que des caractères ASCII, sélectionnez alors 7 bits puisqu'il est plus rapide de transmettre 7 bits que 8. Certains vieux terminaux n'afficheront que des caractères 7 bits.

12.5.4 Quel contrôle de flux (prise de contact) ?

Il faut choisir entre le contrôle de flux "matériel" (par exemple dtr/cts) et "logiciel" (Xon/Xoff). (le menu du terminal Adds utilise de manière incorrecte "Xon/Xoff" pour indiquer tout type de contrôle de flux.) Bien que le contrôle de flux matériel soit plus rapide (si les deux fils supplémentaires nécessaires sont présents dans le câble et si le terminal le supporte) dans la plupart des cas Xon/Xoff devrait fonctionner correctement. Certaines personnes disent qu'elles ont résolu des problèmes dérangeants (voir ci-dessous) en passant au contrôle de flux matériel mais le contrôle de flux logiciel a bien fonctionné sur les autres installations (et chez moi personnellement).

Si vous utilisez le contrôle de flux logiciel (Xon/Xoff) et avez des utilisateurs qui ne le savent pas, ils peuvent alors envoyer par accident un Xoff à l'hôte et bloquer leur terminal. Pendant qu'il est bloqué, ils peuvent très bien taper frénétiquement en une vaine tentative pour le débloquer. Quand alors Xon est enfin envoyé pour rétablir la communication, tout ce qui a été tapé à toute vitesse est exécuté, peut-être avec des résultats inattendus. Ils ne risquent pas ceci avec le contrôle de flux matériel. Voyez 10 (contrôle de flux) pour une explication sur le contrôle de flux.

12.5.5 Sélection du port

Puisque la plupart des terminaux possèdent deux connecteurs ou plus derrière eux, il est en général possible de dire que l'un de ces connecteurs sera relié à l'ordinateur et de dire que l'autre connecteur sera le port imprimante. Il peut y avoir un nom à côté du connecteur (regardez-le bien) et ce nom (comme Aux ou Serial 2, ou Modem) peut être donné à la connexion vers l'hôte principal ou vers l'imprimante (ou autre).

12.6 Essai rapide

Bien que tout ce qui est au-dessus puisse sembler très compliqué, faire fonctionner un terminal est souvent assez simple. La section 3 (installation rapide) décrit une manière simple d'essayer de le faire. Mais si cela ne fonctionne pas ou si vous voulez améliorer l'affichage, vous avez besoin d'en lire plus.

13 Détails de la mise en place (configuration) du terminal

À part la prochaine sous-section qui décrit l'envoi de séquences d'échappement au terminal, cette section présente principalement les détails de la configuration du terminal à la main en s'asseyant devant le terminal et en se déplaçant entre différents menus. Si vous ne l'avez pas encore fait, vous devriez lire 12.2 (vue d'ensemble de la mise en place (configuration) d'un terminal). C'est mieux si vous avez le manuel du terminal, mais même si vous ne l'avez pas il y a ici des informations concernant beaucoup d'options que vous serez peut-être amené à configurer.

Les paramètres de communication tels que la vitesse d'envoi doivent toujours être configurés sur le terminal puisque si vous ne le faites pas ici on ne peut pas communiquer avec le terminal. Une fois que la communication est établie vous avez deux choix pour effectuer le reste de la configuration du terminal. Vous pouvez continuer à faire la configuration à la main sur le terminal et sauver les résultats dans la mémoire permanente du terminal ou vous pouvez le faire en envoyant des séquences d'échappement au terminal à partir de l'ordinateur à chaque fois que le terminal est allumé.

Si vous savez comment mettre en place et sauver une bonne configuration dans le terminal ce peut être la meilleure façon. Si vous ne savez pas, vous pouvez simplement envoyer la chaîne d'initialisation de terminfo à votre terminal à chaque fois que vous utilisez le terminal. Peut-être que ne rien faire vous donnera quand même un terminal utilisable. Vous (ou une application) pouvez toujours changer les choses en envoyant certains séquences d'échappement au terminal.

13.1 Envoyer des séquences d'échappement au terminal

Une fois que l'interface de communication est en place, le reste de la configuration des terminaux peut parfois être fait en envoyant des séquences d'échappement aux terminaux depuis l'ordinateur. Si vous avez un grand nombre de terminaux, il peut être utile d'écrire (ou de trouver) un script shell pour faire ceci de manière automatique. Il peut (ou non) y avoir une commande à envoyer au terminal pour lui dire de sauver sa configuration en cours dans sa mémoire permanente pour qu'elle soit présente la prochaine fois qu'on allumera le terminal.

Il y a une manière simple d'envoyer ces séquences d'échappement et une manière compliquée. Avec la manière simple, vous ne regardez jamais les séquences d'échappement mais envoyez des commandes qui trouvent toutes seules la séquence d'échappement appropriée dans la base de données terminfo et l'envoient. Malheureusement, toutes les séquences d'échappement que vous avez besoin d'envoyer ne sont pas toujours dans la base de données terminfo. Ainsi la manière plus compliquée (mais peut-être meilleure) est d'envoyer directement les séquences d'échappement.

Pour cette méthode compliquée vous aurez besoin d'un manuel avancé. Les vieux manuels de terminaux comprenaient autrefois une liste détaillée de séquences d'échappement mais les manuels récents n'en ont généralement pas. Pour les trouver vous devrez peut-être acheter un autre manuel intitulé "manuel du programmeur" (ou similaire) qui n'est pas fourni avec le terminal. Une 20.1 (liste de séquences d'échappement) de certains terminaux est sur Internet mais c'est une liste brute et sûrement incomplète.

Même sans manuel ou autre, vous pouvez quand même envoyer des commandes pour configurer le terminal en utilisant les programmes "tput" et "setterm". Voyez 16.11 (changer les paramètres du terminal). Vous pourriez simplement envoyer une chaîne d'initialisation au terminal à partir de l'entrée terminfo si la chaîne d'initialisation configure le terminal comme vous le désirez. Voyez 15.5 (chaîne d'initialisation). Excepté si vous pensez envoyer ces séquences de l'ordinateur au terminal à chaque fois que vous allumez ce dernier, vous devez sauver ces paramètres dans la mémoire permanente du terminal.

13.2 Configuration des vieux terminaux

Sur les vieux terminaux, cherchez sur le clavier des noms juste au-dessus de la rangée supérieure du pavé numérique. S'ils existent, ces noms peuvent représenter l'action des touches en mode configuration. Certains vieux terminaux peuvent n'avoir qu'un menu de configuration (setup). Certains encore plus anciens possèdent des interrupteurs physiques. Dans certains cas toutes les touches ne sont pas bien nommées mais peuvent être bien cachées. Bien sûr, si vous activez quelque chose en basculant un interrupteur, c'est "sauvegardé" et vous n'avez pas besoin de sauver le paramètre en mémoire permanente.

13.3 Entrer dans le mode de configuration

Pour sélectionner des options (configurer) sur le terminal, vous devez d'abord entrer en mode "setup" et ensuite sélectionner des options (c'est-à-dire configurer) en utilisant des menus stockés dans le terminal et affichés sur l'écran. Pour ce faire, le terminal n'a même pas besoin d'être relié à un ordinateur. La manière d'entrer en mode de configuration est expliquée dans le manuel du terminal, mais voici quelques indices qui peuvent aider :

S'il y a une touche "setup", essayez de l'enfoncer. Essayez aussi en appuyant sur la touche "Shift" en même temps.

- Wyse : essayez d'abord la touche "Shift" + "Select" ; ensuite remplacez Shift par Ctrl dans tout ce qui est au-dessus.
- VT, Dorio : F3 peut être la touche de configuration. Sur les VT420 et modèles postérieurs cette touche peut avoir été reprogrammée pour faire autre chose, alors éteignez le terminal. Quand vous remettrez le courant, appuyez sur la touche F3 dès que vous obtenez un message initial à l'écran.
- IBM : 3151 : Ctrl-ScrollLock. 3153 : Ctrl-Moins_du_pavé_numérique (ou pareil que pour le 3151).

Pour vous déplacer dans les menus de configuration, utilisez les touches fléchées. Utilisez Retour, Espace ou une touche spéciale ("toggle" sur les vieux terminaux) pour sélectionner quelque chose. Pour sortir du mode de configuration choisissez exit dans un menu (ou sur certains vieux terminaux appuyez de nouveau sur la touche setup).

13.4 Options de communication

Pour que le terminal fonctionne, il faut paramétrer correctement la vitesse, la parité, le nombre de bits par caractère et le mode de communication. Un contrôle de flux incorrect peut être la cause de la perte ou de la corruption des données affichées à l'écran. On a parlé des options de communication essentielles (à la fois pour le terminal et pour l'ordinateur) dans une autre section : voyez 12.5 (interface de communication). La liste suivante fournit quelques liens dans cette section, ainsi que des options de communication supplémentaires positionnées uniquement sur le terminal.

- 12.5.1 (Vitesse (bits/seconde)) (vitesse de transmission) : 9600, 19200, etc.
- 12.5.2 (Parité) : aucune, paire, impaire, marque, espace
- 12.5.3 (Bits par caractère) {Données} : 7 ou 8
- 10 (Contrôle de flux) ou prise de contact {Hndshk} : aucun, Xon-Xoff, ou matériel (DTR, etc)
  - Prise de contact du récepteur {Rcv Hndshk} protège les données reçues par le terminal en envoyant des signaux de contrôle de flux à l'hôte.
  - Prise de contact de l'émetteur {Xmt Hndshk} est une protection des données transmises par le terminal. Le terminal reçoit des signaux de contrôle de flux (et bloque/débloque le clavier). Comprend "Xon/Xoff entrant".
- Nombre de bits d'arrêt : 1 ou 2. Voyez 21.2.2 (séquence de tension pour un octet)
- Niveau de contrôle de flux {Rcv Hndshk Level} {{Xoff à ...}} : le contrôle de flux enverra un "arrêt" quand ce nombre d'octets sera dépassé dans le tampon du terminal.
- 5.3 (Mode de communication) {Comm} : 5.3 (Full Duplex {FDX}, Half Duplex {HDX}) {{Local Echo}}, 17.12 (Mode Local) {{Online/Local}}
- Limite du taux de transmission (vitesse) {Xmt Lim} : limite le taux de transmission au nombre de cps (caractères par seconde) spécifié même si la vitesse est plus grande.
- Limite du taux de touches de fonction : comme ci-dessus mais pour les messages de touches de fonction.
- 12.5.5 (Sélection du port) : quel connecteur physique va à l'hôte {Host Port} ?

13.5 Sauver la configuration

Votre configuration doit être sauvée dans la mémoire permanente du terminal pour être en place la prochaine fois que vous allumez le terminal. Si vous manquez à cette sauvegarde, les nouveaux paramètres seront perdus quand vous éteindrez le terminal. Avant de vous fatiguer à configurer un terminal, assurez-vous que vous savez sauvegarder les paramètres. Sur les terminaux modernes la commande de sauvegarde se fait grâce à un menu. Dans certains vieux terminaux, seul le manuel vous indique comment sauver. Pour beaucoup d'entre eux, pressez Ctrl-S pour sauver.

13.6 Paramètres/options de configuration

Voyez la *page de configuration* `http://www.pericom-usa.com/twdocs/doc/twusec7.htm` de Teemworld pour avoir une description d'un grand nombre de ces paramètres utilisés en émulation de terminaux. L'émulation est souvent légèrement différente du terminal réel.

Ce qui suit dans cette section décrit certaines options disponibles dans les menus de configuration de beaucoup de terminaux. Les options sont aussi appelées paramètres ou capacités. Beaucoup d'options sont parfois appelées "modes". Paramétrer des options est souvent appelé "configurer". On peut paramétrer beaucoup de ces options en envoyant certaines séquences d'échappement au terminal. Des modèles et marques différents de terminaux ont des options variées et la même option peut avoir des noms différents (qu'on ne donnera pas tous ici). Les noms abrupts utilisés par Wyse sont entourés de {...}. Les noms utilisés principalement pour les terminaux VT sont entourés de {{...}}.

13.7 Émulation {Personnalité} {{Modes de terminaux}}

La plupart des terminaux modernes peuvent émuler plusieurs autres terminaux. Le terminal peut sûrement faire plus si on le fait s'émuler lui-même (en fait sans émulation) {personnalité native}. Quelque fois il y deux émulations différentes pour le même modèle de terminal. Par exemple, VT220-7 émule un VT220 avec 7 bits par octet alors que VT220-8 émule un VT220 avec 8 bits par octet (256 caractères possibles).

Des vieux modèles de terminaux possèdent en général moins de possibilités que les modèles plus récents. Supposez qu'on veuille émuler un vieux terminal tout en gardant certaines possibilités avancées du modèle de terminal plus récent devant lequel on s'assied. Ceci est parfois possible (jusqu'à un certain niveau). Cette possibilité est parfois appelée {Enhance} (ou Enhanced ??).

13.8 Options d'affichage

13.8.1 Taille de cellule de caractère{Char Cell}

C'est la taille de la cellule dans laquelle tient un caractère. On la mesure en pixels (= petits points). Plus il y a de points, meilleure est la résolution. 10x16 représente 10 points de large sur 16 points de haut (16 lignes et 10 colonnes). Notez que la notation est inversée par rapport à la notation de la dimension des matrices qui donnent les lignes (la hauteur) d'abord. De même, la cellule de caractère comprend les lignes et les colonnes allouées à l'espace entre des caractères adjacents, donc la taille de la cellule qui définit les bords d'un caractère réel peut être plus petite.

13.8.2 Colonnes / lignes

En général il y a 80 colonnes et 24 ou 25 lignes. Ceci veut dire qu'il peut y avoir jusqu'à 80 caractères sur une ligne de l'écran. Beaucoup de terminaux possèdent une option de 132 caractères par colonne mais, sauf si vous avez un grand écran, les petits caractères peuvent être difficiles à lire. {{Activer le mode 132 colonnes}}

13.8.3 Curseur

On peut paramétrer le curseur pour qu'il apparaisse comme un rectangle (= bloc) {Blk}. D'autres options sont le souligné {Line} ou le clignotement. Je préfère un bloc non clignotant {Steady} puisqu'il est suffisamment grand pour qu'on le trouve rapidement sans qu'il y a ait de clignotement distrayant. Si vous le rendez invisible (une option sur certains terminaux) il disparaîtra mais les nouveaux caractères apparaîtront à l'écran au fur et à mesure que vous tapez sous le curseur invisible.

13.8.4 Attributs d'affichage (cookies magiques)

Les 7.4 (attributs d'affichage) peuvent être soit des cookies magiques soit des octets d'attributs assignés à chaque caractère. Pour les cookies magiques, il y a une limite à leur étendue : leur effet s'étend-il jusqu'à la fin de la ligne ou jusqu'à la fin de la page ? Il est mieux d'utiliser des octets d'attributs (qui peuvent en réalité être des demi-octets = petits bouts).

13.8.5 Caractères de contrôle d'affichage {Monitor}

Ils peuvent avoir des noms variés tels que "contrôles d'affichage". Désactivés (par défaut), ils veulent dire "interpréter les caractères de contrôle". Quand on les active, vous voyez les séquences d'échappement de l'hôte (que vous ne voyez jamais à l'écran en temps normal). Pour que ces séquences soient visualisées d'un coup sur une ligne, le terminal ne les interprète pas, sauf la séquence CR LF, qui crée une nouvelle ligne. Voyez 7.3.1 (codes de contrôle).

13.8.6 Largeur/hauteur double

Certains terminaux peuvent afficher des caractères en largeur et/ou en hauteur double. Cette possibilité est rarement nécessaire. En changeant une ligne en largeur double (DW = Double Width) la moitié gauche (RH = Right Half) est poussée à l'extérieur de l'écran et il faut se poser la question de l'effacer ou non. "Préserver" veut dire garder la moitié gauche des lignes en largeur double. En mode hauteur double, il peut être nécessaire d'envoyer chacune de ces lignes deux fois (la deuxième fois une ligne plus bas) pour obtenir une ligne en hauteur double à l'écran.

13.8.7 Vidéo inverse {Display} (Fond clair/foncé)

La vidéo normale consiste en des lettres (premier plan) claires (blanches, vertes, ambre) sur un fond foncé (noir). La vidéo inverse {Display Light} est le contraire : du texte noir sur un fond clair. Cela est plus commode pour les yeux (sauf si la pièce est sombre).

13.8.8 Ligne d'état

Une ligne d'état est une ligne en haut ou en bas de l'écran qui affiche des informations sur l'application en cours d'utilisation. Elle est souvent mise en valeur d'une certaine manière. Avec une telle ligne d'état, une application peut envoyer au terminal une séquence d'échappement spéciale qui veut dire que le texte qui suit concerne la ligne d'état. Cependant, beaucoup d'applications n'utilisent pas cette possibilité et à la place simulent une ligne d'état réelle en positionnant directement le curseur. L'utilisateur de base qui le regarde ne voit pas la différence.

13.8.9 Pendant le changement 80/132 : effacer ou préserver ?

En basculant le nombre de colonnes de 80 à 132 (ou l'inverse), est-ce que les données affichées dans le format précédent doivent être effacées ou préservées ? {80/132 Clr} {{Screen Width Change}}. La manière dont vous positionnez cette option ne devrait pas faire de différence puisque si une application utilise 132 colonnes, elle devrait positionner cette option correctement grâce à une séquence d'échappement.

13.9 Options liées aux pages

Pour qu'un terminal Wyse puisse accéder à plusieurs pages dans la mémoire d'affichage, il faut activer {Multipage}.

13.9.1 Taille de la page

La mémoire du terminal peut être divisée en un certain nombre de pages. Voyez 8.4 (pages) et 20.6 (pages (définition)) pour une explication sur les pages. Vous pouvez partitionner la mémoire de pages en un certain nombre de pages de taille déterminée. Les applications Linux ne semblent pas utiliser les pages pour le moment, il ne devrait donc pas y avoir de différence au niveau de la configuration.

13.9.2 Couplage (du curseur et de l'affichage)

La mémoire du terminal peut être divisée en un certain nombre de pages. Voyez 8.4 (pages) et 20.6 (pages (2)) pour avoir une explication sur les pages. Quand on déplace le curseur à un endroit en mémoire vidéo qui n'est pas affichée à cet instant (comme une autre page, ou sur la même page mais à un endroit non affiché à l'écran), est-ce que l'affichage doit suivre pour que l'on voie la nouvelle position du curseur ? Si c'est le cas, c'est ce qu'on appelle le "couplage". En ce qui concerne le mouvement du curseur dans la même page, il y a le "couplage vertical" et le "couplage horizontal". En ce qui concerne le mouvement vers une autre page, il y a le "couplage de page".

13.10 Faire un rapport et répondre

Le terminal donnera son identité et son état, ou enverra un message pré-enregistré en réponse à certaines séquences d'échappement.

13.10.1 Message de réponse (chaîne)

Vous pouvez écrire un message court pendant la configuration qui peut être envoyé de manière optionnelle à l'hôte durant le démarrage ou être envoyé à l'hôte en réponse à une demande de l'hôte (peut-être le caractère de contrôle ENQ (enquire, demande)).

13.10.2 Réponse automatique

Si ce paramètre est positionné, envoie le message de réponse à l'hôte durant le démarrage sans que l'hôte le demande. Est-ce qu'un processus "getty" quelconque cherchent ce message ??

13.10.3 Réponse cachée

Si ce paramètre est positionné, il ne laissera personne voir le message de réponse (sauf bien sûr l'ordinateur hôte). S'il faut le modifier, désactivez "réponse cachée" et le message précédemment caché sera détruit pour que vous puissiez alors entrer un nouveau message (mais vous ne verrez pas le message précédent).

13.10.4 Numéro ID du terminal {ANSI ID}

Le terminal envoie cette réponse à la réception d'une demande d'identité.

13.11 Options du clavier

13.11.1 Clic de touche

Quand ce paramètre est positionné, la pression de n'importe quelle touche provoque un cliquetis (émis par un minuscule haut-parleur dans le clavier). Ces cliquetis ennuient certaines personnes et je pense qu'il est mieux de désactiver cette option.

13.11.2 Verrouillage majuscule {Keylock}

Quand la touche de verrouillage majuscule est enfoncée, les touches alphabétiques doivent-elles être les seules à générer des caractères en majuscule ? Si le paramètre est positionné à {Caps} ou majuscule uniquement (upper-case-only), alors l'appui sur une touche numérique avec le verrouillage majuscules tapera le chiffre en question. Pour obtenir le symbole au-dessus du chiffre il faudra appuyer sur la touche shift à la main. C'est le mode normal. S'il est positionné sur {Shift} toutes les touches tapent alors le caractère "décalé" quand le verrouillage majuscules est enfoncé (l'appui sur la touche 5 devrait afficher % sans devoir appuyer sur Shift, etc.) NdT : sur les claviers français, c'est l'inverse, les chiffres étant normalement obtenus en appuyant sur la touche Shift.

13.11.3 Répétition automatique {Repeat}

Si on maintient une touche enfoncée, celle-ci est "tapée" de manière répétée. Cela est pratique pour taper répétitivement le même caractère pour créer une ligne à travers une page.

13.11.4 Sonnette de marge

Quand le curseur arrive à 8 colonnes du bord droit de l'écran, une sonnette sonne (comme sur une vieille machine à écrire). Quasiment tous les éditeurs créeront automatiquement une ligne nouvelle si nécessaire (pas besoin d'appuyer sur la touche entrée), par conséquent cette possibilité est rarement nécessaire.

13.11.5 Redéfinir les touches

Le code envoyé à l'hôte quand on appuie sur une touche correspond normalement au code ASCII de cette touche (et dépend aussi des touches Shift et Control). Sur certains terminaux vous pouvez faire en sorte que n'importe quelle touche envoie n'importe quel code. C'est-à-dire que vous pouvez complètement redéfinir le clavier en configurant le terminal de cette façon. Cela peut être utile pour certaines langues étrangères et pour les claviers Dvorak etc. qui vous permettent de taper plus rapidement.

13.11.6 Touche de coin (uniquement pour les Wyse)

Les terminaux Wyse possèdent une touche près du coin inférieur gauche qu'on peut configurer pour des fonctions variées. Elle peut être appelée "Funct", "Compose Character", "Alt", "Hold" or "Scroll Lock". Les modèles récents n'ont pas toutes les options suivantes : Quand on la met sur {Hold} (arrêt du défilement, No-Scroll), elle permet d'arrêter le flux de données (en utilisant le contrôle de flux) vers le terminal. L'appui à nouveau sur cette touche redonne un flux normal. Quand on la met sur {Compose} elle permet de générer un nombre limité de caractères non-latins prédéfinis. Quand on la met sur Meta, cela en fait une touche Meta qui positionne le bit de haut rang sur chaque octet. Quand on la met sur {Funct} (et qu'on l'enfonce), chaque touche alphanumérique enfoncée envoie un octet préfixe (SOH) et un octet suffixe (CR) qui encadrent le code de l'octet ASCII. Quand on la met sur {Kpd Compose} (et qu'on l'enfonce) alors la frappe d'un nombre décimal sur le pavé numérique (suivi de "entrée") envoie le même chiffre en hexadécimal ??

13.11.7 Envois grâce au pavé numérique ou aux touches fléchées

On peut configurer le pavé numérique (le rectangle de touches presque toutes numériques à la droite de la partie principale du clavier) pour envoyer des codes spéciaux qui feront certaines choses dans certaines applications. Idem pour les touches fléchées. Il y a donc un mode "normal" où elles envoient ce qui est marqué sur la touche (ou la séquence d'échappement normale pour une touche fléchée) et un mode "application" où une séquence d'échappement spéciale est envoyée. Dans certains cas il y a un mode numérique "hexadécimal" qui ressemble au mode numérique à la différence que 6 touches non numériques envoient les lettres A à F. On peut ainsi taper par exemple "B36F" sur le pavé numérique.

13.11.8 Qu'envoient les touches Shift+Del et Shift+Backspace ?

Selon la manière dont les touches sont configurées Shift-Del envoie parfois le caractère de contrôle CAN et Shift-Backspace envoie parfois DEL.

13.11.9 Codes de balayage PC

Les terminaux récents peuvent émuler un clavier de PC en envoyant des codes de balayage PC (voyez le Keyboard-and-Console-HOWTO) au lieu de codes ASCII. Cela pourrait être utilisé si vous étiez directement relié à un PC sous DOS/Windows. Positionnez {Keycode} sur {Scan}. L'émulation du "PCTerm" DEC devrait faire la même chose et même plus. Un port série sous Linux ne peut pas s'occuper de tels codes de balayage.

13.11.10 Caractères alternés

Certaines touches peuvent contenir des lettres alternatives. Quand on positionne les touches sur "machine à écrire" (Typewriter) elles envoient ce qu'elles enverraient normalement sur une machine à écrire. Quand on les positionne sur autre chose, les caractères alternatifs sont envoyés.

13.12 Signification des codes de contrôle reçus

13.12.1 Nouvelle ligne automatique {Newline}

Dans ce cas "nouvelle ligne" veut dire une ligne nouvelle démarrant sur le bord gauche en dessous de la ligne en cours. Sous Linux et en C "nouvelle ligne" (NL) peut avoir une signification différente : le caractère de saut de ligne LF qu'on appelle aussi nouvelle ligne ou NL. C'est parce que dans les fichiers texte Linux, le caractère LF veut dire "une nouvelle ligne commence ici" et donc on l'appelle NL. Normalement, un LF (NL) envoyé à un terminal ne fait que descendre le curseur d'une ligne par rapport à l'endroit où il se trouvait et ne donne pas une nouvelle ligne.

Si le paramètre Auto New Line (nouvelle ligne automatique) est positionné, la situation "normale" ci-dessus est annulée et une nouvelle ligne physique est créée sur l'écran en recevant un LF de l'hôte. C'est exactement ce qu'on veut sous Linux. Sauf que (quand Auto New Line est positionné) la touche Retour (ou Entrée) envoie une séquence CR LF à l'hôte (pour les Wyse et les VT100, mais pour les VT420 ??). Puisque Linux utilise LF comme marqueur de "nouvelle ligne" dans les fichiers, Linux n'aimerait recevoir qu'un LF (et pas un CR LF). L'option "New Line" est par conséquent rarement utilisée. À la place, les traductions nécessaires sont effectuées par le pilote du port série par défaut. C'est comme si on envoyait la commande "stty onlcr icrnl" mais vous n'avez pas besoin de le faire puisque c'est le comportement par défaut.

13.12.2 Saut de ligne automatique {Rcv CR}

Cela est simplement un autre type de "nouvelle ligne automatique". Quand un caractère CR (retour chariot) est reçu, une action LF (saut de ligne) est ajoutée ce qui résulte en l'affichage d'une nouvelle ligne. Puisque Linux marque les fins de lignes par LF, cette option n'est pas utilisée.

13.12.3 Reconnaître DEL (seulement pour Wyse ??) ou NULL

Si ce paramètre est désactivé, tout caractère DEL reçu par le terminal est ignoré. S'il est activé, DEL effectue un retour en arrière destructif. Les caractères NULL sont en général ignorés dans n'importe quel cas. Les caractères DEL et NULL sont parfois utilisés ensemble pour faire du remplissage. Voyez 10.2 (Remplissage).

13.13 Où va le nouveau texte

13.13.1 Passage à la ligne

On l'appelle aussi passage automatique (Auto Wrap, Auto Wraparound). Qu'arrive-t-il quand on atteint le bord droit de l'écran (colonne 80, etc.) et que l'hôte n'a pas envoyé de caractère retour (ou autre) ? Si le passage à la ligne est activé, le reste de la ligne s'affiche alors sur la ligne suivante, etc. Sinon, le reste de la ligne est perdu et n'est pas affiché à l'écran. Toute application qui se respecte devrait envoyer des caractères "retour" pour que le passage s'effectue avant d'atteindre le bord de l'écran. Cependant une commande de copie brute (et d'autres situations) peut ne pas le faire, il est donc en général mieux de positionner le passage à la ligne.

Pour un écran de 80 colonnes, la plupart des terminaux ne passent à la ligne que si le 81ème caractère venant de l'hôte est un caractère graphique (imprimable). Cela autorise les cas où le 81ème caractère venant de l'hôte peut être un "retour" ou "nouvelle ligne" (caractères non graphiques) qui veut dire que l'application gère bien le passage à la ligne et le terminal n'a pas besoin d'intervenir.

13.13.2 Défilement

Le défilement {Scrl} est quand toutes les lignes de l'écran se déplacent vers le haut ou vers le bas. On parle aussi de "panoramique" (pan), ce qui comprend un mouvement sur les côtés. Dans le défilement ordinaire, les lignes disparaissent vers le bas ou vers le haut de l'écran et de nouvelles lignes envoyées par l'hôte apparaissent sur le bord opposé (haut ou bas). Il y en a trois sortes : doux, par sauts ou instantané. Le défilement instantané n'est pas vraiment un défilement puisqu'il y a remplacement instantané d'une page écran par une nouvelle (bien que certaines lignes du nouvel écran puissent provenir de l'écran précédent). Le défilement par sauts est quand les nouvelles lignes arrivent en sautillant les unes après les autres [NdT

: en fait, elles arrivent une à la fois, mais la tentation était trop forte ;-)]. Le défilement doux {Smth} est quand le texte se déplace à une vitesse soutenue vers le haut ou vers le bas. Si la vitesse de défilement douce est suffisamment lente, on peut lire les lignes qui apparaissent alors qu'elles sont encore en train de bouger (en mouvement).

Le défilement doux sur les terminaux lents était utile autrefois puisqu'on pouvait continuer à lire alors que l'affichage défilait. Mais avec les vitesses de transfert plus rapides, le défilement par sauts est si rapide qu'on ne perd que peu de temps en attendant l'apparition du nouvel affichage. Puisque la lecture d'un texte qui défile prend légèrement plus de temps que la lecture d'un texte fixe, on peut en fait perdre plus de temps si on sélectionne un défilement doux.

Si on désactive le défilement automatique {Autoscrl}, le nouveau texte venant de l'hôte doit aller quelque part, alors il est mis en haut de l'affichage. Si l'ancien texte n'est pas effacé, le nouveau texte se mélange (de façon insensée) à l'ancien. Si le vieux texte est effacé, le nouveau texte sort alors de son contexte. Conservez donc le défilement automatique.

13.13.3 Nouvelle page ?

Voyez 8.4 (pages) et 20.6 (pages (définition)) pour avoir une explication sur les pages. Quand la page courante est pleine (la dernière ligne est terminée), est-ce que la page doit défiler, ou est-ce qu'une nouvelle page doit être créée (laissant la page précédente stockée dans la mémoire d'affichage du terminal) ? Si le paramètre {Autopage} est positionné, une nouvelle page est créée. Puisque vous n'utilisez probablement pas les pages, vous devriez sûrement désactiver ce paramètre.

13.14 Touches de fonction

Ce sont les touches nommées F1, F2, etc. Sur les vieux terminaux elles s'appellent peut-être PF1, PF2, etc. où le P veut dire (P)rogrammable. Certains claviers ont les deux. On peut programmer (redéfinir) ces touches pour envoyer une chaîne d'octets définie par l'utilisateur. On peut souvent les "programmer" facilement en utilisant un certain menu de configuration {FKey}. Sur certains terminaux, on peut aussi spécifier où la chaîne est envoyée quand on appuie sur la touche. En mode "normal" l'appui sur la touche ressemble à la frappe d'une chaîne au clavier. En mode "local" l'appui sur la touche envoie la chaîne au terminal (comme si le terminal était en mode local). On peut utiliser cela pour envoyer des séquences d'échappement au terminal pour le configurer d'une manière spéciale. En mode "lointain" la chaîne est toujours envoyée sur le port série vers l'ordinateur hôte (même si le terminal est en mode local).

13.15 Options en mode par blocs

Certaines options ne concernent que le cas du 21.7 (mode par blocs). Cette option est utile puisqu'elle fournit des formulaires et retire une charge à l'hôte en transmettant par rafales. Mais c'est plus compliqué à configurer et ce n'est donc pas beaucoup utilisé.

13.15.1 Affichage de formulaires

En mode par blocs, certaines parties de l'écran concernent le texte des formulaires et sont donc protégées en écriture "Prot" {WPRT}. Des options peuvent faire que les caractères dans ces parties apparaissent assombris, en vidéo inverse {WPRT Rev} et/ou soulignés {WPRT Undrln}. {WPRT Intensity} (l'intensité) peut être sombre, normale ou même vierge (invisible).

13.15.2 Envoi par blocs

Est-ce que le texte protégé en écriture (le texte d'origine du formulaire) doit être envoyé à l'hôte lors de la transmission d'un bloc : {Send All} ou est-ce que le texte protégé en écriture est aussi protégé en lecture : {Send Erasable} ?

13.15.3 Partie à envoyer

Doit-on envoyer l'écran entier ou simplement la partie qui défile ? {Send Area}. L'envoi doit-il s'arrêter quand la position courante du curseur est atteinte ? Si {Xfer Term} est mis sur Cursor, seules les données sur l'écran jusqu'au curseur sont envoyées.

13.15.4 Délimiteur de bloc / de page

Quel est le symbole de délimitation à ajouter à un bloc de données ? {Blk End} ou à la fin d'une page {Send Term}inaison.

13.16 Blocages

Il y a divers types de blocages. L'un est le clavier bloqué à cause du contrôle de flux. Voyez 10.5 (blocage du clavier). Un autre blocage {Feature Lock} est celui qui empêche l'ordinateur hôte de modifier la configuration du terminal en envoyant certaines séquences d'échappement au terminal. Placer un tel verrou peut provoquer des comportements inattendus quand les applications enverront des séquences d'échappement qui seront ignorées par le terminal. Tous les paramètres de configuration ne sont pas verrouillés. Sauf si vous avez une bonne raison de le faire, vous ne devriez pas activer de tels verrous.

Un blocage de touche de fonction empêchera l'ordinateur de redéfinir ce qu'envoie une touche de fonction programmable. Vous voudrez utiliser cela si vous avez programmé quelque chose d'important dans les touches de fonction.

13.17 Économiseur d'écran {Scrn Saver}

On l'appelle aussi "CRT Saver" (NdT : ce qui revient au même). Ceci éteint (ou diminue la luminosité) de l'écran quand le terminal n'est pas utilisé pendant un certain temps. La durée de vie de l'écran est prolongée et cela peut économiser de l'énergie. L'appui sur une touche permettra en général de retrouver l'écran et peut "exécuter" cette touche donc il est préférable d'appuyer sur la touche Shift etc.

13.18 Imprimante

Pour les Wyse, s'il n'y a pas d'imprimante reliée, désactivez {Printer Attached}. Ce n'est pas essentiel de faire ceci, mais si vous le faites toute séquence d'échappement envoyant du texte à l'imprimante (au lieu du terminal) sera ignorée.

Paramétrer le port de l'imprimante est à peu près la même chose (en général plus simple) que de paramétrer les communications sur le port principal. Il y a quelques options spécifiques à l'imprimante. Est-ce une imprimante série ou parallèle ? Si c'est parallèle, on doit la désigner comme telle dans la configuration et la connecter sur le port parallèle du terminal (s'il y en a un). Doit-on envoyer un FF (saut de page) à l'imprimante à la fin d'un travail d'impression ? Si {Print Term} est mis sur FF, c'est ce qui se passera.

14 Détails de la configuration de l'ordinateur

Il faut éditer plusieurs fichiers pour configurer l'ordinateur pour gérer le terminal. Si vous avez de la chance, vous ne devrez éditer que /etc/inittab. On fait ce travail d'édition à partir de la console (ou de n'importe quel terminal qui fonctionne).

14.1 Getty (dans /etc/inittab)

Afin de lancer un processus de login sur un port série quand l'ordinateur démarre (ou change de niveau d'exécution) une commande getty doit être placée dans le fichier /etc/inittab. Getty permet de faire fonctionner (GET) un terminal (TTY). Chaque terminal a besoin de sa commande getty. Il y a aussi au moins une commande getty pour la console dans chaque fichier /etc/inittab. Trouvez-la et ajoutez-y les commandes getty pour les vrais terminaux. Ce fichier peut contenir des lignes d'exemples de commandes getty pour les terminaux texte mises en commentaire, et donc tout ce qu'il vous reste à faire est d'enlever les commentaires (enlevez le # au début de la ligne) et de modifier quelques arguments.

Les arguments autorisés dépendent du getty que vous utilisez :

Les deux meilleurs getty pour les terminaux reliés de manière directe sont :

- agetty (qu'on appelle parfois simplement getty) : très facile à configurer. Pas de fichiers de configuration. Voyez 14.1.1 (Agetty) ;
- Voyez 14.1.2 (getty (fait partie du paquet getty_ps)) ;

Les deux gettys plus appropriés pour les modems (évitez-les pour les terminaux) sont :

- mgetty : le meilleur pour les modems ; fonctionne avec les terminaux mais inférieur ;
- uugetty : uniquement pour les modems, fait partie du paquet getty_ps ;

Un getty simple à utiliser uniquement pour les logins sur la console :

- mingetty : uniquement pour les consoles.

Si vous n'avez pas le getty que vous désirez, cherchez-le dans d'autres distributions et utilisez le programme `alien` pour le convertir entre paquets RPM et Debian. Le code source sur *Metalab (logiciels série)* `http://metalab.unc.edu/pub/Linux/system/serial/` .

Si vous n'utilisez pas les lignes de contrôle du modem (par exemple si vous n'utilisez que les 3 conducteurs minimums : transmission, réception et masse commune) vous devriez le faire savoir à getty en utilisant un drapeau "local". Le format de celui-ci dépend du getty que vous utilisez.

14.1.1 Agetty (peut s'appeler getty)

Un exemple de ligne dans /etc/inittab :

```
S1:23:respawn:/sbin/getty -L 19200 ttyS1 vt102
```

S1 vient de ttyS1. 23 veut dire que getty est lancé en entrant dans les niveaux d'exécution 2 ou 3. respawn veut dire que si getty est tué, il se relancera automatiquement. /sbin/getty est la commande getty. Le -L veut dire Local (ignorer les signaux de contrôle du modem). -h (non montré dans l'exemple) permet le contrôle de flux matériel (même chose que stty crtscts). 19200 est la vitesse de transmission. ttyS1 veut dire /dev/ttyS1 (COM2 sous MS-DOS). vt102 est le type de terminal et ce getty donnera cette valeur à la variable d'environnement TERM. Il n'y a pas de fichiers de configuration. Tapez "init q" sur la ligne de commande après avoir édité la ligne de getty et vous devriez apercevoir une invite de login.

La détection de parité de Agetty Le programme `agetty` détectera automatiquement la parité configurée dans le terminal. Excpté si vous utilisez 8bit d'octet de données avec 1-bit de parité. Si vous utilisez `stty` pour fixer la parité, `agetty` la désactivera automatiquement puisqu'il veut que le bit de parité passe comme si c'était un bit de donnée. C'est parce qu'il a besoin d'obtenir le dernier bit (qui peut être un bit de parité) pendant que vous tapez votre nom de login afin d'auto-détecter la parité. Donc, si vous utilisez la parité, ne l'activez que du côté du terminal et laissez `agetty` la détecter automatiquement et la positionner sur l'ordinateur. Si votre terminal supporte la parité en réception, l'invite de login sera brouillée jusqu'à ce que vous tapiez quelque chose et que getty positionne la parité. L'invite brouillée repoussera les visiteurs etc. qui essaient de se logger. Cela peut être exactement ce que vous voulez.

Il y a parfois des problèmes avec l'autodetection de parité. Cela arrive car après la première frappe de votre login, `agetty` utilise le programme `login` pour finir de vous loguer. Si le premier essai de login échoue, `login` se relance pour s'occuper des éssais futurs de login (incluant l'écriture de votre login). Le problème est que seulement agetty peut detecter la parité tandis que le programme `login` ne le fait pas. Donc, si vous remontez dans le programme `login` pour quelque raison que ce soit et que la parité n'a pas encore été détectée, vous etes en difficulté tant que le programme `login` ne peut pas detecter la parité. Avec une mauvaise parité, `login` ne peut pas lire correctement ce que vous écrivez et vous ne pouvez pas vous loguez. Si votre terminal supporte la réception de parité, vous continuerez à voir un écran brouillé.

On peut arriver dans cette "boucle de login" de plusieurs façons. Supposez que vous tapez une ou deux lettres seulement pour votre login et que vous tapez Entrée. Si ces lettres ne sont pas suffisantes pour la detection de parité, alors `login` se lancera avant que la parité soit detectée. Quelque fois ce problème arrive si vous n'avez pas le terminal allumé et connecté quand `agetty` démarre pour la première fois. Si vous restez bloqué dans cette "boucle de login", une solution est d'attendre à peu près une minute, le temps qu'agetty se relance dû au "timeout".

La parité d'Agetty avec des octets de données de 8-bit Malheureusement, `agetty` ne peut pas detecter cette parité. Il (fin 1999) n'y a pas d'options pour desactiver l'auto-detection de parité et cela detectera les parités incorrects. Le résultat est que le processus de login sera brouillé et la parité sera mal reglé. Ainsi il ne parait pas faisable d'essayer d'utiliser des octets de données de 8-bit avec parité.

14.1.2 getty (fait partie de getty_ps)

(Ceci est tiré du vieux Serial-HOWTO de Greg Hankins).
Ajoutez des entrées pour `getty` pour utiliser votre terminal dans le fichier de configuration `/etc/gettydefs` si elles ne sont pas déjà présentes :

```
# 38400 bps Dumb Terminal entry
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

# 19200 bps Dumb Terminal entry
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# 9600 bps Dumb Terminal entry
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

Si vous voulez, vous pouvez faire en sorte que `getty` affiche des choses intéressantes dans la bannière de login. Dans mes exemples, je fais afficher le nom du système et la ligne série. Vous pouvez ajouter d'autres choses :

```
@B    la vitesse courante (évaluée au moment où @B est rencontré).
@D    la date courante, au format MM/JJ/AA.
@L    la ligne série à laquelle est attaché getty.
@S    le nom du système.
@T    l'heure courante, au format HH:MM:SS (24 heures).
@U    le nombre d'utilisateurs actuellement loggés. C'est le compte
      du nombre d'entrées dans le fichier /etc/utmp qui possèdent
      un champ ut_name non nul.
@V    la valeur de VERSION, donnée dans les fichiers de valeurs par
      défaut.
Pour afficher un caractère '@', utilisez soit '\@', soit '@@'.
```

Quand vous avez fini d'éditer `/etc/gettydefs`, vous pouvez vérifier que la syntaxe est correcte en faisant :

```
linux# getty -c /etc/gettydefs
```

Assurez-vous qu'il n'y a pas de fichier de configuration `getty` ou `uugetty` pour le port série auquel est attaché votre terminal (`/etc/default/`{uu}`getty.ttyS`*N* ou `/etc/conf.`{uu}`getty.ttyS`*N*), car cela entrera sûrement en conflit avec le lancement de `getty` sur un terminal. Enlevez le fichier s'il existe.

Éditez le fichier `/etc/inittab` pour lancer `getty` sur le port série (en mettant les informations correctes pour votre environnement – port, vitesse et type de terminal par défaut) :

```
S1:23:respawn:/sbin/getty ttyS1 DT9600 vt100
```

Relancez `init` :

```
linux# init q
```

À ce point, vous devriez voir une invite de login sur votre terminal. Vous devrez peut-être appuyer sur Retour pour que le terminal soit attentif.

14.1.3 mgetty

Le "m" veut dire modem. Ce programme est d'abord destiné aux modems et en mi-1999 ne fonctionnait pas toujours très bien pour les terminaux texte. Il est très mal documenté pour les terminaux et vous devrez parcourir beaucoup de documentation sur les modems pour déterminer comment l'utiliser pour un terminal. Regardez les dernières lignes de /etc/mgetty/mgetty.config pour avoir un exemple de la configuration d'un terminal. [Note du relecteur : je le trouve au contraire bien documenté (janvier 1999) dans man mgetty : un mgetty -r -s 9600 /dev/ttyS0 (par exemple) est suffisant. Le -r indique que la connexion est directe (sans modem).] Ceci sera, espérons-le, réparé dans le futur. Il serait bien d'avoir le même getty à la fois pour les terminaux et les modems mais mgetty nécessite quelques améliorations avant de convenir pour les deux utilisations.

14.2 Stty et Setserial

Il y a à la fois une commande "stty" et une commande "setserial" pour configurer les ports série. Certains (ou tous les) paramètres stty nécessaires peuvent être positionnés grâce à getty et il peut ne pas être nécessaire d'utiliser setserial ; l'utilisation de ces deux commandes peut donc ne pas être nécessaire. Celles-ci (stty et setserial) paramètrent différents aspects du port série. Stty en fait la plupart tandis que setserial configure la partie bas niveau comme les interruptions et les adresses de ports. Pour "sauvegarder" les paramètres, ces commandes doivent être écrites dans certains fichiers (scripts shell) qui sont lancés à chaque démarrage de l'ordinateur. Les distributions de Linux fournissent souvent un script shell qui lance `setserial` mais en fournissent rarement un qui lance `stty` tant qu'on en aura rarement besoin..

14.3 Setserial

14.3.1 Introduction

N'utilisez jamais `setserial` avec des portables (PCMCIA). `setserial` est un programme vous permettant d'indiquer au logiciel pilote l'adresse d'entrée/sortie du port série, quelle interruption (IRQ) est positionnée dans le matériel du port, le type d'UART que vous possédez, etc. Il peut aussi vous montrer comment le pilote est configuré à ce moment. En plus, il peut faire des requêtes au matériel (si certaines options sont données).

Si vous avez seulement un ou deux ports séries, ils seront bien configuré sans utiliser setserial. Autrement (ou si il y'a des problèmes avec ce port série) vous devrez utiliser setserial. En plus du manuel de `setserial`, regardez les informations dans `/usr/doc/setserial.../` ou autre. Cela devrait vous indiquer comment setserial se comporte dans votre distribution de Linux.

`Setserial` est souvent lancé automatiquement au démarrage par un script shell. Il ne fonctionnera que si le module série est chargé. Si vous devez pour une raison ou pour une autre décharger le module série plus tard, les modifications faites précédemment par `setserial` seront oubliées. `Setserial` doit donc être relancé pour les prendre en compte à nouveau. En plus de le lancer avec un script de démarrage, quelque chose semblable à `setserial` se lance quand le module série est chargé. Ainsi quand vous regardez les messages de démarrage sur l'écran il pourra vous sembler être lancé deux fois, et en fait c'est ce qui s'est passé.

Setserial peut régler le temps que le port restera actif après qu'il soit fermé (pour sortir les caractères qui sont encore dans leurs buffers dans la RAM principale). C'est necéssaire pour un taux de transferts de 1200 baud ou plus bas. C'est aussi necéssaire pour des vitesses plus rapides si il y'a beaucoup de "contrôle de flux" en attente.

Si votre port série est Plug-and-Play, vous devrez peut-être consulter d'autres HOWTOs, comme Plug-and-Play et Serial.

Avec les bonnes options, `setserial` peut chercher (à une adresse d'entrée/sortie donnée) un port série mais vous devez deviner l'adresse d'entrée/sortie. Si vous lui demandez de chercher /dev/ttyS2 par exemple, il ne cherchera qu'à l'adresse où il pense trouver ttyS2. Si vous dites à setserial que ttyS2 est à une adresse différente, alors il cherchera à cette adresse, etc. Voyez 14.3.2 (Recherche).

Setserial ne positionne pas lui-même les IRQ ou les adresses d'entrée/sortie dans le matériel du port série. Ceci est fait soit avec des cavaliers, soit par plug-n-play. Vous devez dire à setserial les valeurs mêmes qui ont été configurées dans le matériel. N'inventez pas simplement des valeurs dont vous pensez qu'elles font joli en les soumettant à setserial. Cependant, si vous connaissez les adresses d'entrée/sortie mais pas l'IRQ, vous pouvez demander à setserial de tenter de déterminer l'IRQ.

Vous pouvez voir une liste des commandes possibles et utilisables (mais pas les options à une lettre telles que -v pour verbeux – que vous devriez normalement utiliser pour déboguer) en tapant simplement `setserial` sans argument. Notez que setserial nomme une adresse d'entrée/sortie un "port". Si vous tapez :

```
setserial -g /dev/ttyS*
```

vous verrez quelques informations sur la manière dont ce pilote de périphériques est configuré pour vos ports. Ajoutez un "v" à l'option "-g" pour en voir plus. Mais ceci ne vous dira pas si le matériel dispose vraiment de ces valeurs. En fait, vous pouvez lancer setserial et assigner une adresse d'entrée/sortie purement fictive, n'importe quelle IRQ, et tout type d'UART que vous aimeriez avoir. Alors, la prochaine fois que vous lancerez "setserial ...", il affichera ces valeurs fausses sans se plaindre. Notez que les assignations faites par setserial sont perdues quand le PC est éteint donc il est en général lancé automatiquement quelque part à chaque fois que Linux est démarré.

14.3.2 Recherche

Afin de tenter de trouver si vous avez un certain type de matériel série, vous devez d'abord savoir (ou deviner) son adresse d'entrée/sortie (ou le pilote de périphérique doit en avoir une adresse d'entrée/sortie, sûrement positionnée précédemment par setserial). Pour tenter de détecter le matériel physique, utilisez l'option -v (verbeux) et la commande `autoconfig` de `setserial`. Si le message résultant montre un type d'UART tel que 16550A, alors tout est bon pour vous. Si par contre il affiche un type d'UART "`unknown`" (inconnu), alors il n'y a sûrement pas de port série du tout à cette adresse d'entrée/sortie. Certains ports série bon marché ne s'identifient pas correctement donc si vous voyez "`unknown`" vous avez peut-être quand même un port série à cet endroit.

En plus de faire une auto-détection sur le type d'UART, setserial peut aussi déterminer automatiquement les IRQs, mais ceci ne fonctionne pas toujours bien non plus. Dans les versions de setserial >= 2.15, votre dernier test de recherche peut être sauvé et placé dans le fichier de configuration `/etc/serial.conf` qui sera utilisé au prochain démarrage de Linux. Le script qui lance `setserial` au démarrage ne fait généralement pas de recherche, mais vous pouvez le modifier pour qu'il le fasse. Voyez la section suivante.

14.3.3 Linux peut-il configurer les périphériques série automagiquement ?

Oui, mais... Votre distribution doit deja le faire au démarrage. Mais vous pouvez le customiser. C'est facile à faire avec setserial < 2.15. Ajoutez simplement quelques lignes au fichier qui lance setserial au démarrage. Voyez 14.3.5 (vieille méthode de configuration : édition d'un script). Par exemple, pour ttyS3 vous ajouteriez :

```
/sbin/setserial /dev/ttyS3 auto_irq skip_test autoconfig
```

au fichier qui lance setserial au démarrage. Faites ceci pour chaque port série que vous voulez auto-configurer. Assurez-vous de donner un nom de périphérique qui existe vraiment sur votre machine. Dans certains cas ca ne marchera pas bien à cause du matériel, donc vous pouvez lui assigner un irq et/ou un type d'uart. Par exemple:

```
/sbin/setserial /dev/ttyS3 irq 5 uart 16550A skip_test
```

Pour les versions >= 2.15 (à condition que votre distribution ait inclus la modification, Redhat ne l'a pas fait), il est plus difficile de le faire puisque le fichier qui lance setserial au démarrage, /etc/init.d/setserial ou autre n'a pas été prévu pour être édité par l'utilisateur. Il peut ne pas y avoir de commentaires utiles comme il y en avait dans les versions précédentes.

14.3.4 Configuration au démarrage

Quand le noyau charge le module série (ou si le "module" est intégré au noyau) alors seuls `ttyS{0-3}` sont auto-détectés et le pilote est configuré avec les IRQs 4 et 3 (peu importe la configuration réelle du matériel).

Vous le voyez sur un message au démarrage comme si `setserial` avait été lancé. Si vous utilisez 3 ports ou plus, ceci peut engendrer des conflits d'IRQ.

Pour régler de tels conflits en donnant à setserial les vraies IRQs (ou pour d'autres raisons) il peut y avoir un fichier quelque part qui lance `setserial` à nouveau. Ceci se passe tôt pendant le démarrage avant que n'importe quel processus utilise le port série. En fait, votre distribution peut avoir configuré les choses pour que le programme setserial se lance automatiquement à partir d'un script au démarrage. On peut trouver plus d'informations pour gérer cette situation dans /usr/doc/setserial.../ ou autre.

Nouvelle méthode de configuration en utilisant /etc/serial.conf les versions inferieurs à la 2.15 de `setserial`, la façon pour le configurer était d'édité le script shell qui lancait setserial au démarrage. Avec la version 2.15 (1999) de `setserial` le script shell n'est pas édité mais est lancé au démarrage et obtient ses données à partir d'un fichier de configuration: /etc/serial.conf. Mais on n'édite normalement jamais /etc/serial.conf. À la place, utilisez simplement `setserial` sur la ligne de commande.

Normalement, ce que vous avez modifié avec la commande setserial est sauvé dans le fichier de configuration (serial.conf) quand vous éteignez (normalement) ou que vous redémarrez. Ceci ne fonctionne que si "###AUTOSAVE###" ou similaire se trouve sur la première ligne de serial.conf. Si vous devez utiliser `setserial` de manière expérimentale et qu'il ne fonctionne pas correctement, alors n'oubliez pas de le relancer pour que les paramètres expérimentaux ne soient pas sauvés par erreur. Le fichier le plus couramment utilisé pour lancer setserial au démarrage (en restant conforme avec le fichier de configuration) est maintenant /etc/init.d/setserial (Debian) ou /etc/init.d/serial (Redhat), ou etc., mais ne devrait normalement pas être édité non plus.

Pour désactiver un port, utilisez `setserial` pour le positionner à "uart none". Le format de /etc/serial.conf apparaît être comme celui des paramètres placés après "setserial" sur la ligne de commande avec une ligne pour chaque port. Si vous n'utilisez pas autosave, vous pouvez éditer /etc/serial.conf à la main. Pour la version 2.15, la distribution Debian installe le système avec la sauvegarde automatique activée, mais Redhat 6.0 avait simplement un fichier /usr/doc/setserial-2.15/rc.serial que vous deviez déplacer dans /etc/init.d/.

BOGUE : en juillet 1999 il un bogue/problème puisqu'avec ###AUTOSAVE### seuls les paramètres de setserial affichés par "setserial -G /dev/ttyS?" (où ? vaut 0, 1, 2, ...) sont sauvés mais pas les autres paramètres. Ceci n'affecte qu'une minorité d'utilisateurs puisque les paramètres non sauvés sont de toute façon rarement utilisés. Cela a été rapporté comme un bogue et peut être réparé maintenant.

Afin de forcer les paramètres courants positionnés par setserial à être sauvés dans le fichier de configuration (serial.conf) sans éteindre la machine, faites ce qui se passe normalement quand vous éteignez : lancez le script shell `/etc/init.d/{set}serial stop`. La commande "stop" sauvera la configuration courante mais les ports série continueront de fonctionner correctement.

Dans certains cas vous pouvez avoir l'ancienne et nouvelle méthode d'installé mais heureusement juste une d'elles se lance au démarrage. Debian étiquette les fichiers obsolètes avec "...pre-2.15".

14.3.5 Ancienne méthode de configuration : édition d'un script

Avant la version 2.15 (1999) il n'y avait pas /etc/setserial.conf pour configurer setserial. Ainsi vous devez chercher un fichier qui lance setserial au démarrage et l'édité. S'il n'existe pas, vous devez en créer un (ou placer les commandes dans un fichier qui se lance tôt au démarrage). Si un tel fichier est utilisé en ce moment, il se trouve sûrement dans l'arborescence /etc. Mais Redhat <6.0 l'a mis dans /usr/doc/setserial/ bien que vous deviez le déplacer dans l'arborescence /etc avant de l'utiliser. Vous pouvez utiliser "locate" pour essayer de trouver un tel fichier. Par exemple, vous pouvez taper : locate "*serial*".

Ce que pouvez chercher peut s'appeler rc.serial ou 0setserial (Debian). Si un tel fichier est fourni, il devrait contenir un certain nombre d'exemples commentés. En décommentant certains d'entre eux et/ou en les modifiant, vous devriez pouvoir configurer les choses correctement. Assurez-vous que vous utilisez un chemin valide pour `setserial`, et un nom de périphérique valide. Vous pouvez faire un test en exécutant ce fichier à la main (tapez simplement son nom en tant que super-utilisateur) pour voir si ça fonctionne bien. Un test comme celui-ci est bien plus rapide que de faire des redémarrages à répétition pour avoir le bon résultat. Bien sûr, vous pouvez aussi tester une commande `setserial` unique en la tapant simplement sur la ligne de commande.

Le script `/etc/rc.d/rc.serial` était couramment utilisé dans le passé. La distribution Debian a utilisé `/etc/rc.boot/0setserial`. Un autre fichier qui a été utilisé est `/etc/rc.d/rc.local` mais ce n'est pas une bonne idée car il peut ne pas être lancé assez tôt. On a indiqué que d'autres processus peuvent essayer d'ouvrir le port série avant l'exécution de rc.local, ce qui entraîne des échecs de communication série.

14.3.6 IRQs

Par défaut, ttyS0 et ttyS2 partagent l'IRQ 4, tandis que ttyS1 et ttyS3 partagent l'IRQ 3. Le partage des interruptions série n'est permis que si : 1. vous avez un noyau 2.2 ou supérieur, 2. vous avez compilé le support pour le faire et 3. votre matériel série le supporte. Voyez le HOWTO Série, sur le partage des interruptions et les noyaux 2.2 et plus.

Si vous n'avez que deux ports série, ttyS0 et ttyS1, cela fonctionne encore puisque les conflits de partage d'IRQ n'existent pas pour des périphériques non existants.

Si vous ajoutez un modem interne et gardez ttyS0 et ttyS1, vous devriez alors tenter de trouver une IRQ non utilisée et la positionner à la fois sur votre port série (ou carte modem) et ensuite utiliser setserial pour l'assigner à votre pilote de périphérique. Si l'IRQ 5 n'est pas utilisée par une carte son, ce peut être une IRQ utilisable pour un modem. Pour positionner l'IRQ de manière matérielle vous devrez peut-être utiliser isapnp, un BIOS PnP ou modifier Linux pour le rendre PnP. Pour vous aider à déterminer quelles IRQs sont disponibles, tapez "man setserial" et cherchez, disons "IRQ 11".

14.4 Stty

14.4.1 Introduction

`stty` effectue la plupart de la configuration du port série mais puisque les applications (et le programme getty) la gèrent souvent, vous n'aurez peut-être pas besoin de l'utiliser souvent. C'est pratique si vous avez des problèmes ou voulez voir comment le port est paramétré. Essayez de taper “stty -a” sur votre terminal/console pour voir les paramètres actuels. Essayez aussi de taper la commande sans le -a (all = tout) pour obtenir une liste courte qui montre les paramètres différents de la normale. N'essayez pas d'apprendre tous les réglages à moins de vouloir devenir un gourou du port série. La plupart des valeurs par défaut conviennent et certains réglages ne sont nécessaires que pour certains terminaux non-intelligents et obsolètes fabriqués dans les années 1970 (Mais pas après)

Alors que `setserial` ne travaille qu'avec les ports série réels, stty s'utilise à la fois pour les ports série et pour les terminaux virtuels comme l'interface texte standard de Linux sur un moniteur de PC. Pour le moniteur de PC, la plupart des paramètres de stty n'ont pas de signification. Le changement de la vitesse de transmission, etc. ne semble pas faire grand chose.

Voici quelques uns des items que stty peut configurer : vitesse (bits/seconde), parité, bits par octet, nombre de bits de stop, enlever le 8ème bit ?, signaux de contrôle du modem, contrôle de flux, signal d'arrêt, délimiteurs de fin de ligne, changer la casse, remplissage, sonner si le tampon déborde ?, écho, permettre à des tâches de fond d'écrire sur le terminal ?, définir des caractères spéciaux (de contrôle, comme quelle touche presser pour faire une interruption). Voyez la page de manuel de `stty` ou la page info pour plus de détails. Voyez aussi la page de manuel : `termios` qui couvre les mêmes options que stty mais (en mi-1999) couvre des possibilités que la page de manuel de stty ne mentionne pas. Pour l'utilisation de certains caractères spéciaux, voyez 16.8 (caractères spéciaux (de contrôle)).

Avec certaines implémentations de getty (paquet getty_ps), les commandes qu'on enverrait normalement à stty sont entrées dans un fichier de configuration getty : /etc/gettydefs. Même sans ce fichier de configuration, la ligne de commande de getty devrait suffire pour paramètrer les choses de sorte que vous n'ayez pas besoin de stty.

On peut écrire des programmes en C qui modifient la configuration de stty etc. Regarder la documentation pour ce faire peut aider quelqu'un à mieux comprendre l'utilisation des commandes stty (et ses nombreux arguments possibles). Le Serial-Programming-HOWTO est utile. La page de manuel de termios contient la description de la structure au sens langage C (de type termios) qui stocke la configuration de stty dans la mémoire de l'ordinateur. Bien des noms de drapeaux dans cette structure C sont quasiment les mêmes (et font la même chose) que les arguments de la commande stty.

14.4.2 Utilisation de stty pour un terminal "étranger"

L'utilisation de `stty` pour inspecter ou configurer le terminal que vous utilisez est facile. Faire ca pour terminal (étranger) different ou un port série est délicat. Par exemple, supposons que êtes sur le moniteur du PC (tty1) et vouliez utiliser `stty` pour le port série ttyS2. Vous devez utiliser l'opérateur de redirection <. D'abord, soyez prévenu que s'il y a un terminal sur ttyS2 et un shell tourne sur ce terminal, ce que vous verrez alors sera décevant et une tentative de le paramétrer sera infructueuse. Voyez 14.4.3 (deux interfaces sur un terminal) pour comprendre ceci.

Tapez “stty -a < /dev/ttyS2” pour regarder les paramètres de ttyS2. Utilisez le même opérateur de redirection < pour paramétrer ttyS2. Cela fait de ttyS2 l'entrée standard de stty. Ca donne au programme stty un lien vers le "fichier" ttyS2 donc il doit le "lire". Mais au lieu de lire les octets envoyés vers ttyS2 comme on pourrait le prévoir, il utilise le lien pour trouver les paramètres de configuration du port donc il devrait les lire ou les changer. Certaines personnes tentent d'utiliser “stty ... > /dev/ttyS2” pour paramétrer le terminal. Ceci ne le fera pas. À la place, il prendra le message normal affiché par la commande stty pour le terminal sur lequel vous êtes (tty1) et envoie ce message à ttyS2 mais ne change aucun paramètre pour ttyS2.

Arrive un autre problème avec l'opérateur de redirection. Quelques fois quand on essaye d'utiliser stty, la commande s'arrette et rien ne se passe (vous n'avez pas de prompt pour une autre commande, même après avoir frapper <entrée>). C'est probablement due au port étant bloqué car il attend une ligne de controle modem pour être déclaré. Par exemple, tant que vous n'avez pas parametrer "clocal" pour ignorer les lignes de controles modem, Alors si il n'y a pas de signal CD de déclaré, le port ne s'ouvrira pas et stty ne fonctionnera pas. Une situation similaire doit exister pour le control de flux materiel. Si le cable du port n'a pas de fils pour la broche qui doit être déclaré donc il n'y a pas besoin d'arreter l'attente.

Une façon à essayer pour se passer de cette attente, est d'utiliser un programme sur le port qui le forcera à être opérationnel même si les lignes de controles disent le contraire. Ainsi heureusement, ce programme doit parametrer le port donc il n'a plus besoin du signal de controle pour ouvrir: clocal ou -crtscts. Pour se servir de "minicom" pour faire ca, il faut le reconfigurer pour un autre ttyS, etc, et le redémarrer. Puisque vous avez à reconfigurer minicom, c'est plus simple de redémarrer le PC.

Les versions à partir de 1.17 (pas encore sortie en mi-1999) n'auront plus besoin de la redirection (<) mais à la place utiliseront “stty ... -F /dev/ttyS2” (ou –file au lieu de -F), etc. Cela devrait forcer le port à s'ouvrir et eviter le second problème de redirection.

14.4.3 Deux interfaces sur un terminal

En utilisant un shell (tel que bash) avec l'édition de la ligne de commande activée, il y a deux interfaces de terminal différentes (ce que vous voyez quand vous tapez stty -a). Quand vous tapez sur la ligne de commande vous avez une interface "brute" temporaire (mode brut, ou "raw") où chaque caractère est lu par l'éditeur de ligne de commande au moment où vous le tapez. Une fois que vous appuyez sur la touche <entrée>, l'éditeur de ligne de commande sort et l'interface du terminal est modifiée en interface nominale "améliorée" (mode amélioré ou "cooked") pour le terminal. Ce mode amélioré dure jusqu'à ce que l'invite suivante soit envoyée au terminal. Notez qu'on ne tape jamais rien dans ce mode "amélioré" mais ce qui a été tapé en mode raw passe en mode amélioré dès qu'on a tapé sur la touche <entrée>.

Quand une invite est envoyée au terminal, le terminal passe du mode "amélioré" au mode "brut" (comme il le fait quand vous démarrez un éditeur puisque vous démarrez l'éditeur de ligne de commande). Les paramètres pour le mode "brut" ne sont basés que sur les paramètres de base pris à partir du mode "amélioré". Le mode brut garde ces paramètres mais modifie plusieurs autres paramètres afin de passer en mode "brut". Il n'est pas du tout basé sur les paramètres utilisés dans le mode "brut" précédent. Ainsi si on utilise stty pour modifier les paramètres du mode brut, de tels paramètres seront perdus dès qu'on appuiera sur la touche <entrée> sur le terminal qu'on suppose avoir "configuré".

Maintenant, quand on utilise tape stty pour regarder l'interface du terminal, on peut avoir une vue soit du mode amélioré, soit du mode brut. Vous devez trouver lequel vous regardez. Si vous utilisez stty à partir d'un autre terminal pour vous occuper d'un terminal qui affiche une ligne de commande, vous aurez la vue du mode brut. Tout changement effectué ne le sera que pour le mode brut et sera perdu quand quelqu'un appuiera sur <entrée> sur le terminal que vous avez tenté de "paramétrer". Mais si vous tapez une commande stty sur votre terminal (sans utiliser < pour la redirection) et ensuite tapez sur <entrée>, c'est une histoire différente. <entrée> met le terminal en mode amélioré. Vos modifications seront sauvées et seront toujours présentes quand le terminal reviendra en mode brut (sauf, bien sûr, si c'est un paramètre non permis en mode brut).

Cette situation peut créer des problèmes. Par exemple, supposez que vous corrompez votre interface de terminal et que pour la récupérer vous alliez sur un autre terminal et tapiez "stty sane <dev/ttyS1" pour la récupérer. Ceci ne fonctionnera pas ! Bien sûr vous pouvez essayer de taper "stty sane ..." sur le terminal corrompu mais vous ne pouvez pas voir ce qui est tapé. Tout ce qui précède ne s'applique pas aux terminaux non-intelligents mais aux terminaux virtuels utilisés sur un moniteur de PC ainsi que sur les terminaux fenêtrés sous X. En d'autres termes, ceci s'applique à presque tout le monde qui utilise Linux. Heureusement, un fichier qui lance stty au démarrage s'occupera certainement d'un terminal (ou d'un port série sans terminal) n'ayant aucun shell tournant dessus, donc il n'y a pas de problème.

14.4.4 Où mettre la commande stty ?

Si vous avez besoin que `stty` configure l'interface série à chaque fois que l'ordinateur démarre, vous devez mettre la commande `stty` dans un fichier qui sera exécuté à chaque démarrage de l'ordinateur (de Linux). Il devrait être lancé avant l'utilisation du port série (ce qui comprend le lancement de getty sur le port). Il y a de nombreux endroits disponibles pour le mettre. S'il est mis à plus d'un endroit et que vous n'en connaissez (ou rappelez) qu'un, il y aura sûrement un conflit. Assurez-vous donc de documenter ce que vous faites.

Un bon endroit pour placer cette commande serait dans le même fichier qui lance setserial quand le système démarre. L'emplacement dépend des distributions et des versions. Il semblerait mieux de la placer après la commande setserial pour que la partie de bas niveau soit faite en premier. Si vous avez un répertoire dans le /etc où tous les fichiers sont éxecutés au démarrage (System V Init), ainsi vous pourriez créer un fichier nommé "stty" dans ce but.

14.5 Terminfo et Termcap (bref)

Voyez 15 (Terminfo et Termcap (en détails)) pour une discussion plus détaillée sur terminfo. Beaucoup d'applications que vous lancez utilisent la base de données terminfo (anciennement termcap). Celle-ci possède une entrée (ou fichier) pour chaque modèle ou type (tel que le vt100) de terminal et indique ce que le terminal peut faire, quels codes envoyer pour diverses actions, et quels codes envoyer au terminal pour l'initialiser.

Puisque beaucoup de terminaux (et de PC aussi) peuvent émuler d'autres terminaux et possèdent des "modes" d'opération variés, il peut y avoir plusieurs entrées terminfo parmi lesquelles choisir pour un terminal physique donné. Ils auront en général des noms similaires. Le dernier paramètre de getty (à la fois pour agetty et getty_ps) devrait être le nom terminfo du terminal (ou de l'émulation de terminal) que vous utilisez (comme vt100).

La base terminfo fait plus que simplement spécifier de quoi le terminal est capable et de donner les codes à envoyer au terminal pour le faire faire certaines choses. Elle spécifie à quoi "gras" ressemblera (sera-ce en vidéo inverse ou en intensité forte), comment sera le curseur, si les lettres seront noires, blanches ou d'une autre couleur, etc. En terminologie PC on appelle ceci des "préférences". Elle spécifie les codes d'initialisation à envoyer au terminal (analogues aux chaînes d'initialisation qu'on envoie aux modems). Linux n'envoie pas automatiquement de telles chaînes au terminal. Voyez 15.5 (chaîne d'initialisation). Si vous n'aimez pas l'affichage à l'écran ni son comportement, vous devrez peut-être éditer (et ensuite mettre à jour) le fichier terminfo (ou termcap). Voyez 15.2.3 (compilateur terminfo (tic)) sur la manière de faire la mise à jour.

14.6 Positionner TERM et TERMINFO

Voici deux variables d'environnement pour les terminaux : TERM et TERMINFO, mais vous ne devriez rien avoir à faire avec elles. TERM doit toujours être positionnée au nom du terminal que vous utilisez (comme vt100). Si vous ne connaissez pas son type (nom), voyez 14.6.1 (quel est le nom terminfo de mon terminal ?). TERMINFO contient le chemin vers la base de données terminfo, mais peut ne pas être nécessaire si la base de données est dans un endroit prédéfini (ou TERMINFO peut être positionné automatiquement par un fichier qui est livré avec votre distribution de Linux). Vous voudrez voir 15.2.2 (Emplacement des bases de données compilées).

Heureusement, le programme getty positionne en général TERM pour vous juste avant le login. Il utilise juste le type de terminal qui a été spécifié sur la ligne de commande de getty (dans /etc/inittab). Cela permet aux applications de trouver le nom de votre terminal et ensuite de regarder les capacités du terminal dans la base de données terminfo. Voyez 15.6 (variable TERM) pour plus de détails sur TERM.

Si votre base de données terminfo ne peut pas être trouvée, vous verrez un message d'erreur à ce propos sur votre terminal. Si cela arrive il est temps de vérifier où réside terminfo et de positionner TERMINFO si nécessaire. Vous pouvez découvrir où se trouve la base de données terminfo en cherchant un fichier terminfo courant comme "vt100" avec la commande "locate". Assurez-vous que votre terminal est dans cette base de données. Un exemple de positionnement de TERMINFO : export TERMINFO=/usr/share/terminfo (mettez ceci dans /etc/profile ou autre). Si les données concernant votre terminal dans cette base de données ne vous conviennent pas, vous devrez l'éditer. Voyez 14.5 (terminfo et termcap (bref)).

14.6.1 Quel est le nom terminfo de mon terminal ?

Vous avez besoin du nom exact afin de positionner la variable d'environnement TERM ou pour renseigner `getty`. Le même nom doit être utilisé à la fois par la base termcap et la base terminfo, vous n'avez donc besoin de le trouver qu'une seule fois. Un terminal dispose généralement d'alias mais si vous trouvez plus d'un nom, utilisez le premier.

Pour le trouver, essayez de regarder le fichier /etc/termcap... (si vous l'avez). Sinon, regardez soit dans l'arborescence terminfo (voyez ref id="tc_compiled_locs" name="emplacement des bases de données compilées">), soit essayez de trouver le fichier de code source de terminfo (voyez ref id="tc_source_loc" name="emplacements du code sources des bases de données">).

14.7 Fichier /etc/ttytype rarement nécessaire

Le fichier de configuration /etc/ttytype est utilisé pour faire la correspondance entre /dev/ttySn et les noms de terminaux comme dans terminfo. tset l'utilise, mais si la variable d'environnement TERM est déjà positionnée correctement, alors ce fichier n'est pas nécessaire. Puisque le getty de Linux positionne TERM pour chaque tty, vous n'avez pas besoin de ce fichier. Dans d'autres systèmes Unix comme FreeBSD, le fichier /etc/ttys fait la correspondance entre les ttys et bien plus de choses, comme la commande getty appropriée, et la catégorie de connexion (comme "dialup"). Un exemple de ligne pour le ttytype sous Linux : vt220 ttyS1

14.8 Restrictions sur les logins

Par défaut, l'utilisateur root ne peut pas se logger à partir d'un terminal. Pour permettre cela vous devez créer (ou éditer) le fichier /etc/securetty en suivant la page de manuel "securetty". Mais cette utilisation est spécifique à la distribution, la Suse n'utilise pas /etc/securetty. Pour restreindre les logins de certains utilisateurs et/ou de certains terminaux etc., éditez /etc/login.access (cela remplace le vieux fichier /etc/usertty ??). /etc/login.defs détermine si /etc/securetty doit être utilisé et peut être édité afin que /etc/securetty ne soit pas nécessaire (ou utilisé). /etc/porttime restreint les heures auxquelles certains ttys et utilisateurs peuvent utiliser l'ordinateur. S'il y a trop de tentatives de login ratées pour un utilisateur, cet utilisateur peut se voir interdire l'accès au système. Voyez la page de manuel "faillog" sur la manière de contrôler cela.

14.9 Lancer des commandes uniquement si TERM=mon_terminal

Il y a parfois des commandes qu'on ne veut exécuter au démarrage que pour un certain type de terminal. Faire cela pour la commande stty ne pose pas de problèmes puisque l'on utilise l'opérateur de redirection < pour spécifier le terminal vers lequel la commande est destinée. Mais quid des alias de shell ou des fonctions ? Vous aurez envie de créer une fonction pour la commande ls qui mettra en couleur la liste des répertoires uniquement sur des terminaux couleur ou sur la console. Pour les terminaux monochromes vous voudrez le même nom de fonction (mais un corps de fonction différent) qui utilisera des symboles à la place du codage par couleurs. Où mettre de telles définitions de fonctions qui doivent être différentes pour des terminaux différents ?

Vous pouvez les mettre à l'intérieur d'opérateurs "if" dans /etc/profile qui est lu au départ à chaque fois que quelqu'un se logge. L'opérateur confitionnel "if" définit certaines fonctions etc., seulement si le terminal est d'un type spécifique.

14.9.1 Exemple pour la fonction ls

Bien que la plupart de ce que fait cet opérateur if puisse être fait dans le fichier de configuration de dircolors, voici un exemple dans le cas du shell bash :

```
if [ "$TERM" = linux ]; then
    eval `dircolors`;
elif [ "$TERM" = vt220 ]; then
    ls () { command ls -F $* ; }
```

```
# pour exporter la fonction ls():
    declare -xf ls
else echo "De /etc/profile : terminal de type $TERM inconnu"
fi
```

15 Terminfo et Termcap (en détails)

15.1 Introduction à Terminfo

Terminfo (anciennement termcap) est une base de données des capacités des terminaux et plus. Pour chaque (enfin presque) modèle de terminal elle indique aux applications ce que le terminal est capable de faire. Elle indique quelles séquences d'échappement (ou caractère de contrôle) envoyer au terminal afin de faire des choses telles que déplacer le curseur vers un nouvel endroit, effacer une partie de l'écran, faire défiler l'écran, changer de mode, changer l'apparence (couleurs, luminosité, clignotement, soulignement, vidéo inverse, etc.). À partir de 1980 environ, beaucoup de terminaux supportaient plus d'une centaine de commandes (certaines d'entre elles prenant des paramètres numériques).

La façon dont terminfo donne les information a un programme est par les fonctions "ncurses" que le programmeur met dans un programme C. Par exemple si un programme veut bouger le curseur à la ligne 3, colonne 6, il appelle simplement: move(3,6). La fonction move() (faisant partie de ncurses) sait comment faire pour votre terminal (elle peut lire terminfo). Donc il envoit la sequence d'échappement appropriée au terminal qui lui fait faire ce mouvement.

Les abréviations terminfo sont en général plus longues que celles de termcap et il est ainsi plus facile de deviner ce qu'elles veulent dire. Les pages de manuel de terminfo sont plus détaillées (et incluent les anciennes abréviations de termcap). Ainsi, sauf si vous êtes déjà obligé de travailler avec termcap, il est suggéré que vous utilisiez les fichier terminfo.

15.2 Base de données terminfo

15.2.1 Introduction

La base de données terminfo est compilée et possède ainsi une partie source et une partie compilée. La vieille base de données termcap ne possède qu'une partie source mais cette source peut, grâce à une seule commande, être à la fois convertie en source terminfo et ensuite compilée. Vous pouvez ainsi vous en tirer sans avoir le source terminfo puisque le source termcap peut créer la base terminfo compilée. Pour voir un affichage de la base de données pour le terminal que vous utilisez actuellement (y compris un moniteur de PC), tapez "infocmp" et vous devriez voir le "fichier" source terminfo correspondant.

Pour voir si votre terminal (disons vt100) est dans la base de données terminfo, tapez "locate vt100". Si vous ne connaissez pas le nom de votre terminal, explorez la liste des fichiers dans la base de données compilée ou voyez 14.6.1 (quel est le nom terminfo de mon terminal ?).

15.2.2 Où se trouve la base de données ?

Emplacements de la base de données compilée Le fait de taper "locate vt100" peut vous indiquer /usr/lib/terminfo/v/vt100, /usr/share/terminfo/v/vt100, /home/vous/.terminfo/v/vt100 et/ou /etc/terminfo/v/vt100. Tout ceci sont des localisations possibles des fichiers terminfo compilés. Quoique le répertoire /etc/terminfo ne soit pas l'emplacement standard pour ca, avoir quelques types de terminal a cet endroit peut être pratique au cas ou le répertoire /usr ne soit pas accessible. Par exemple, /usr peut être placé sur un disque ou une partition qui n'ait pas pue être montée. Normalement, les applications qui utilisent votre base de donnée terminfo principale sont capables de les trouver si il est mis dans au moin un des emplacements mentionnés avant. Autrement la variable d'environnement TERMINFO devrait être positionnée sur le chemin vers cette base de données. Exemple : TERMINFO=/usr/share/terminfo.

Si le terminfo compilé se trouve à plus d'un endroit, tout se passe en général bien jusqu'à ce que quelqu'un récupère un ou plusieurs fichiers terminfo (d'une distribution plus récente, d'Internet, en éditant la plus vieille, etc.). Le nouveau terminfo doit être placé à tous les emplacements existants (ou les emplacements redondants doivent être abolis). Si vous ne vous assurez pas que c'est fait, alors plusieurs applications peuvent se retrouver à chercher et utiliser les anciennes données terminfo (boguées) qui existent encore à un emplacement "traditionnel".

Emplacements du code source des bases de données Le code source que vous utilisez peut résider dans /etc/termcap et/ou dans terminfo.src (ou un autre nom). Voyez les pages de manuel : terminfo(5) ou termcap(5) pour voir le format nécessaire pour créer (ou modifier) ces fichiers source. Le fichier terminfo.src peut se trouver en divers endroits sur votre ordinateur ou peut ne pas être inclus dans votre distribution Linux. Utilisez la commande locate pour essayer de le trouver. Il est disponible pour téléchargement (sous le nom termtypes.ti) à `http://sagan.earthspace.net/terminfo` .

15.2.3 Compilateur terminfo (tic)

Les données des fichiers sources sont compilées avec le programme "tic" qui est capable de faire les conversions entre les formats termcap et terminfo. Vous pouvez ainsi créer une base de données terminfo compilée à partir d'un source termcap. Le programme d'installation utilisé pour installer Linux a probablement installé les fichiers compilés sur votre disque dur donc vous ne devez rien compiler sauf si vous modifiez /etc/termcap (ou terminfo.src). "tic" installera automatiquement les fichiers compilés résultant dans un répertoire terminfo prêt à être utilisé par les applications.

15.2.4 Regardez votre terminfo

C'est une bonne idée de jeter un coup d'oeil à votre entrée terminfo pour le terminal que vous utilisez (le code source, bien sûr) et de lire les commentaires. Une manière rapide de l'inspecter sans les commentaires est de taper simplement "infocmp". Mais les commentaires pourront vous dire des choses spéciales sur le terminal comme la manière dont vous devez le configurer pour qu'il fonctionne correctement avec la base de données terminfo.

15.2.5 Effacer des données non nécessaires

Afin d'économiser de l'espace disque, on peut effacer toute la base de données à part les types de terminaux que l'on possède (ou dont on pourrait avoir besoin dans le futur). N'effacez aucun termcap pour un "terminal Linux" (la console) ou les entrées xterm si vous utilisez X Window. Le type de terminal "dumb" peut être nécessaire quand une application ne peut pas déterminer le type de terminal que vous utilisez. Cela économiserait de l'espace disque si les programmes d'installation n'installaient les terminfo que pour les terminaux que vous possédez et que vous puissiez obtenir un termcap pour un nouveau terminal sur Internet en quelques secondes.

15.3 Bogues dans les fichiers terminfo existants (et dans le matériel)

Malheureusement, il y a un certain nombres de bogues dans les fichiers terminfo et termcap. En plus, bien des définitions ne sont pas complètes et ne définissent pas certaines capacités disponibles sur les terminaux.

Parfois vous pouvez vous en tirer sans modifier terminfo mais dans d'autres cas vous devez le modifier ou peut-être utiliser une autre émulation qui possède un bon terminfo.

Le mauvais état des fichiers terminfo fournis est dû à un certain nombre de raisons. L'une d'entre elles est que durant les années 1980, quand on en a écrit beaucoup (souvent au format termcap), les applications n'utilisaient les capacités avancées des terminaux. Ainsi, si une telle capacité n'était pas dans le fichier termcap (ou terminfo), personne ne se plaignait. Aujourd'hui, les programmes tels que vim utilisent la "colorisation contextuelle" et minicom utilise le code de caractère graphique du terminal. Ceux-ci ont souvent besoin de plus de définitions à ajouter à l'ancien termcap. Ceci peut (ou peut ne pas) avoir déjà été fait.

La plupart des terminaux avaient des bogues matériels (dans leur programme interne) qui étaient parfois "réparés" en modifiant le termcap. Ensuite, le fabricant pouvait envoyer des puces de remplacement qui réparaient le bogue. Tous les propriétaires ne passaient pas leur temps à obtenir les puces de remplacement. Ainsi il peut y avoir deux terminfos ou plus pour votre terminal, selon les puces qu'il possède. Cette situation n'était pas toujours notée dans le termcap et un seul termcap peut être fourni avec Linux. Certains bogues matériels présents pour des capacités qui n'étaient quasiment jamais utilisées dans le passé n'ont sûrement jamais été réparés. De plus, certaines bogues matériels rapportés peuvent ne jamais avoir été réparés puisqu'ils n'étaient pas très importants à l'époque ou que la société est décédée, etc.

15.4 Modification des fichiers terminfo

Pour faire ceci vous avez besoin du manuel de votre terminal indiquant les séquences d'échappement qu'il utilise. Les manuels récents des années 1990 ne les montrent pas souvent. Vous avez aussi besoin d'un manuel terminfo (ou similaire). Par exemple, afin d'ajouter des capacités graphiques vous devez assigner des valeurs aux variables terminfo : enacs, rmacs et smacs en éditant un fichier source. Alors en utilisant "tic" vous pouvez le compiler. "tic" devrait placer automatiquement le fichier terminfo compilé dans le répertoire correct réservé à cet usage.

Si vous voulez trouver un meilleur terminfo pour un certain terminal que celui qui est fourni, vous pouvez essayer de chercher sur Internet (mais ce que vous y trouverez peut être pire). Si votre nouvelle entrée terminfo est meilleure que l'ancienne et semble stable (vous l'avez utilisée pendant un moment sans problèmes) vous devriez en envoyer une copie au mainteneur de terminfo comme indiqué au début du fichier source de terminfo (ou termcap).

15.5 Chaîne d'initialisation

Dans le terminfo sont souvent incluses des chaînes d'initialisation qu'on peut envoyer au terminal pour l'initialiser. Cela peut modifier l'apparence de l'écran, changer le mode dans lequel se trouve le terminal et/ou faire que le terminal émule un autre terminal. Une chaîne d'initialisation n'est pas envoyée automatiquement au terminal pour l'initialiser. On pourrait espérer que le programme getty le fasse mais s'il le faisait, on pourrait faire un changement de configuration sur le terminal et ce changement ne serait pas pris en compte parce que la chaîne d'initialisation l'annulerait automatiquement. Vous devez utiliser une commande sur la ligne de commande (ou dans un script shell) pour envoyer la chaîne d'initialisation telle quelle. De telles commandes sont : "tset", "tput init" ou "setterm -initialize". Parfois il n'y a pas besoin d'envoyer la chaîne d'initialisation puisque le terminal peut se configurer correctement quand il est allumé (en utilisant les options et préférences qu'on a sauvées dans la mémoire permanente du terminal).

15.6 Variable TERM

La variable d'environnement TERM devrait être initialisée au nom de terminal que vous utilisez. Si TERM n'a pas encore été positionnée et que vous ne connaissez pas le nom de votre terminal, voyez 14.6.1 (quel est le nom terminfo de mon terminal ?). Elle est normalement positionnée par le paramètre terminal_type passé au programme getty (regardez-le dans le fichier /etc/inittab). Ce nom doit se trouver dans la base de données terminfo. Tapez simplement "set" sur la ligne de commande pour voir quelle valeur a TERM (ou tapez : tset -q). Sur la console (moniteur) TERM est positionné à "linux" qui représente le moniteur du PC émulant un modèle de terminal fictif appelé "linux". Puisque "linux" est proche d'un terminal vt100 et que beaucoup de terminaux le sont aussi, l'appellation "linux" fonctionnera parfois comme un recours temporaire sur un terminal texte.

Si on peut connecter plus d'un type de terminal sur le même port (/dev/tty...) (par exemple, si un commutateur permet à différents types de terminaux d'utiliser le même port série, ou si le port est relié à un modem que des personnes appellent depuis différents types de terminaux) alors TERM doit être positionné à chaque fois que quelqu'un se connecte sur le port série. Il y a souvent une séquence d'échappement de requête pour que l'ordinateur puisse demander au terminal de quel type il est. Une autre façon est de demander à l'utilisateur de taper (sélectionner) le type de terminal qu'il ou elle utilise. Vous aurez peut-être besoin d'utiliser tset ou d'écrire un petit script shell pour gérer cela.

Une manière est d'utiliser "tset" (voir la page de manuel). tset essaie de déterminer le nom du terminal à partir du terminal que vous utilisez. Il regarde ensuite les données dans terminfo et envoie une chaîne d'initialisation à votre terminal. Il peut aussi positionner la valeur de TERM. Par exemple, un utilisateur appelle et se logge. Le script de login .profile est exécuté et il contient la commande suivante : eval 'tset -s ?vt100'. Ceci fait que : on demande à l'utilisateur s'il ou elle utilise un vt100. L'utilisateur répond oui ou bien tape le type de terminal réel qu'il ou elle utilise. tset envoie ensuite la chaîne d'initialisation et positionne TERM à ce nom (type) de terminal.

15.7 Documents sur terminfo/termcap

- pages de manuel de terminfo(5) (la meilleure) et/ou termcap(5). *Le manuel Termcap* `http://www.delorie.com/gnu/docs/termcap/termcap_toc.html` (2ème éd.) par Richard M. Stallman est un manuel GNU qui est quelque peu obsolète depuis qu'il n'inclut plus terminfo.
- les fichiers : terminfo.src et /etc/termcap possèdent des informations sur les diverses versions des fichiers termcap, les conventions de nommage pour les terminaux et des codes de capacités spéciales nommées u6-u9. Si vous ne les avez pas, allez à `http://sagan.earthspace.net/terminfo`
- "Termcap et Terminfo" est un livre publié par O'Reilly en 1988.

16 Utilisation du terminal

16.1 Introduction à l'utilisation du terminal

Cette section parle du contrôle de l'interface terminal-ordinateur et/ou du changement de configuration du terminal pendant son utilisation. Elle explique (ou pointe vers des explications sur) la manière dont l'utilisateur d'un terminal peut contrôler et inspecter l'interface, et comment utiliser diverses commandes fournies par le pilote de périphérique. Elle n'explique pas comment utiliser les nombreuses applications, shells ou la plupart des utilitaires Linux. Deux commandes utilisées couramment sur un terminal sont :

- clear (pour effacer l'écran)

- reset (pour réinitialiser le terminal)

16.2 Démarrer le terminal

Évidemment il faut allumer le terminal pour qu'il fonctionne. Si vous ne voyez pas d'invite de login, appuyez sur la touche "Retour" (ou "Entrée") plusieurs fois. Saisissez ensuite le nom de votre compte (suivi d'un retour/entrée) et votre mot de passe quand on vous le demande (suivi aussi d'un retour/entrée). Prenez garde à ne pas tout taper en lettres majuscules. Si vous le faites, l'ordinateur peut croire que vous avez un vieux terminal qui ne peut pas transmettre de lettres minuscules et le pilote série peut se configurer pour n'envoyer que des lettres majuscules au terminal.

Si rien ne se passe, assurez-vous que l'ordinateur hôte va bien. Si l'ordinateur hôte est éteint (pas de courant), ce que vous tapez sur le clavier du terminal peut apparaître à l'écran puisque les broches de transmission et de réception sur l'ordinateur peuvent être reliées ensemble, ce qui fait que les caractères sont retournés par un ordinateur "éteint". Si vous ne pouvez pas vous logger quand l'ordinateur fonctionne, voyez 17 (résolution des problèmes).

16.3 Pilote de périphérique (série) du terminal

En tapant sur la ligne de commande, le shell (tel que le shell Bash) lit ce que vous tapez et y réagit. Ce que vous tapez passe d'abord par la partie pilote de terminal de votre système d'exploitation. Ce pilote peut traduire certains caractères (comme changer le caractère "retour" généré par la touche "retour" en un caractère "nouvelle ligne" pour les fichiers Linux). Il reconnaît aussi certains codes de contrôle que vous pourriez taper au clavier comme ^C pour interrompre l'exécution d'un programme. Il retourne normalement ce que vous tapez à l'écran. On peut utiliser 14.4 (stty) pour configurer le comportement de ce terminal, ce qui comprend l'arrêt de tout ou partie de cette fonctionnalité.

16.4 Problèmes avec les éditeurs

Il peut y avoir quelques problèmes pendant l'utilisation d'emacs et de vi sur certains terminaux.

16.4.1 Emacs et ^S,^Q

Si le contrôle de flux logiciel est présent, la commande ^S dans Emacs bloquera l'affichage. La commande ^Q débloquera l'affichage. La solution est de relier cette commande à une autre touche dans le fichier de configuration de Emacs.

16.4.2 Vi et les touches curseur

Vi utilise la touche ESC comme commande pour sortir du mode d'insertion. Si on appuie sur une touche fléchée (touche de curseur) une séquence d'échappement (démarrant par le caractère ESC) est envoyée à l'hôte. Vi doit faire la différence entre ces deux significations d'ESC. Un vi intelligent (comme vim) doit être capable de détecter la différence en regardant le temps entre la touche ESC et celle qui suit. Si il s'agit d'un court délai, c'est comme si une touche curseur avait été pressée. Utilisez "help cursor-keys" dans vim pour en savoir plus.

Il y'a une autre façon d'arranger ça.Sur les terminaux VT on peut faire en sorte que la touche flèche gauche envoie soit ESC [D soit ESC O D. Les autres touches fléchées sont similaires mais utilisent A, B et C au lieu de D. Si vous avez des problèmes, choisissez ESC [D puisque le "O" dans l'autre alternative peut être interprétée comme une commande d'"ouverture de ligne". Le "[" devrait être interprété par vi pour dire qu'une touche fléchée a été pressée. ESC [D sera envoyé à condition que le "mode application touches fléchées" (Cursor Key Application Mode) n'ait pas été activé. ESC [D est normalement la valeur par défaut donc tout devrait être correct. Sauf que de nombreux termcaps contiennent une chaîne (pas la chaîne d'initialisation) qui positionne ce que vous voulez éviter : "Mode Application". Les éditeurs peuvent envoyer cette chaîne au terminal quand ils démarrent. Là, vous avez des problèmes.

Cette chaîne possède le code termcap "ks" (smkx dans terminfo) qui veut dire activer les touches de fonctions (et similaires, incluant les touches fléchées). Une application active ces touches en envoyant la chaîne "ks" au terminal. La personne qui a écrit le termcap a conclu que si une application voulait activer ces touches, elles devraient être mises en "Mode Application" puisque c'est une "application", mais vous ne voulez pas cela.

La console Linux n'a pas de chaîne "ks" pour que vous ne tombiez pas dans ce piège sur la console. Pour d'autres terminaux vous aurez besoin d'éditer termcap (ou terminfo) ou d'utiliser une autre entrée termcap. Vous devez modifier non seulement la chaîne "ks" mais aussi les définitions termcap de ce qu'elle envoie : kd, kl, kr, ku. Lancez ensuite tic pour l'installer.

Pour vim (VI aMélioré) il y a une manière de le configurer pour qu'il fonctionne correctement avec ESC O D (pour que vous ne deviez pas éditer termcap) : cherchez "vt100-cursor-keys" dans l'aide de vim. Vous pouvez lancer "gitkeys" et ensuite appuyer sur les touches fléchées pour voir ce qu'elles envoient mais on peut les configurer pour qu'elles envoient autre chose quand vous êtes dans un éditeur.

16.5 Corruption du ls en couleur

Si `ls` met en l'air l'émulation de votre terminal avec la possibilité de couleur, arrêtez-la. `ls -color` et `ls -colour` utilisent la possibilité de couleur. Certaines installations font que `ls` utilise la couleur par défaut. Vérifiez dans `/etc/profile`, etc. s'il y a des alias pour `ls`. Voyez 14.9.1 (exemple de fonction ls) pour savoir comment faire pour que ls soit en couleur sur la console et en noir et blanc sur les terminaux.

16.6 L'affichage se bloque (terminal bloqué)

Les symptômes d'un terminal bloqué sont que ce que vous tapez ne s'affiche pas sur le terminal (ou dans certains cas s'affiche mais ne fait rien). Si ce que vous tapez est invisible (ou ne fait rien) tapez ^Q pour relancer le flux (si le contrôle de flux l'a stoppé). Le blocage peut aussi venir de :
16.7.2 (envoi d'un binaire au terminal) ou 16.7.3 (un programme s'est terminé de façon anormale)
Si vous n'avez fait ni l'un ni l'autre, votre programme peut alors être boggé ou vous avez intéragi de manière fatale avec lui.

Si vous voulez quitter le programme que vous avez lancé et que vous ne pouvez pas le faire par les méthodes normales (certains programmes ont des touches spéciales que vous devez taper pour sortir) essayez de le tuer à partir d'un autre terminal en utilisant "top" ou "kill". Si le processus refuse de s'arrêter, vous pouvez essayer de lui envoyer un signal 9 à partir de top qui devrait le forcer à s'arrêter. Le type de sortie forcée "9" peut laisser certains fichiers temporaires qui traînent ainsi qu'une interface corrompue. Tuer le shell de login devrait relancer getty avec une nouvelle invite de login.

Les personnes débutantes sous Linux peuvent sans le faire exprès appuyer sur Ctrl-S (^S) (ou la touche "Arrêt Défil") qui bloque l'écran de manière mystérieuse (bien que ce soit ce que cette touche est supposée faire si vous utilisez le contrôle de flux logiciel). Pour retrouver une interaction normale avec l'écran, pressez Ctrl-Q (^Q). Notez que tout ce qui est tapé durant le "blocage" est exécuté mais vous n'en verrez rien avant d'appuyer sur ^Q. Ainsi quand il est bloqué, ne tapez rien de spécial qui pourrait effacer des fichiers etc. L'un des arguments en faveur du contrôle de flux matériel est qu'il empêche de tels blocages.

16.7 Interface du terminal corrompue

Cela inclut le cas de l'"affichage bloqué" = "terminal stoppé net" de la section précédente.

16.7.1 Symptômes

Quand l'affichage ne semble pas correct, ou quand ce que vous tapez ne s'affiche pas correctement (si même un affichage se produit), ou que rien ne se passe quand vous tapez une commande, il y a des chances que vous soyez en face d'une corruption de l'interface du terminal. Dans les cas rares où la partie matérielle du port série elle-même serait corrompue, le seul remède peut être de basculer l'interrupteur (éteindre le PC et redémarrer). Le problème peut provenir de choses telles qu'un bogue dans le programme que vous utilisez, une panne matérielle (ce qui inclut un défaut matériel obscur avec lequel vous pouvez normalement vivre) ou peut-être une configuration incorrecte. Si tout fonctionnait correctement mais que ça va soudainement mal, il se peut que l'interface ait été corrompue par une de vos actions. Vous pouvez avoir fait l'une de ces trois erreurs :

- 16.7.2 (Envoyer des données binaires au terminal)
- 16.7.3 (Stopper un programme de manière anormale)
- 16.6 (Taper Ctrl-S par erreur)

16.7.2 Envoyer des données binaires au terminal

Votre terminal modifiera ses caractéristiques si on lui envoie certaines séquences d'échappement ou des caractères de contrôle. Si vous essayez par inadvertance d'afficher un fichier binaire, il peut contenir par hasard de telles séquences qui peuvent placer votre terminal dans un mode de fonctionnement étrange voire le rendre inutilisable. Visualisez ou éditez toujours un fichier binaire avec des programmes faits à cet effet pour que cela n'arrive pas. La plupart des éditeurs et des afficheurs manipuleront les binaires de la bonne manière afin de ne pas corrompre l'interface. Certains peuvent afficher un message vous avertissant qu'ils ne peuvent éditer du binaire. Par contre, l'utilisation de "cat ...." ou "cp .... /dev/tty.." où .... est un fichier binaire enverra le binaire au terminal et risque fortement de générer des problèmes.

La corruption peut aussi arriver en utilisant un programme de communication où un ordinateur distant peut envoyer des données binaires à l'écran. Il y a de nombreuses autres façons pour lesquelles ces problèmes peuvent arriver, alors soyez-y préparé. Même un fichier qu'on prend pour un fichier ASCII peut contenir des codes de contrôle indésirables.

Pour résoudre ce problème, ré-initialisez le terminal. Vous pouvez essayer de taper soit "reset" soit "setterm -reset" (bien que vous ne puissiez pas voir ce que vous tapez). Ceci enverra la chaîne de ré-initialisation à partir de l'entrée du terminal dans la base terminfo. Si la configuration correcte a été sauvée à l'intérieur du terminal, alors l'appui sur certaine(s) touche(s) (peut-être en mode de configuration) peut retrouver ce paramétrage. Vous voudrez alors ensuite encore utiliser "reset" pour envoyer la chaîne d'initialisation si vous l'utilisez pour configurer votre terminal.

16.7.3 Terminer un programme de façon anormale

De grandes applications (comme des éditeurs) utilisent souvent les commandes stty (ou autres) dans leur code pour modifier de manière temporaire la configuration stty pendant que vous utilisez ce programme. Cela peut mettre le pilote de périphérique en mode "brut" pour que chaque caractère que vous tapez aille directement à l'application. L'écho de retour que fait le pilote est désactivé pour que tout ce que vous voyez à l'écran vienne directement de l'application. Ainsi de nombreuses commandes de contrôle (comme ˆC) peuvent ne pas fonctionner dans certaines applications.

Quand vous quittez de telles applications, celle-ci remet d'abord en place les paramètres stty aux valeurs qu'ils avaient avant le démarrage de l'application. Si vous quittez le programme de manière anormale (vous pouvez deviner que ça s'est passé ainsi quand ce que vous tapez ne s'affiche plus à l'écran) vous risquez d'être encore en mode "brut" sur la ligne de commande.

Pour sortir du mode brut et revenir aux paramètres stty normaux, tapez "stty sane". Cependant, vous devez taper ceci juste après un "retour" et le terminer par un "retour". Mais l'appui sur la touche "retour" ne donne pas le résultat escompté puisque le code "retour" n'est plus traduit en caractères nouvelle ligne que le shell attend. Tapez donc simplement nouvelle ligne (ˆJ) à la place de "retour". L'interface de terminal "sane" peut ne pas être exactement la même que d'habitude mais elle fonctionne en général. "stty sane" peut aussi être utile pour sortir d'une interface corrompue par d'autres causes.

16.8 Caractères (de contrôle) spéciaux

Un certain nombre de caractères de contrôle que vous pouvez taper au clavier sont "attrapés" par le pilote de terminal et effectuent diverses tâches. Pour voir ces commandes de contrôle tapez : stty -a et regardez les lignes 2 à 4. Elles sont expliquées de manière vague dans les pages de manuel de stty. On peut modifier les caractères ou les désactiver en utilisant la commande stty. Ainsi vos caractères de contrôle peuvent différer de ceux décrits ci-dessous. On les utilise pour l'édition de la ligne de commande, l'interruption, le défilement et pour se déplacer sur le caractère suivant de manière transparente.

16.8.1 Édition de la ligne de commande

Alors que le pilote de terminal possède quelques commandes pour l'édition de la ligne de commande, certains shells possèdent un vrai éditeur intégré (comme "readline" dans le shell Bash). Un tel éditeur est normalement activé par défaut donc vous n'avez besoin de rien faire pour l'activer. S'il est disponible vous ne devez pas apprendre les commandes suivantes bien qu'elles fonctionnent souvent en plus de l'éditeur de lignes de commande. Les plus importantes à apprendre sont ˆC (interruption), ˆD et comment arrêter le défilement.

- Delete-key (touche d'effacement, que stty montre comme ˆ?) efface le dernier caractère
- ˆU détruit (efface) la ligne
- ˆW efface un mot en arrière
- ˆR réaffiche la ligne. Utile principalement sur les terminaux à sortie papier ??

16.8.2 Interruption (et Quit, Suspend, EOF, Flush)

- ˆC interrompt. Quitte le programme et vous remet sur l'invite de la ligne de commande.
- ˆ/ quitte. Comme l'interruption ˆC mais plus faible. Fait aussi générer un fichier "core" (dont vous n'avez probablement pas l'utilité) dans votre répertoire de travail).
- ˆZ suspend. Stoppe le programme et le met en tâche de fond. Tapez fg pour le relancer.
- ˆD fin de fichier. S'il est tapé sur l'invite de la ligne de commande, quitte le shell et va là où vous étiez avant que le shell démarre.
- ˆO chasser. Pas implémenté sur Linux. Envoie la sortie vers /dev/null.

16.8.3 Arrêt et reprise du défilement

Si ce que vous désirez voir défile hors du bas de l'écran, vous pouvez empêcher cela en envoyant un signal d'arrêt "stop" (^S ou Xoff) à l'hôte (à condition que le 10 (contrôle de flux) Xon-Xoff soit activé). Envoyez un signal de départ "start" (^Q ou Xon) pour reprendre. Certains terminaux possèdent une touche "Pas de défilement" qui enverra de manière alternée Xoff et Xon ou peut-être enverra des signaux de contrôle de flux matériel ?? Voici ce que font ctrl-S (^S) et ctrl-Q (^Q) :

- ^S arrête le défilement (Xoff)
- ^Q reprend le défilement (Xon)

Si vous voulez à la fois arrêter le défilement et quitter, utilisez ^C. Si vous voulez arrêter le défilement pour faire autre chose mais voulez garder le programme qui générait la sortie en mémoire pour que vous puissiez reprendre le défilement plus tard, utilisez ^Z pour suspendre.

Une méthode de défilement différente est d'envoyer la sortie dans un tube vers un afficheur comme more, less ou most. Cependant, la sortie peut ne pas être la sortie standard mais peut être la sortie d'erreur que l'afficheur ne reconnaît pas. Pour résoudre ceci vous pouvez utiliser une redirection "2>&1" pour que l'afficheur fonctionne correctement. Il est souvent plus simple d'utiliser simplement ^S et ^Q sauf si vous devez défiler en arrière.

Sur une console PC (qui émule un terminal), vous pouvez défiler en arrière en utilisant Shift-PageHaut. Cela est fréquemment nécessaire puisque le défilement est souvent trop rapide à arrêter en utilisant ^S. Une fois que vous avez défilé en arrière Shift-PageBas défilera en avant à nouveau.

16.8.4 Prendre littéralement en compte le caractère suivant

^V envoie le caractère tapé suivant (en général un caractère de contrôle) directement au pilote de périphérique sans action ou interprétation. En retour deux caractères ASCII comme ^C sont affichés.

16.9 Visualiser des fichiers Latin-1 sur un terminal 7 bits

Certains fichiers texte sont au format Latin1 sur 8 bits (voyez 8.5 (ensembles de caractères)). Si vous avez un terminal qui n'affiche pas les caractères Latin1 (ou sur lequel on n'a pas sélectionné l'ensemble de caractères Latin1), un symbole de boulet s'affichera comme un 7, etc. En visualisant des pages de manuel (elles sont en Latin1) vous pouvez passer l'option -7 à man afin de traduire les 7, etc. à quelque chose proche d'un boulet (en ASCII). Y a-t-il des afficheurs qui font ces traductions ??

16.10 Inspection de l'interface

Ces utilitaires vous fourniront des informations sur l'interface du terminal :

- gitkeys : montre quel(s) octet(s) chaque touche envoie à l'hôte.
- tty : montre à quel port tty vous êtes connecté.
- set : montre la valeur de TERM (le nom de l'entrée terminfo)
- stty -a : montre tous les paramètres stty.
- setserial -g /dev/tty?? (remplissez les ??) montre le type d'UART, l'adresse du port et le numéro d'IRQ.
- infocmp : montre l'entrée de terminfo en cours (moins de commentaires)

16.11 Modifier les paramètres du terminal

Les paramètres du terminal sont normalement positionnés une fois quand le terminal est installé en utilisant les procédures de configuration du manuel du terminal. Cependant, certains paramètres peuvent être modifiés alors que le terminal est en cours d'utilisation. Vous ne donnez normalement aucune commande "stty" ou "setserial" quand le terminal est en cours d'utilisation car elles causeront certainement des problèmes à l'interface du terminal. Cependant, vous pouvez faire certaines modifications à l'apparence de l'écran du terminal ou à son comportement sans détruire l'intégrité de l'interface. Parfois ces modifications sont faites automatiquement par les applications et donc vous ne devriez pas avoir besoin de vous en occuper.

Une méthode directe pour effectuer de telles modifications est d'utiliser la touche de configuration (ou autre) sur le terminal et ensuite d'utiliser les menus pour faire les modifications. Pour cela, vous aurez besoin de bien connaître le terminal. Les trois autres méthodes envoient une séquence d'échappement de l'ordinateur vers le terminal pour faire les modifications. Ces trois exemples montrent des méthodes différentes pour faire cela en positionnant la vidéo inverse :

1. setterm -reverse
2. tput -rev
3. echo ^[[7m

16.11.1 setterm

C'est la commande la plus facile à utiliser. Elle utilise des options longues (mais n'utilise pas les – normaux devant). Elle consulte la base de données terminfo pour déterminer le code à envoyer. Vous pouvez modifier la couleur, la luminosité, la coupure de ligne, la vitesse de répétition du clavier, l'apparence du curseur etc.

16.11.2 tput

La commande "tput" est similaire à "setterm" mais au lieu d'utiliser des mots ordinaires comme arguments, vous devez utiliser les abréviations utilisées par terminfo. Beaucoup d'abréviations sont relativement laconiques et difficiles à retenir.

16.11.3 echo

Dans l'exemple "echo ^[[7m" pour positionner la vidéo inverse, ^[est le caractère d'échappement. Pour le taper, tapez ^V^[(ou ^V suivi de la touche ESC). Pour utiliser cette méthode "echo" vous devez trouver quel code utiliser à partir d'un manuel de terminal ou à partir de terminfo ou termcap. Il est plus facile d'utiliser setterm ou tput si vous tapez sur la ligne de commande. Puisque "echo ..." s'exécutera plus vite (puisqu'il ne fait de requête), il est bon pour être utilisé dans les scripts shell lancés au démarrage, etc.

16.11.4 Sauver les modifications

Quand vous éteignez le terminal les modifications que vous avez faites seront perdues (sauf si vous les avez sauvées dans la mémoire permanente du terminal en allant dans le mode configuration et en les sauvant). Si vous voulez les utiliser à nouveau sans devoir les retaper, mettez les commandes dans un script shell ou créez une fonction shell. Lancez-la ensuite quand vous voulez faire les modifications. Une manière de rendre les modifications semi-permanentes est de mettre ces commandes dans un fichier lancé à chaque fois que vous vous loggez ou que vous démarrez l'ordinateur.

16.12 Faire d'un terminal une console

C'est aussi ce que l'on appelle une "console série". Beaucoup de messages en provenance du système ne sont normalement envoyés que sur la console (Le moniteur). On peut aussi voir sur un terminal certains messages envoyés sur la console durant le démarrage après un démarrage réussi en tapant la commande : dmesg. Si le démarrage ne réussit pas, ceci ne sera d'aucun utilité. Il est possible de modifier le noyau Linux pour qu'un terminal serve de console et reçoive tous les messages de Linux destinés à la console. Malheureusement, les messages du BIOS (qui s'affiche sur le moniteur quand un PC est démarré) seront perdus puisqu'ils ne seront pas affichés sur le terminal.

Creer une "console série" veut dire que la console (dans la forme d'un terminal) se trouve maintenant sur un port série. Bien sûr, elle n'aura pas les couleurs, les fontes, les capacités graphiques, ni les possibilités de redéfinition du clavier de la vraie console du PC.

Avant le noyau 2.2, vous deviez patcher le noyau à la main. À partir du noyau 2.2, le support est inclus dans le noyau à condition que ce dernier ait été configuré pour cela. Ces deux cas sont traités dans les deux sous-sections suivantes :

16.12.1 Pour les noyaux 2.2 et supérieurs

Les instructions pour faire une console série sont incluses dans la documentation livrée avec le code source dans le fichier : serial-console.txt. Normalement, le périphérique /dev/console est un lien vers tty0 (la console PC). Pour une console série vous créez un nouveau /dev/console qui est un vrai périphérique (et non pas un lien vers autre chose). Vous devez aussi inclure une déclaration à propos de la console série dans /etc/lilo.conf et ensuite lancer lilo. C'est parce que l'équivalent de "setserial" doit être lancé pour configurer votre console série avant le chargement du noyau. Voyez la documentation ci-mentionnée pour plus de détails.

16.12.2 Pour les noyaux antérieurs à 2.2

Le Linux Journal d'avril 1997 avait un article sur la façon d'appliquer une rustine au noyau. Vous ajoutez quelques #define au début de src/linux/drivers/char/console.c :

```
<item> #define CONFIG_SERIAL_ECHO
<item> #define SERIAL_ECHO_PORT 0x2f8  /* Serial port address  */

Ce qui suit n'était pas dans l'article de Linux Journal. Dans les noyaux 2.+
(et précédents ??) vous avez aussi besoin de positionner la vitesse
d'émission (sauf si 9600 est convenable). Cherchez ces deux lignes :

serial_echo_outb(0x00, UART_DLM); /* 9600 baud */
serial_echo_outb(0x0c, UART_DLL);

Changez 0x0c dans la ligne ci-dessus en (selon la vitesse que vous
désirez) :

115200 baud: 0x01      19200 baud: 0x06        2400 baud: 0x30
 57600 baud: 0x02       9600 baud: 0x0c        1200 baud: 0x60
 38400 baud: 0x03       4800 baud: 0x18
```

Si vous utilisez la console pour sélectionner quel système d'exploitation démarrer (avec LILO), mais que vous voudriez le faire à partir d'un terminal, vous devez ajouter une ligne au fichier /etc/lilo.conf. Voyez la page de manuel de lilo.conf et cherchez la chaîne "serial=".

16.12.3 Puis-je lancer Linux sans moniteur (console PC) ?

Oui, en utilisant un terminal et en le faisant passer pour une console comme indiqué ci-dessus. Vous aurez sans doute quand même besoin d'une carte graphique puisque la plupart des BIOS en ont besoin pour démarrer le PC. Votre BIOS peut aussi avoir besoin d'un clavier pour démarrer, ou bien il peut avoir une option grâce à laquelle vous pouvez indiquer au BIOS qu'il n'a pas besoin de clavier.

16.13 Sessions multiples

Le paquet "`screen`" lance des sessions multiples un peu comme les terminaux virtuels sur la console : voyez 6.5 (la console : /dev/tty?). Cependant, ce n'est pas comme les "pages" (8.4 (section sur les pages)) puisque l'image des pages est stockée sur l'ordinateur hôte et non à l'intérieur du terminal comme elles le sont avec les "pages".

16.14 Se délogger

Pour vous délogger, tapez soit "logout", soit "exit". Dans certains cas votre demande sera refusée, mais on devrait vous dire pourquoi. L'une des raisons du refus est que vous n'êtes pas sur le même shell que celui avec lequel vous vous êtes loggé. Une autre manière de vous délogger est d'appuyer sur ˆD. Puisqu'on utilise aussi ˆD à d'autres fins, vous ne voudrez pas forcément que cette touche vous délogge. Si vous positionnez la variable IGNOREEOF dans le shell Bash, alors ˆD ne vous déloggera plus.

16.15 Discuter entre terminaux, espionner

Si deux personnes loggées par l'intermédiaire de terminaux sur le même ordinateur hôte désirent discuter ensemble, ils peuvent utiliser les programmes "write" ou "talk". Sur Internet, on peut discuter en utilisant le navigateur "lynx".

Pour espionner ce qu'une autre personne fait sur son terminal, utilisez le programme "ttysnoop". Dans "ttysnoop", les deux terminaux ont le même état et tout ce qui est tapé sur l'un des deux claviers apparaît sur les deux écrans (au même endroit). Donc si vous espionnez et ne voulez pas être détecté, vous ne devriez rien taper. On peut utiliser ttysnoop pour s'entraîner avec un professeur et un étudiant assis côte à côte, chacun sur son terminal. Le professeur peut regarder ce que tape l'étudiant et peut corriger les erreurs en tapant correctement. L'étudiant peut regarder ce que tape le professeur et ensuite le répéter lui-même. C'est comme s'ils utilisaient un seul terminal, les deux personnes ayant leurs mains sur le clavier en même temps.

17 Résoudre les problèmes (logiciels)

Si vous suspectez que le problème soit matériel, voyez la section 18 (réparation et diagnostic). Si le problème concerne le port série lui-même, voyez le Serial-HOWTO.

Voici une liste des problèmes possibles :

- 17.3 (Le terminal fonctionne-t-il ?) Soupçonnez que le terminal est en panne.
- 17.4 (Texte manquant) Soit passe au-dessus d'une partie du texte soit affiche correctement du texte puis se bloque
- 17.5 (Getty se relance trop rapidement) (message d'erreur sur la console)
- 17.6 (Ne fonctionne pas juste après le login)

- 17.7 (Ne peut pas se logger) mais l'invite de login est OK.
- 17.8 (Invite de login embrouillée)
- 17.9 (Aucun signe d'une invite de login)

Il y a deux cas dans lesquels le terminal se comporte mal. L'un arrive quand il a fonctionné correctement et s'est mis à mal fonctionner tout à coup. C'est ce dont on parle dans la sous-section suivante. L'autre cas arrive quand les choses vont mal juste après l'installation du terminal. Dans ce cas, vous pouvez passer à la section suivante.

17.1 Le terminal fonctionnait correctement

Quand un terminal qui fonctionnait correctement ne tourne tout d'un coup pas bien, il est souvent facile de déceler le problème. Si vous réfléchissez à ce qui s'est passé récemment cela vous donnera certainement un indice quant à la cause du problème.

Le problème peut être évident comme un message d'erreur au démarrage du terminal. S'il émet un bruit, il a sûrement besoin d'une réparation. Voyez 18 (réparation et diagnostics). D'abord, réflechissez à ce que vous avez fait ou modifié récemment car c'est sûrement la cause du problème. Est-ce que le problème est apparu juste après l'installation d'un nouveau logiciel ou après une modification de configuration ?

Si le terminal ne répond pas correctement (s'il répond tout court) à ce que vous tapez, vous avez peut-être une 16.7 (interface de terminal corrompue).

17.2 Terminal nouvellement installé

Si vous venez de relier un terminal à votre ordinateur en suivant les instructions et qu'il ne fonctionne pas, cette section vous concerne. Si un terminal qui fonctionnait correctement auparavant ne fonctionne plus, voyez 17.1 (le terminal fonctionnait). Si vous présumez que le port série de votre ordinateur est défectueux, vous pouvez essayer de lancer un programme de test et de diagnostics sur ce port. À présent (juin 1998) il semble que Linux ne dispose pas encore d'un tel programme de diagnostics et vous devrez donc lancer les diagnostics sous MS DOS/Windows. Il y a quelques programmes pour surveiller les diverses lignes série comme DTR, CTS, etc. et qui peuvent vous aider. Voyez 17.11 (surveillance/diagnostic série).

Une manière est d'abord de voir si le terminal fonctionne en essayant de copier un fichier vers le terminal (cp mon_fichier /dev/ttyS?) dans la situation la plus simple. Ceci implique la désactivation des lignes de contrôle du modem et à une vitesse qui ne nécessite pas de contrôle de flux (assurez-vous que le contrôle de flux matériel est désactivé). Si cette copie fonctionne, compliquez alors un petit peu la situation et voyez si ça fonctionne encore, etc., etc. Quand le problème apparaît juste après avoir fait une modification, alors ce changement est sûrement la cause du problème. En fait, il est plus efficace (mais plus compliqué) de sauter de la situation simple à à peu près la moitié de la configuration finale pour que le test élimine à peu près la moitié des causes possibles restantes pour le problème. Répétez alors cette méthode pour le test suivant. De cette manière il ne faudrait que dix tests environ pour trouver la cause sur un millier de causes possibles. Vous devriez vous écarter un peu de cette méthode en vous basant sur des intuitions et des indices.

17.3 Est-ce que le terminal va bien ?

Un bon terminal démarre en général en affichant quelques mots à l'écran. Si ces mots ne donnent aucun message d'erreur, le terminal va probablement bien. S'il n'y aucun signe de courant (aucune lumière n'est allumée, etc.), réenfoncez le câble d'alimentation des deux côtés. Assurez-vous qu'il y ait du courant sur la prise murale (ou au bout du cordon d'alimentation). Essayez un autre cordon si vous en avez un. Assurez-vous que le terminal est allumé et que son fusible n'a pas sauté. Un écran blanc (ou sombre) peut parfois être réparé simplement en tournant les molettes de luminosité et de contraste ou par une touche de clavier dans le mode de configuration. Si cela ne fonctionne toujours pas, voyez 18 (réparations et diagnostics) pour avoir des astuces sur la réparation du terminal.

Si le terminal démarre correctement, mais que vous soupçonnez qu'il y ait un problème, mettez-vous en "mode local" où il fonctionnera comme une machine à écrire et essayez de taper. Voyez 17.12 (mode local).

17.4 Texte manquant

Si le texte s'affiche normalement sur le terminal puis s'arrête sans avoir terminé (au milieu d'un mot, etc.) ou si des morceaux de texte manquent, vous avez sûrement un problème avec le contrôle de flux. Si vous ne pouvez pas trouver tout de suite ce qui le cause, baissez la vitesse. Si cela l'arrange, c'est sûrement un problème de contrôle de flux. Il se peut que le contrôle de flux ne fonctionne pas du tout à cause d'un manque de configuration correcte ou à cause d'un câblage incorrect (pour le contrôle de flux matériel). Voyez 10 (contrôle de flux).

Si des caractèrs isolés manquent, le port série est peut-être dépassé par une vitesse trop élevée. Essayez une vitesse plus petite.

Si vous utilisez une vitesse de transmission en dessous de 1200 (très lente, principalement utilisée par les anciens terminaux à copie papier et les imprimantes) et que le texte est tronqué, alors le problème peut provenir du pilote de périphériques série. Voyez le Printing-HOWTO à la section "périphériques série" sur la manière de régler ceci.

17.5 Getty se relance trop rapidement

17.5.1 Module série non chargé

Utilisez la commande "lsmod" pour voir si le module série est chargé.

17.5.2 Pas de tension de contrôle du modem

Si getty ne peut pas ouvrir et/ou utiliser un port à cause du manque de tension de contrôle de modem positive sur l'une des broches, alors getty peut se terminer. Alors, grâce aux instructions dans inittab, getty se relance et essaie encore, uniquement pour être terminé à nouveau, etc. etc. Vous pouvez voir un message d'erreur indiquant que, à cause de getty qui se relance trop rapidement, il a été temporairement désactivé. Essayez d'utiliser l'option "local" dans getty et/ou de vérifier les paramètres et les tensions de contrôle du modem.

17.5.3 Touche enfoncée

Une autre cause possible du relancement de getty est qu'une touche du clavier soit enfoncée, ce qui donne le même résultat que si la touche était maintenue appuyée en continu. Avec la répétition automatique activée, ceci "tape" des milliers de caractères à l'invite de login. Cherchez un écran rempli de caractères identiques (dans certains cas avec deux caractères différents ou plus).

17.6 Échec après le login

Si vous pouvez vous logger correctement mais ne pouvez utiliser le terminal il se peut que le démarrage du shell de login ait reconfiguré le terminal (avec des paramètres incorrects) à cause d'une commande que quelqu'un a mise dans l'un des fichiers qui sont lancés quand vous vous loggez, et qu'un shell soit lancé. Ces fichiers comprennent /etc/profile et /.bashrc. Cherchez une commande commençant par "stty" ou "setserial" et assurez-vous qu'elle est correcte. Même si elle est correcte dans un fichier d'initialisation, elle peut être repositionnée de manière incorrecte dans un autre fichier d'initialisation que vous ne soupçonnez pas. Des méthodes pour revenir sur le système afin de le réparer et d'utiliser un autre terminal ou console est d'utiliser une disquette de secours ou de taper : "linux single" à l'invite de LILO qui vous mettra en mode utilisateur unique sans lancer les fichiers de démarrage.

17.7 Impossible de se logger

Si vous obtenez une invite de login mais pas de réponse (ou peut-être une réponse embrouillée) à vos tentatives de login, une cause possible est que la communication se fait mal dans un sens du terminal vers l'ordinateur. Cela peut être dû à un connecteur/câble mauvais ou mal câblé. Si vous n'utilisez pas encore l'option "local" de getty, faites-le afin de désactiver les lignes de contrôle du modem. Voyez 14.1 (getty (dans /etc/inittab)). Vous pourriez aussi désactiver le contrôle de flux matériel (stty -crtscts) s'il était activé. Si cela fonctionne maintenant correctement, alors soit les lignes de contrôle de votre modem sont câblées de manière incorrecte, soit il y a une erreur dans votre configuration. Certains terminaux permettent le positionnement de valeurs différentes (comme la vitesse de transmission) pour envoyer et recevoir, de sorte que la réception soit bonne mais pas l'envoi.

Vous devriez aussi (sur la console) essayer "stty < /dev/ttyS1" (si vous utilisez ttyS1) pour vérifier qu'il est configuré correctement. Il sera souvent en mode brut (et c'est sûrement bon) avec -icanon et -echo, etc. Si le terminal est configuré de manière incorrecte en half-duplex (HDX), alors une partie des caractères que vous voyez quand vous tapez viennent du terminal lui-même. Si les caractères sont doublés, alors les échos de l'ordinateur sont bons et vous pouvez passer en full-duplex pour corriger ceci. Mais si vous êtes en half-duplex et que vous ne voyez que ce qui semble être des "échos" normaux, ils ne viennent donc pas de l'ordinateur comme ils devraient le faire.

Si vous obtenez un message qui ressemble à "login failed" (le login a échoué) alors, si vous n'avez fait aucune erreur en tapant ou dans votre mot de passe, il peut y avoir des restrictions sur les logins qui ne vous permettent pas de vous logger. Malheureusement, ce message peut ne pas vous dire pourquoi la tentative a échoué. Voyez 14.8 (restrictions sur les logins).

17.8 Invite de login embrouillée

Ceci peut-être à cause de l'utilisation d'un jeu de caractères incorrect, des erreurs de transmission dues à des vitesses de connexion trop élevées, des vitesses de connexion incompatibles, des parités incompatibles ou un nombre de bits par octet incorrect. Si c'est une variété de caractères étranges vous avez un jeu de caractères incorrect ou un bit de haut rang est positionné par erreur. Si les mots ont des fautes d'orthographe, essayez une vitesse de transmission plus basse. Pour les incompatibilités de vitesse de transmission, de parité ou de bits/caractère, vous voyez beaucoup d'erreurs "character error" identiques (erreur de caractère) qui représentent le fait qu'un vrai caractère ne peut être affiché correctement à cause d'une erreur dans la parité ou la vitesse de transmission.

Si vous utilisez agetty (souvent nommé simplement getty), le programme agetty détectera et positionnera la parité et/ou les bits/caractère si vous tapez quelque chose. Essayez-le avec un retour chariot pour voir si cela répare quelque chose.

17.9 Aucun signe d'une quelconque invite de login

Cela arrive quand rien ne se passe du tout sur le terminal, mais que le terminal semble fonctionner correctement. L'une des premières choses à faire est de s'assurer que toutes les connexions câblées sont fermes et reliées au bon connecteur à la fois sur l'ordinateur et sur le terminal. D'autres causes comprennent du matériel ou des câbles défectueux (les câbles doivent être des null-modem), getty ne tournant pas, une différence de vitesse de transmission, un terminal en mode local, etc. À ce point, deux possibilités d'approche sont (vous pouvez en suivre plus d'une à la fois) :

- 17.9.1 (diagnostiquer les problèmes à partir de la console)
- 17.9.2 (mesurer les tensions)

17.9.1 Diagnostiquer les problèmes à partir de la console

Sur la console (ou sur un autre terminal qui fonctionne), utilisez "top" ou "ps -al" pour voir si getty fonctionne sur le port. Ne le confondez pas avec d'autres programmes getty qui tournent sur d'autres ports ou sur les consoles virtuelles. Vous n'obtiendrez pas d'invite de login si getty ne tourne pas.

Un test possible est de tenter la copie d'un court fichier vers le terminal (une bonne idée serait d'essayer cela au début du processus d'installation avant de configurer getty). Utilisez la commande Linux de copie comme ceci : cp nom_fichier /dev/ttyS1. Si cela ne fonctionne pas, utilisez stty pour rendre l'interface aussi simple que possible en désactivant tout (comme le contrôle de flux matériel : -crtscts ; la parité, et les signaux de contrôle du modem : clocal). Assurez-vous que les vitesses de transmission et le nombre de bits par octet sont les mêmes. Si rien ne se passe, vérifiez que le port est vivant avec un voltmètre grâce à la section suivante.

17.9.2 Mesure des tensions

Si vous disposez d'un voltmètre à portée de main, vérifiez qu'il y a une tension négative (-4v à -15v) sur la broche 3 (réception de données) du côté du terminal sur le câble null modem. La borne positive du voltmètre devrait être reliée à une bonne terre (les connecteurs métalliques sur les extrémités des câbles ne sont souvent pas reliés à la terre). S'il n'y a pas de tension négative, vérifiez-la sur la broche de transmission (TxD) sur l'ordinateur (voyez 11.2.2 (DB9-DB25) pour le brochage). Si elle est présente là mais pas sur la broche de réception (RxD) du terminal, alors le câble est mauvais (connexion flottante, câble cassé ou le câble n'est pas null modem). S'il n'y a pas de tension du côté de l'ordinateur, le port série de l'ordinateur est mort. Testez-le avec un programme de diagnostics ou remplacez-le.

Si le port série est vivant, vous pouvez lui envoyer un fichier (avec les contrôles de modem désactivés) et voyez si quelque chose y arrive. Pour vérifier qu'un signal est transmis avec un voltmètre analogique, regardez l'aiguille à -12 V quand la ligne est inactive. Commencez ensuite à envoyer un fichier (ou lancez getty). Vous devriez voir l'aiguille revenir à zéro et bouger autour de 0 alors qu'elle mesure des moyennes de courte distance sur le flux de données. Vous pouvez aussi le voir sur l'échelle de courant alternatif à condition que votre voltmètre dispose d'une capacité pour bloquer les tensions continues quand vous êtes sur l'échelle alternative. S'il n'en a pas, alors le -12 V continu en inactif donnera une lecture alternative erronée. Sans voltmètre, vous pourriez relier un périphérique que vous savez en bon état (comme un autre terminal ou un modem externe) au port série et voir s'il fonctionne correctement.

17.10 Ralentissement: Pauses de quelques secondes entre l'affichage des caractères

Vous avez surement mal parametrer les interruptions: Voyez les sections du Serial-HOWTO commençant par "Ralentissement:".

17.11 Surveillance et diagnostics du port série

Quelques programmes Linux surveilleront les lignes de contrôle du modem et indiqueront si elles sont positives (1) ou négatives (0).

- statserial (dans la distribution Debian)
- serialmon (ne surveille pas RTS, CTS, DSR mais indique les autres fonctions)
- modemstat (ne fonctionne que sur les consoles PC Linux. Fonctionnera en concordance avec la ligne de commande)

Vous les avez peut-être déjà. Sinon, allez à *logiciels série* `http://metalab.unc.edu/pub/Linux/system/serial/` . En les utilisant, gardez à l'esprit que ce que vous voyez est l'état des lignes sur l'ordinateur hôte. La situation sur le terminal sera différente puisque certains fils sont souvent manquants des câbles alors que d'autres fils se croisent. En juin 1998, je ne connais aucun programme de diagnostic sous Linux pour le port série.

17.12 Mode local

En mode local, le terminal se déconnecte de l'ordinateur et se comporte comme une machine à écrire (sauf qu'il n'imprime pas sur papier mais sur l'écran). En revenant en ligne, le terminal se reconnecte à l'ordinateur vous permettant de reprendre les activités au point où vous vous étiez arrêté quand vous êtes passé en mode "local". Ceci est utile à la fois pour tester le terminal et à des fins éducatives. En mode local vous pouvez taper des séquences d'échappement (en commençant par la touche ESC) et observer ce qu'elles font. Si le terminal ne fonctionne pas correctement en mode local, il est quasiment certain qu'il ne fonctionnera pas mieux quand il sera relié à l'ordinateur. Si vous n'êtes pas vraiment sûr de ce que fait une séquence d'échappement, vous pouvez l'essayer en mode local. Vous pouvez aussi l'utiliser pour essayer un terminal qui est à vendre. Pour aller en mode local vous devez d'abord entrer en mode de configuration et ensuite sélectionner "local" dans un menu (ou presser une certaine touche). Voyez 13.3 (aller dans le mode de configuration).

17.13 Équipement de test électrique pour le port série

17.13.1 Gadgets d'évasion etc.

Alors qu'un multimètre (utilisé comme voltmètre) peut être tout ce dont vous avez besoin pour quelques terminaux, un équipement de test spécial simple a été fait pour tester les lignes des ports série. Certains s'appellent "évasion ..." (breakout, NdT) où évasion veut dire sortir des conducteurs d'un câble. Ces gadgets possèdent quelques connecteurs et s'insèrent dans le câble série. Certains possèdent des points de tests pour y relier un voltmètre. Certains possèdent des LEDs qui s'allument quand certaines lignes de contrôle sont activées (allumées). D'autres encore possèdent des cavaliers pour que vous puissiez relier n'importe quel fil à n'importe quel fil. Certains possèdent des interrupteurs.

Radio Shack vend (en 1998), un "expert de RS-232" ou "testeur de ligne RS-232" qui vérifie TD, RD, CD, RTS, CTS, DTR, et DSR. Une lumière verte veut dire "allumé" (+12 V) alors que rouge veut dire "éteint" (-12 V). Ils vendent aussi une "boîte de cavaliers série RS-232" qui permet de relier les broches de la manière dont vous le souhaitez.

17.13.2 Mesurer des tensions

N'importe quel voltmètre ou multimètre, même les moins chers qu'on vend pour environ 60 F, devraient fonctionner correctement. Essayer d'utiliser d'autres méthodes pour tester la tension est compliqué. N'utilisez pas de diode électroluminescente (LED) sauf si on lui adjoint une résistance en série pour réduire la tension sur la diode. On utilise une résistance de 470 ohms pour une diode de 20 mA (mais toutes les diodes ne font pas 20 mA). La diode ne s'allumera que pour une certaine polarité pour que vous puissiez tester les tensions positives ou négatives. Personne ne fait un tel gadget pour tester automatiquement les circuits ?? Les sondes logiques peuvent être endommagées si vous essayez de les utiliser puisque les tensions TTL pour lesquelles elles sont faites ne sont que de 5 volts. Tenter d'utiliser une ampoule incandescente de 12 V n'est pas une bonne idée. Cela ne vous montrera pas la polarité et à cause du courant de sortie limité de l'UART, l'ampoule ne s'allumera probablement pas.

Pour mesurer la tension sur un connecteur femelle, vous pouvez enfoncer un trombone déplié dans l'ouverture désirée. Le diamètre du trombone ne doit pas être plus grand que les broches afin de ne pas abîmer le contact. Mettez une pince croco (ou autre) sur le trombone pour vous connecter.

17.13.3 Goûter la tension

En dernier recours, si vous n'avez pas d'équipement de test et voulez risquer d'être choqué (ou même électrocuté) vous pouvez toujours goûter à la tension. Avant de toucher à l'une des broches de test avec votre langue, testez-les pour vous assurez qu'il n'y a pas de haute tension sur elles. Prenez les deux broches (à la fois) dans une main pour voir si ça vous fait de l'effet. Si ce test vous choque, vous n'aurez sûrement pas envie d'utiliser votre langue.

Pour tester du 12 V, léchez un doigt et tenez-y une broche de test. Mettez l'autre broche de test sur votre langue. Si la broche sur votre langue est positive, il y aura un goût reconnaissable. Vous pouvez d'abord essayer ceci avec des piles 4,5 V pour savoir quel goût ça aura.

18 Réparations et diagnostics

La réparation d'un terminal possède beaucoup en commun avec la réparation d'un moniteur et/ou d'un clavier. Parfois les diagnostics intégrés au terminal vous indiqueront à l'écran ce qui ne va pas. Sinon, par les symptômes, on peut souvent isoler le problème à l'une des choses suivantes : clavier défectueux, écran mort, panne de l'électronique numérique du terminal. Il est mieux d'avoir un manuel de service, mais même si vous n'en avez pas, beaucoup de terminaux peuvent encore être réparés.

18.1 Livres et sites Web sur la réparation

18.1.1 Livres

Bigelow, Stephen J. : Troubleshooting & Repairing Computer Monitors, 2ème édition, McGraw-Hill, 1997. Ne couvre pas l'électronique de génération des caractères ni le clavier.

18.1.2 Sites Web

La FAQ `http://www.repairfaq.org` du groupe de nouvelles sci.electronics.repair est longue et complète, bien qu'elle ne couvre pas les terminaux en soi. Voyez la section "Moniteurs d'ordinateurs et vidéos" ("Computer and Video Monitors", NdT). La plupart de ces informations peuvent s'appliquer aux terminaux ainsi que dans les sections "tester les capacités", "tester les transformateurs", etc. Peut-être que dans le futur, les "informations" de réparation de ce HOWTO consisteront principalement en des liens vers la FAQ ci-dessus (ou un document identique). Une autre source d'informations est l' *archive de réparations de Shuford* `http://www.cs.utk.edu/~shuford/terminal/repair_hints_news.txt` , archive de posts dans des groupes de nouvelles sur la réparation des terminaux.

18.2 Sécurité

Les écrans utilisent de très hautes tensions jusqu'à 30000 volts pour la couleur (un peu moins pour le noir et blanc). Faites attention de ne pas toucher cette tension si l'écran est allumé et le couvercle retiré. Cela ne vous tuera probablement pas même si vous le faites puisque la somme de courant qu'il peut fournir est limité. Mais il est possible que cela vous brûle gravement et vous choque, etc. Les hautes tensions peuvent passer à travers des couches d'air et traverser certains isolants, alors gardez vos mains à une distance sûre. Vous devriez remarquer le câble à haute tension bien isolé connecté à un bout du tube d'images. Même quand l'écran est éteint, il reste suffisamment de tension résiduelle sur la connexion du câble au tube d'images pour vous donner un certain choc. Pour décharger cette tension quand l'écran est débranché, utiliser un tournevis (poignée isolée) avec la lame en métal relié au câble de masse du tube d'images par un fil cavalier. N'utilisez pas la masse du chassis.

Les tensions plus basses (quelques centaines de volts) peut être encore plus dangereuses parce qu'elles ne sont pas limitées en courant. Elles sont même encore plus dangereuses si vos mains sont mouillées ou si vous portez un bracelet de montre en métal, un anneau ou autre. Dans certains cas rares, des gens en sont morts alors faites attention. Les tensions plus faibles de seulement quelques volts sur les circuits numériques sont relativement sûrs mais ne touchez à rien (sauf avec un outil bien isolé) sauf si vous savez ce que vous faites.

18.3 Apparence de l'affichage

Si l'affichage est trop faible, augmentez la luminosité et/ou le contraste en utilisant les molettes à l'extérieur de l'unité (si elles existent). Si la largeur, la hauteur ou le centrage sont incorrects, il y a souvent des molettes de contrôle pour ceux-ci. Sur certains terminaux anciens, on doit presser une touche fléchée (ou autre) en mode de configuration.

Vous aurez peut-être besoin d'enlever le couvercle pour faire des ajustements, surtout sur les modèles anciens. Vous pourriez arranger les choses pour qu'un grand miroir soit en face du terminal afin de voir l'affichage dans le miroir tout en faisant les ajustements. Ce qu'il faut tourner peut se trouver sur un circuit imprimé. Alors qu'un tournevis (peut-être avec une tête Phillips) peut être tout ce dont vous avez besoin, les bobines peuvent nécessiter certains outils spéciaux d'alignement de télévisions (clés en plastique, etc.). Le nom abrégé de l'ajustement devrait être imprimé sur le circuit imprimé. Par exemple, voici de tels noms :

- V-Size ajuste la hauteur verticale (taille)
- H-Size ajuste largeur horizontale (taille). Ce peut-être une bobine.
- V-Pos ajuste la position verticale
- H-Pos ajuste la position horizontale

- V-Lin ajuste la linéarité verticale (à utiliser si la largeur des lignes de balayage diffère en haut et en bas de l'écran)
- V-Hold ajuste le maintien vertical (à utiliser si l'écran défile de manière incontrôlable)
- Bright ajuste la luminosité (une molette extérieure peut aussi exister)
- Sub-Bright ajuste la luminosité du mode d'intensité atténuée (souvent le mode normal : plus faible que le mode gras ou fort).

Changer la linéarité peut modifier la taille et donc il peut être nécessaire de la réajuster. Un terminal qui a été stocké pendant quelque temps peut avoir un petit rectangle d'affichage sur l'écran entouré d'un grand bord noir. S'il est difficile à ajuster, attendez un peu avant de l'ajuster puisqu'il va en récupérer un peu avec l'utilisation (les bords noirs vont rétrécir).

18.4 Diagnostiquer

18.4.1 Le terminal a émis un bruit

Si le terminal a émis un bruit juste avant de tomber en panne (ou quand vous l'allumez juste après qu'il est tombé en panne), ce bruit est un indice de ce qui ne va pas. Si vous entendez un crépitement ou voyez/sentez de la fumée, éteignez immédiatement le terminal pour empêcher des dommages supplémentaires. Le problème est sûrement dans l'alimentation en haute tension de plusieurs milliers de volts. Enlevez le couvercle et si le point faible n'est pas évident, rallumez-le pendant une courte période de temps dans une pièce peu éclairée et regardez les arcs électriques. Le câble à haute tension (qui court entre le transformateur et le côté du tube d'images) peut avoir une isolation défectueuse qui provoque des arcs avec la terre. Réparez-le avec de l'isolant haute tension, ou du chatterton électrique spécial fait pour, disons, 10000 volts.

Le transformateur (haute tension) peut ne faire qu'un cliquetis ou un crépitement faible quand il tombe en panne. Vous pouvez ne pas l'entendre jusqu'à ce que vous éteigniez le terminal pendant un moment pour le reposer et l'allumiez ensuite à nouveau. Pour déterminer la provenance du bruit, vous pouvez utiliser un morceau de tube en caoutchouc (comme on en utilise dans les voitures) comme stéthoscope pour écouter. Mais pendant que vous écoutez le son, le terminal souffre de plus de dommages alors essayez de le trouver rapidement (mais pas rapide au point de risquer d'être électrocuté).

Un court-circuit dans l'alimentation peut faire sauter un fusible avec un bruit "pop". Le remplacement d'un fusible éclaté peut ne pas résoudre le problème car le même court-circuit peut faire éclater le fusible à nouveau. Recherchez les points noircis à cause d'une chaleur trop importante et testez ces composants. Les transistors de puissance court-circuités peuvent faire éclater le fusible. On peut les tester avec un vérificateur de transistors ou même avec un ohmmètre. Utilisez une petite échelle d'ohms sur un ohmmètre pour que la tension appliquée par l'ohmmètre soit faible. Ceci réduira les dommages possibles sur les composants sains causés par ce test de tension.

Si le terminal a été exposé à l'humidité, en étant stocké dans un endroit humide ou près d'une cuisine avec la vapeur de la cuisine, une solution peut être de sécher l'unité. Chauffer un transformateur "en panne" avec un sèche-cheveux pendant quelques minutes peut le ranimer.

18.4.2 Le terminal n'a émis aucun bruit

Un écran vide peut être causé par une personne qui a tourné le contrôle de luminosité au plus bas niveau ou par l'âge. La chose à faire alors est de vérifier les câbles pour voir si les connexions sont mal faites ou cassées. S'il n'y a pas de signe de courant, mettez un nouveau cordon d'alimentation après vous être assuré que la prise de courant murale délivre du courant.

Si vous soupçonnez le clavier, essayez-le sur un autre terminal du même type ou mettez un bon clavier. Manipulez les extrémités du câble du clavier et la prise. Les fils à l'intérieur du câble peuvent casser, surtout vers leurs extrémités. Si la cassure est vérifiée en bougeant le câble (et en alternant la panne et la bonne marche du terminal en même temps que le mouvement), il faut alors soit obtenir un nouveau câble, soit couper le câble et ressouder les cassures, etc.

L'une des premières choses à faire si le clavier fonctionne est de mettre le terminal en 17.12 (mode local). Si cela fonctionne en local, alors le problème vient sûrement de la connexion à l'ordinateur hôte (ou d'une interface incorrecte) ou dans les puces UART du terminal.

En inspectant avec attention les circuits, on peut souvent trouver la cause du problème. Regardez les changements de couleurs, les craquelures, etc. Un problème intermittent peut se révéler en appuyant sur les composantes avec un stylo à bille (pas la partie métallique, bien sûr). Une cassure de la partie conductrice d'un circuit imprimé peut parfois être révélée en tordant le circuit. De la soudure qui semble avoir fait une goutte ou un joint avec un peu de soudure peut avoir besoin d'être refaite. La soudure peut faire chauffer les transistors (et d'autres composants) et les endommager, utilisez donc un puits de chaleur si c'est faisable.

Si vous avez une marque de terminal connue, vous pouvez chercher des posts sur les groupes de nouvelles sur l'Internet pour trouver les types de problèmes les plus fréquents pour votre terminal et peut-être des informations sur la manière de les réparer.

Pour voir si l'électronique numérique fonctionne, essayez (en utilisant un bon clavier) de taper sur le mauvais terminal. Essayez de lire cela en tapant sur un bon terminal (ou sur la console) en utilisant la commande de copie ou avec un programme de communication avec les terminaux comme Minicom. Vous aurez peut-être besoin d'appuyer sur la touche retour chariot afin d'envoyer une ligne. On peut demander l'identité, etc. du mauvais terminal à partir d'un autre terminal. Cela montrera si la communication dans les deux sens fonctionne.

18.5 Messages d'erreur à l'écran

Vous avez de la chance si vous voyez un message d'erreur à l'écran. Cela arrive en général quand vous allumez pour la première fois le terminal.

18.5.1 Erreur de clavier

Ceci veut dire en général que le clavier n'est pas branché, ou que la connexion est branlante. Pour des problèmes plus sérieux, voyez 18.7 (claviers).

18.5.2 Erreur de somme de contrôle en NVR

La NVR est la mémoire non volatile (Non-Volatile RAM, NdT). Ceci veut dire que la NVR, où sont stockées les informations de configuration, est corrompue. Le terminal fonctionnera sûrement encore mais la configuration qui avait été sauvegardée la dernière fois que quelqu'un a configuré le terminal a sûrement été perdue. Essayez de refaire la configuration et de la sauver. Cela a des chances de fonctionner. Sur certains terminaux très vieux (début des années 1980) il y avait un CMOS alimenté sur pile pour sauver la configuration donc dans ce cas le problème peut venir d'une pile morte. Parfois la puce EEPROM (pas besoin de pile) devient mauvaise après trop de sauvegardes. On aura du mal à en trouver. Si vous ne pouvez pas la réparer vous êtes soit bloqué avec la configuration par défaut ou vous pouvez envoyer des séquences d'échappement au terminal quand vous le démarrez ou quand vous essayez de le configurer.

18.6 Capacités

Les capacités électrolytiques possèdent une coquille de métal et peuvent faiblir ou tomber en panne s'ils restent inutilisés pendant des années. Parfois le fait de laisser le terminal allumé pendant un certain temps peut aider à les restaurer en partie. Si vous le pouvez, faites faire de l'exercice aux terminaux que vous avez en stock en les allumant pendant quelques instants chaque année ou tous les deux ans.

18.7 Claviers

18.7.1 Interchangeabilité

Les claviers pour terminaux ne sont pas les mêmes que les claviers pour PC. La différence ne réside pas seulement dans la disposition des touches mais aussi dans les codes générés quand on presse une touche. De plus, les claviers pour diverses marques et modèles de terminaux ne sont pas toujours interchangeables. On obtient parfois un clavier "incompatible" qui fonctionne en partie sur un terminal. Toutes les touches ASCII fonctionneront correctement, mais les touches spéciales pour la configuration et la pause ne fonctionneront pas correctement.

18.7.2 Comment ils fonctionnent

La plupart des claviers font simplement un contact entre deux conducteurs quand vous appuyez sur une touche. L'électronique à l'intérieur d'une puce dans le clavier convertit l'établissement de ce contact en un code envoyé à travers le câble externe du clavier. Au lieu d'avoir un fil (ou conducteur) séparé allant de chaque touche à la puce, le principe suivant est utilisé. Numérotez les conducteurs disons de 1 à 10 et de A à J. Par exemple : le conducteur 3 conduit à plusieurs touches et le conducteur B conduit à plusieurs touches, mais seule une touche a les deux conducteurs qui la rejoignent. Quand cette touche est pressée, un court-circuit est établi entre 3 et B. La puce ressent ce court-circuit et sait quelle touche a été pressée. Un tel principe réduit le nombre de conducteurs nécessaire (et réduit le nombre de broches nécessaires sur la puce). C'est un principe similaire à ce qu'on appelle un commutateur croisé (crossbar).

18.7.3 L'appui sur une touche affiche deux caractères différents

Si, à cause d'un défaut, les conducteurs 3 et 4 sont court-circuités alors l'appui sur la touche 3-B court-circuitera aussi 4 et B et la puce croira que les touches 3-B et 4-B ont été pressées à la fois. Ceci fera sûrement afficher deux caractères différents alors que tout ce que vous souhaitiez était un caractère.

18.7.4 Claviers modernes contre anciens

Alors que le clavier moderne et le type ancien se ressemblent beaucoup, la mécanique d'opération est différente. Les vieux possèdent des contacts de touches individuels sous le capuchon de chaque touche, chaque contact étant inclus dans une enveloppe en plastique dur. Les claviers modernes utilisent de grandes feuilles (membranes) en plastique souple de la taille du clavier. Une feuille de plastique avec des trous est prise en sandwich entre deux autres feuilles de plastique contenant des circuits imprimés (comprenant des points de contact). Quand vous appuyez sur une touche, les deux feuilles "imprimées" sont pressées l'une contre l'autre à un certain point, ce qui ferme les contacts imprimés sur les feuilles à ce point.

18.7.5 Le clavier ne fonctionne pas du tout

Si aucune touche ne fonctionne, essayez un autre clavier (si vous en avez un) pour vérifier que le clavier est effectivement le problème. La cause la plus probable est un fil cassé à l'intérieur du cordon (câble) le reliant au terminal. La position la plus probable de la cassure est à l'une des extrémités du cordon. Essayez de manipuler les extrémités du cordon tout en tapant sur une touche pour voir si ça fonctionne de manière intermittente. Si vous trouvez un point endommagé, vous pouvez couper attentivement le cordon avec un couteau à l'endroit du point endommagé et épisser le conducteur cassé. Parfois une simple goutte de soudure fera l'affaire. Scellez le cordon avec du chatterton ou de la colle.

18.7.6 L'appui sur b affiche bb, etc. (affichage en double)

Si tous les caractères apparaissent en double il n'y a sûrement pas de problèmes avec le clavier. En revanche, votre terminal a sûrement été configuré de manière incorrecte en semi-duplex (HDX ou echo local = oui) et chaque caractère que vous tapez est renvoyé à la fois depuis l'électronique à l'intérieur de votre terminal et depuis votre ordinateur hôte. Si les deux caractères ne sont pas les mêmes, il peut y avoir un court-circuit à l'intérieur de votre clavier. Voyez 18.7.3 (un appui affiche deux caractères différents).

18.7.7 Le clavier tape tout seul

Si une touche est court-circuitée il est probable qu'elle tapera un grand nombre de fois le même caractère si la répétition automatique est activée. Si plus d'une touche est court-circuitée, alors la répétition de séquences de quelques caractères sera tapée. Cela peut amener getty à se relancer trop rapidement si cela arrive à l'invite de login. Voyez 17.5.3 (touche court-circuitée). La solution est de nettoyer les contacts grâce à 18.7.9 (nettoyage des contacts du clavier).

18.7.8 Liquide versé sur le clavier

Si de l'eau ou du liquide aqueux a été versé sur le clavier (ou s'il a été exposé à la pluie, une rosée forte ou à l'humidité), certaines touches ne fonctionneront pas correctement. L'humidité peut faire un court-circuit sur une touche (comme si on appuyait dessus tout le temps) et vous pourrez voir l'écran se remplir avec cette lettre si la répétition automatique est activée. S'il est devenu humide et ensuite séché en partie (ou en totalité), certaines touches pourront ne pas fonctionner à cause de dépôts sur la surface des contacts. Sur les types de claviers modernes, on peut facilement séparer les feuilles de plastique à l'intérieur et les sécher/nettoyer. Pour les plus anciens, on peut les laisser sécher au soleil ou au four (basse température). Quand c'est sec il faudra peut-être nettoyer les contacts comme expliqué ci-dessous.

18.7.9 Nettoyage des contacts du clavier

Claviers avec membrane Sur certains claviers récents, les feuilles de plastique (membranes) sont faciles à enlever pour les inspecter et les nettoyer si nécessaire. Vous n'avez besoin d'enlever que quelques vis pour séparer le clavier en deux et obtenir les feuilles. Sur certains vieux clavier IBM les feuilles ne peuvent pas être enlevées sans casser beaucoup de taquets en plastique qu'il faudra réparer à la colle afin de les remettre (probablement pas la peine de les réparer). Un tel clavier peut parfois fonctionner en tordant, tournant et/ou pesant sur l'assemblage contenant les feuilles de plastique.

Claviers avec contacts individuels Ce qui suit concerne les vieux claviers qui possèdent des contacts séparés en plastique dur pour chaque touche. Avant de faire tout le travail de nettoyage des contacts électriques essayez d'abord de tourner le clavier tête en bas et bougez les mauvaises touches. Ceci peut aider à déloger les saletés, surtout si vous pressez la touche fortement et rapidement pour faire une vibration. (NdT : bien secouer le clavier régulièrement fait effectivement tomber toutes les saletés, miettes de pain, etc. et fait du bien au clavier !) Il est souvent utile d'enfoncer la touche et de l'agiter de bord à bord.

Souvent on peut enlever les capuchons de touches en les découvrant vers le haut en utilisant un petit tournevis comme levier tout en empêchant une inclinaison excessive avec un doigt. Il existe un outil spécial appelé extracteur de touches mais vous pouvez vous en passer. (Attention : les capuchons de touches sur les claviers modernes ne se découvrent pas.) Le capuchon de touche peut basculer un peu et branler alors qu'il se détache. Il peut même s'envoler et atterrir par terre. Vous avez alors deux choix sur le nettoyage des contacts : utiliser un vaporisateur de nettoyant de contact directement au-dessus du contact de la touche, ou séparer le contact de touche et le nettoyer. Un tout autre choix est de remplacer le contact de touche par un nouveau ou un d'occasion.

La vaporisation directe d'un nettoyant de contacts ou autre (obtenu dans un magasin d'électronique) au-dessus du contact de la touche est la méthode la plus rapide mais peut ne pas fonctionner et peut aussi endommager le plastique. Avant de vaporiser, nettoyez la surface près des supports de contacts. Avec le clavier branché (ou en connectant un ohmmètre sur les contacts de touches) utilisez le tube livré avec le vaporisateur pour injecter du nettoyant à l'intérieur du contact de touche. Ne laissez pas le liquide de nettoyage s'en aller dans les touches voisines où il pourrait amasser de la poussière et s'infiltrer (avec la poussière) dans les contacts de touches adjacents. Si vous faites cette erreur, vous pourriez réparer une touche et abimer les touches voisines. Si cela arrive, faites immédiatement bouger les touches adjacentes affectées jusqu'à ce qu'elles fonctionnent correctement.

Vous pouvez basculez le clavier pour que le nettoyant coule à l'intérieur des contacts. Pour le terminal CIT101e avec un clavier Alps, ceci implique de basculer la rangée des chiffres vers le plafond. Faites bouger le contact de touche vers le haut et vers le bas avec un stylo ou le manche d'un petit tournevis pour éviter d'avoir du liquide nettoyant toxique sur votre peau (ou portez des gants). Finalement retournez le clavier tête en bas tout en bougeant la touche pour enlever le nettoyant qui reste. Plus vous injecterez de nettoyant plus vous serez sûr de réparer la touche mais vous aurez aussi plus de chances d'endommager le plastique ou de contaminer les touches adjacentes, utilisez donc ce que vous jugez nécessaire pour faire le travail. Une fois que la touche fonctionne correctement, bougez-la de haut en bas encore un peu et testez-la une demi-minute plus tard, etc. pour vous assurer qu'elle fonctionne encore correctement.

Parfois une touche fonctionne très bien quand les contacts à l'intérieur sont saturés de liquide de nettoyant de contacts, mais quand le liquide sèche quelques minutes plus tard, le dépôt résultant sur les contacts empêche un contact correct et la touche fonctionne avec des ratés (si elle fonctionne). Faire bouger la touche alors que le liquide sèche à l'intérieur peut aider les choses. Certaines touches possèdent des contacts presque scellés à l'intérieur et donc peu de nettoyant pour contact atteint les contacts. Le nettoyant qui arrive effectivement sur les contacts peut apporter la contamination (le nettoyage autour du haut des touches avant la vaporisation peut aider à minimiser cet effet).

Si vous devez désassembler le contact de touche, inspectez-le d'abord pour voir comment il est installé et se sépare. Parfois on peut enlever le capuchon du contact sans enlever le contact du clavier. Pour ce faire, découvrez (ou tirez) le haut du contact de touche après avoir retiré les taquets en plastique fin qui le retiennent. Ne tirez pas trop fort ou vous pourriez casser le plastique fin. Si vous ne pouvez faire cela, vous devrez peut-être dessouder le contact et l'enlever afin de le séparer (ou de le remplacer). Une fois que le contact a été séparé, vous pourrez ne pas encore voir les contacts si les surfaces des contacts sont prises en sandwich (qui se touchent presque). Vous pouvez mettre du nettoyant pour contact sur les contacts en soulevant légèrement les surfaces conductrices et en injectant du nettoyant entre elles. Il peut y avoir une sorte d'attache maintenant les surfaces de contact ensemble qui doit être enlevée avant de soulever ces surfaces. Avec du nettoyant sur les contacts, faites-les bouger. Faire basculer le clavier ou le retourner peut aider. Prenez garde de ne pas perdre de petites parties car elles peuvent s'envoler en l'air quand vous enlevez un contact de touche.

19 Annexe A : généralités

19.1 Liste des commandes Linux pour les terminaux

19.1.1 Envoyer une commande à un terminal

- 16.11.1 (setterm) : options longues
- 16.11.2 (tput) : options courtes
- tset : ne fait que l'initialisation
- clear : efface l'écran
- reset : envoie une chaîne de réinitialisation

19.1.2 Configuration du pilote de périphériques pour les terminaux

- 14.3 (Setserial) :
- 14.4 (Stty)

19.1.3 Terminfo

- 15.2.3 (Compilateur Terminfo (tic)) : compilateur et traducteur pour terminfo
- toe : montre la liste des terminaux pour lesquels vous avez des fichiers terminfo
- 15.2.4 (infocmp) : compare ou affiche des entrées terminfo

19.1.4 Autres

- gitkeys : montre quels octets chaque touche envoie à l'hôte.
- tty : montre sur quel port tty vous êtes connecté.
- set (ou tset -q) : montre la valeur de TERM, le nom de l'entrée terminfo
- 15.6 (tset) : positionne TERM de manière interactive et fait l'initialisation

19.2 Internet et les livres

19.2.1 Information sur le terminal sur l'Internet

- *Site Web de Shuford* `http://www.cs.utk.edu/~shuford/terminal_index.html` à l'université du Tennessee possède beaucoup d'informations utiles sur les terminaux texte ;
- *Boundless* `http://www.boundless.com/textterm/` a racheté la partie terminaux VT et Dorio chez DEC. Pour obtenir des spécifications, choisissez les liens ADDS, VT ou DORIO. Choisissez ensuite un lien "data sheet". Ensuite, sur la feuille de données, sélectionnez le lien "Go to Specs".
- *Wyse* `http://www.wyse.com/service/support/kbase/wyseterm.asp` est un grand fabricant de terminaux. Pour les nouveaux modèles, voyez *terminaux Wyse* `http://www.wyse.com/terminal/`. Voyez aussi *vieilles spécifications des terminaux Wyse* `http://www.wyse.com/service/faq/wysetter.htm`

- *Séquences d'échappement ; Amérique du Nord* `http://www.pericom-usa.com/twdocs/doc/twproae.htm` ou *séquences d'échappement ; Europe* `http://www.pericom.co.uk/teemworld/doc/twproae.htm` est une liste de séquences d'échappement (et codes de contrôles) pour certaines émulations de terminal (qui comprend les VT 100, 300, 420 et Wyse) ;
- comp.terminals est le groupe de nouvelles pour les terminaux.

19.2.2 Livres liés aux terminaux

- port série EIA-232, voir 21.8 (Livres sur EIA-232 (RS-232)).
- réparations, voir 18.1 (livres et sites Web sur la réparation).
- base de données Terminfo, voir 15.7 (documents Termcap)

19.2.3 Livres consacrés entièrement aux terminaux

Autant que je sache, il n'existe pas de livre satisfaisant sur les terminaux texte (sauf si vous vous intéressez aux terminaux antiques des années 70).

- Handbook of Interactive Computer Terminals par Duane E. Sharp ; Reston Publishing Co. 1977. (quasiment obsolète)
- Communicating with Display Terminals par Roger K. deBry ; McGraw-Hill 1985. (principalement sur les terminaux synchrones IBM)

Le "HANDBOOK..." présente les spécifications brèves de plus de cent modèles différents de vieux terminaux fabriqués au début des années 1970 par plus de 60 sociétés différentes. Il explique aussi comment ils fonctionnent physiquement mais montre de manière incorrecte un diagramme pour un écran qui utilise une déviation électrostatique du faisceau d'électrons (même dans les années 1970). Ce livre explique un certain nombre de concepts techniques avancés comme le "balayage au hasard" et le "principe de pénétration de la couleur".

Le livre "COMMUNICATING..." au contraire du "Handbook..." ignore les détails physiques et électroniques des terminaux. Il possède un chapitre entier sur l'explication des nombres binaires (qui n'est pas nécessaire dans un livre sur les terminaux puisque cette information est largement disponible par ailleurs). Il semble couvrir principalement les vieux terminaux IBM (surtout les 3270) dans les modes de fonctionnement en bloc et synchrone. Il est de peu d'utilité pour les terminaux ANSI utilisés couramment de nos jours sur les systèmes de type Unix. Bien qu'il en parle un peu, il ne montre les différents systèmes de câblage utilisés pour les relier aux ports série.

19.2.4 Livres possédant des chapitres sur les terminaux

Ces chapitres ne couvrent presque rien sur les terminaux eux-mêmes et leurs capacités. Par contre, ces chapitres couvrent plutôt la manière de configurer l'ordinateur (et le pilote de terminal) pour qu'il fonctionne avec les terminaux. À cause des différences entre les systèmes Unix, la plupart des informations ne s'appliquent pas à Linux.

- Unix Power Tools by Jerry Peck et. al. O'Reilly 1998. Ch. 5 : configuration de votre terminal, Ch. 41 : paramètres du terminal et de la ligne série, Ch. 42 : problèmes avec les terminaux
- Advanced Programming in the Unix Environment par W. Richard Stevens Addison-Wesley, 1993. Ch. 11 : entrées/sorties avec le terminal, Ch. 19 : pseudo-terminaux

- Essential System Administration par Aleen Frisch, 2ème édition. O'Reilly, 1998. Ch. 11 : terminaux et modems.

Le livre "UNIX POWER TOOLS" possède trois chapitres courts sur les terminaux texte. Il couvre moins de choses que ce HOWTO mais donne plus d'exemples pour vous aider.

Le livre "ADVANCED PROGRAMMING...", dans le chapitre 11, ne couvre que le pilote de périphériques du système d'exploitation pour s'occuper des terminaux. Il explique les paramètres qu'on donne à la commande stty pour configurer le terminal.

Le chapitre du livre "ESSENTIAL SYSTEM..." en dit plus sur les terminaux que sur les modems. Il semble bien écrit.

19.3 Systèmes non Linux

La configuration de l'ordinateur hôte pour les terminaux sur des systèmes d'exploitation différents de Linux est en général largement différente que sous Linux. Voici quelques liens vers des manuels en ligne pour les systèmes de type Unix :

- *Ajouter des terminaux série* `http://www2.sco.com:1996/HANDBOOK/serial_terminal_adding.html` pour SCO OpenServer dans le manuel de SCO OpenServer.
- *Configuration des terminaux et modems* `http://www.software.hp.com/OS_transition/DOCS/PERIPH/TERMS3.HTM` pour HP-UX de Hewlett-Packard.

20 Annexe B : terminologie des commandes de séquences d'échappement

On les appelle parfois "séquences de contrôle". Cette section du Text-Terminal HOWTO est incomplète (et pourra ne jamais être complète car il y a un grand nombre de séquences de contrôle). Cette section sert de référence et appartient peut-être vraiment à ce qu'on pourrait appeler "Text-Terminal-Programming-HOWTO" (HOWTO sur la programmation d'un terminal texte).

Un exemple de séquence d'échappement ANSI normale est ESC[5B qui déplace le curseur vers le bas de cinq lignes. ESC est le caractère d'échappement. Le paramètre 5 est inclus dans la séquence. Si c'était 7 le curseur bougerait vers le bas de sept lignes, etc. Il est facile de comprendre l'explication suivante pour la séquence : "déplacer le curseur vers le bas de x lignes : ESC[xB". Mais un jargon de commande tel que : "requête d'attribut pour périphérique tertiaire" est moins compréhensible. Cette section essaiera d'expliquer une partie du jargon utilisé dans les commandes de séquences d'échappement. Une liste complète (comprenant les codes de séquences d'échappement pour la norme ANSI) est un projet "qu'on voudrait bien faire". Puisque beaucoup de séquences d'échappement font la même chose que ce qui est fait en configurant le terminal avec 13.6 (options de configuration), de telles options en séquences d'échappement ne seront pas répétées ici.

20.1 Liste de séquences d'échappement

Pour avoir une liste de nombreuses séquences d'échappement (mais pas toutes) pour divers terminaux, voyez *séquences d'échappement ; Amérique du Nord* `http://www.pericom-usa.com/twdocs/doc/twproae.htm` ou *séquences d'échappement ; Europe* `http://www.pericom.co.uk/teemworld/doc/twproae.htm` . On utilise celles-ci pour émuler un terminal et elles ne sont pas toujours les mêmes que sur le vrai terminal correspondant. Une liste pour les VT (non maintenue) se trouve à *FAQ Émulateurs* `http:/www.cs.ruu.nl/wais/html/na-dir/emulators-faq/part3.html` . Cherchez "VT".

20.2 Codes de contrôle 8 bits

Table des codes de contrôle 8 bits DEC (en hexadécimal). Fonctionne sur les VT2xx ou plus récents. CSI est le code le plus courant.

```
ACRONYME        NOM_COMPLET                         HEXA    REMPLACE
IND     Index (une ligne vers le bas)               84      ESC D
NEL     Ligne Suivante                              85      ESC E
RI      Index Inverse (une ligne vers le haut)      8D      ESC M
SS2     Décalage Simple 2                           8E      ESC N
SS3     Décalage Simple 3                           8F      ESC O
DCS     Chaîne de Contrôle Périphérique             90      ESC P
CSI     Introduction Séquence de Contrôle           9B      ESC [
ST      Terminaison de Chaîne                       9C      ESC \
```

20.3 Échappement pour l'imprimante

- Auto Print on/off (impression automatique oui/non) : Activée (on), les données venant de l'hôte sont aussi envoyées sur le port imprimante du terminal (et sont aussi affichées sur l'écran du terminal).
- Print Controller on/off (contrôleur d'impression oui/non) : Activée (on), les données venant de l'hôte ne sont envoyées qu'à l'imprimante (rien ne s'affiche sur l'écran du terminal).

20.4 Rapports

Ces séquences sont en général des requêtes envoyées de l'hôte pour demander un rapport du terminal. Le terminal répond en envoyant un rapport (en fait une autre séquence d'échappement) à l'hôte qui y a intégré certaines valeurs indiquant à l'hôte l'état en cours du terminal. Dans certains cas un rapport peut être envoyé à l'hôte même s'il n'a pas été demandé. Ceci arrive parfois quand on quitte la configuration. Par défaut aucun rapport non sollicité ne devrait être envoyé.

- Request for Status (Report Operating Status) (Demande d'état, rapporter l'état d'opération) : la signification des réponses du VT100 est soit "je vais bien", soit "je ne vais pas bien"
- Request for Device Attributes (demande des attributs du périphérique) : le "périphérique" est en général l'imprimante. Y a-t-il une imprimante ? Est-elle prête ?
- Request for Tertiary Device Attributes (pour les VT) (demande des attributs des périphériques tertiaires) : la réponse est le rapport qui a été entré pendant la configuration. Le périphérique tertiaire est le troisième périphérique (l'imprimante ou le périphérique sur le port auxiliaire ??). Le premier périphérique peut être l'ordinateur hôte et le deuxième périphérique le terminal.
- Request for Terminal Parameters (demande des paramètres du terminal) : quelle est la parité, la vitesse de transmission, la largeur d'octets, etc. Cette demande n'a pas l'air d'avoir beaucoup de sens, puisque si l'hôte ne connaissait pas déjà ces données, il ne pourrait pas communiquer avec le terminal ou envoyer une réponse.

20.5 Mouvements du curseur

Le curseur se trouve à l'endroit où le prochain caractère reçu de l'hôte sera affiché. La plupart des mouvements de curseur sont compréhensibles. "index cursor" (indexer le curseur) veut dire déplacer le curseur vers le bas d'une ligne. Les mouvements du curseur peuvent être relatifs à la position en cours comme "déplacer de 4 espaces vers la gauche" ou absolus comme "déplacer à la rangée 3, colonne 39". Le mouvement absolu s'appelle "positionnement direct du curseur" ou "adressage direct du curseur".

La position d'origine est rangée 1, colonne 1 (l'origine de l'index est 1). Mais l'emplacement de cette position d'origine à l'écran n'est pas clair. Si "mode d'origine du curseur", équivalent à "mode d'origine relatif", est choisi, l'origine se trouve en haut de la partie défilante (pas forcément le haut de l'écran). Si le "mode d'origine absolu" est choisi (même chose que désactiver l'un des deux modes de la phrase précédente) alors l'origine se situe dans le coin en haut à gauche de l'écran. Sur certains terminaux anciens si le "mode d'origine du curseur" est activé, cela veut dire que c'est relatif.

20.6 Pages (définition)

Voyez 8.4 (pages) pour avoir une explication sur les pages. Il y a un certain nombre de séquences d'échappement pour s'occuper des pages. Le texte peut être copié d'une page à une autre et on peut déplacer le curseur de page en page. Le passage d'une page à l'autre peut ou peut ne pas être automatique : quand l'écran est plein (page 1), alors les données supplémentaires venant de l'hôte vont sur la page 2. Le curseur peut n'être que sur une page à la fois et les caractères envoyés au terminal vont là. Si cette page n'est pas affichée, le nouveau texte sera reçu par le terminal et ira en mémoire d'affichage, mais vous ne le verrez pas (jusqu'à ce qu'on passe à cette page sur le terminal).

21 Annexe C : communications série sur EIA-232 (RS-232)

21.1 Introduction aux communications série

(Une grande partie de cette section se trouve maintenant dans le Serial-HOWTO.) Les terminaux texte sur les systèmes de type Unix (et sur les PC) sont connectés en général sur un port série asynchrone 232 d'un ordinateur. C'est en général un port RS-232-C, EIA-232-D ou EIA-232-E. Ces trois ports sont à peu près identiques. Le préfixe originel RS est devenu EIA (Electronics Industries Association) et plus tard EIA/TIA après que EIA se soit alliée avec TIA (Telecommunications Industries Association). La spécification EIA-232 décrit aussi les communications synchrones mais le matériel qui supporte les communications synchrones manque quasiment toujours sur les PC. La désignation RS est obsolète mais est toujours utilisée. On utilisera EIA dans cet article.

Le port série représente plus qu'un simple connecteur physique au dos d'un ordinateur ou d'un terminal. Il comprend l'électronique associée qui doit produire des signaux conformes à la spécification EIA-232. Le connecteur standard possède 25 broches, dont la plupart sont inutilisées. Un connecteur différent ne possède que neuf broches. Une broche est utilisée pour envoyer des octets de données et une autre pour en recevoir. Une autre broche est la masse commune du signal. Les autres broches "utiles" sont principalement utilisées à des fins de signalisation avec une tension négative régulière voulant dire "éteint" et une tension positive régulière voulant dire "allumé".

La puce UART (émetteur-récepteur asynchrone universel) fait la plus grande partie du travail. Aujourd'hui, les possibilités de cette puce sont en général incluses dans une autre puce.

21.2 Tensions

21.2.1 Tension pour un bit

Sur le port série EIA-232, les tensions sont bi-polaires (positives ou négatives par rapport à la masse) et devraient être de l'ordre de 12 volts en amplitude (certaines font 5 ou 10 volts). Sur les broches d'émission et de réception +12 volts représente le bit 0 (parfois appelé "espace") et -12 volts est le bit 1 (parfois appelé "marque"). On appelle cela la logique inversée puisque normalement le bit 0 est à la fois faux et négatif alors que le 1 est normalement vrai et positif. Bien que les broches de transmission et réception soient en logique inversée, d'autres broches (les lignes de contrôle du modem) sont en logique normale avec une tension positive étant vraie et une tension négative étant fausse. La tension zéro n'a aucune signification (sauf qu'elle veut dire en général que l'unité est éteinte).

Une étendue de tensions est permise. Les spécifications disent que l'amplitude d'un signal transmis devrait être entre 5 et 15 volts mais ne doit jamais dépasser 25 volts. Toute tension reçue en dessous de 3 volts est indéfinie (mais certains terminaux considèreront qu'une tension plus basse est valide). On voit parfois des affirmations erronnées selon lesquelles la tension est communément 5 volts (ou même 3 volts) mais c'est en général 11-12 volts. Si vous utilisez un port EIA-422 sur un ordinateur Macintosh comme un EIA-232 (cela demande un câble spécial) ou un EIA-423 alors la tension sera vraiment 5 volts. La discussion ici suppose que c'est 12 volts. Il y a beaucoup de confusion à propos des tensions sur Internet.

Notez que la logique d'ordinateur normale n'est que de quelques volts (à une époque, la norme était 5 volts), et que si vous essayez d'utiliser un équipement de test fait pour tester une logique d'ordinateur en 3-5 volts (TTL) sur les 12 volts d'un port série, cela peut endommager l'équipement de test.

21.2.2 Séquence de tension pour un octet

La broche de transmission (TxD) est maintenue à -12 V (marque) comme inactive quand rien n'est envoyé. Pour commencer un octet elle passe à +12 V (espace) pour le bit de départ et reste à +12 V pendant la durée (période) du bit de départ. Après vient le bit de bas niveau de l'octet de données. Si c'est un bit 0 rien ne change et la ligne reste à +12 V pendant une autre période de bit. Après vient le bit suivant, etc. Finalement, un bit de parité peut être envoyé et ensuite un bit de stop de -12 V (marque). La ligne reste à -12 V (inactive) jusqu'au prochain bit de départ. Notez qu'il n'y a pas de retour à 0 volts et il n'y a donc pas de moyen simple (sauf avec un signal de synchronisation) pour dire où finit un bit et où commence le bit suivant dans le cas où deux bits consécutifs ont la même polarité (tous les deux zéro ou tous les deux un).

Un deuxième bit de stop serait aussi à -12 V, identique au premier bit de stop. Puisqu'il n'y a pas de signal pour marquer la frontière entre ces deux bits, le seul effet du deuxième bit de stop est que la ligne doit rester inactive à -12 V deux fois plus longtemps. Le récepteur n'a aucun moyen de faire la différence entre un deuxième bit de stop et un temps d'inactivité plus long entre les octets. Ainsi les communications fonctionnent bien si une extrémité utilise un bit de stop et l'autre extrémité utilise deux bits de stop, mais n'utiliser qu'un bit de stop est visiblement plus rapide. Dans de rares cas, un bit de stop et demi est utilisé. Ceci veut dire que la ligne est gardée à -12 V pendant une période de temps et demie (comme un bit de stop 50 % plus long que la normale).

21.3 La parité expliquée

Les caractères sont normalement transmis sur 7 ou 8 bits (de données). Une parité supplémentaire peut (ou peut ne pas) y être ajoutée, ce qui donne un octet de longueur 7, 8 ou 9 bits. Certains émulateurs de terminaux et terminaux anciens n'autorisent pas 9 bits. Certains interdisent 9 bits si on utilise deux bits de stop (puisque cela ferait beaucoup trop de bits : 12 bits au total).

On peut mettre une parité impaire, paire, ou pas de parité (les parités marque et espace peuvent être des options sur certains terminaux). Avec une parité impaire, le bit de parité est sélectionné de telle sorte que le nombre de bit 1 dans un octet, en comprenant le bit de parité, soit impair. Si un tel octet se détériore par l'inversion d'un bit, le résultat est un octet illégal de parité paire. Cette erreur sera détectée et si c'est un octet arrivant au terminal, un symbole caractère d'erreur apparaîtra à l'écran. La parité paire fonctionne de manière similaire avec tous les octets légaux (comprenant le bit de parité) ayant un nombre de bit 1 pair. Pendant la configuration, le nombre de bits par caractère signifie en général le nombre de bits de données par octet (7 pour de l'ASCII pur et 8 pour les divers codes de caractères ISO).

Une "marque" est un bit 1 (ou un 1 logique) et un "espace" est un bit 0 (ou un 0 logique). Pour la parité marque, le bit de parité est toujours un bit 1. Pour la parité espace c'est toujours un bit 0. La parité marque ou espace ne fait que gâcher de la bande passante et devrait être évitée autant que possible. "Pas de parité" veut dire qu'aucun bit de parité n'est ajouté. Pour les terminaux qui n'autorisent pas les octets de 9 bits, il faut sélectionner "pas de parité" pour utiliser des codes de caractères sur 8 bits puisqu'il n'y a pas de place pour le bit de parité.

21.4 Formation d'un octet (encadrement)

Dans la transmission en série des octets par les ports EIA-232, le bit de bas niveau est toujours envoyé en premier. Les ports série sur les PC utilisent des communications asynchrones quand il y a un bit de départ et un bit de stop pour marquer le début et la fin d'un octet. On appelle cela l'encadrement et l'octet encadré s'appelle parfois un cadre. Au final, 9, 10 ou 11 bits sont envoyés par octet, 10 étant le nombre le plus courant. 8-N-1 veut dire 8 bits de données, pas de parité, 1 bit de stop. Ceci fait en tout 10 bits si on compte le bit de départ. Un bit de stop est utilisé quasiment partout. À 110 bits/seconde (et parfois à 300 bits/seconde) deux bits de stop étaient autrefois utilisés mais maintenant le deuxième bit de stop n'est utilisé que dans des situations très inhabituelles (ou par erreur puisqu'il semble encore fonctionner correctement de cette manière).

21.5 Limitations de EIA-232

21.5.1 Basses vitesses et courtes distances

Le port série EIA-232 traditionnel est à basse vitesse de manière inhérente, et est sérieusement limité en taille (distance). Les publicités disent souvent "grande vitesse" mais cela ne peut fonctionner à grande vitesse que sur de très courtes distances comme pour un modem situé juste à côté de l'ordinateur. Tous les fils utilisent un retour de masse commun et donc la technologie en paire torsadée (nécessaire à de grandes vitesses) ne peut être utilisée sans matériel supplémentaire. Cependant certains ordinateurs possèdent des interfaces plus modernes. Voyez 21.5.2 (successeurs de EIA-232).

Il est parfois décevant que la norme RS-232 de 1969 n'ait pas utilisé la technologie en paire torsadée qui aurait pu fonctionner à peu près 100 fois plus rapidement. Les paires torsadées sont utilisées dans les câbles téléphoniques depuis la fin du 19ème siècle. En 1888 (il y a plus de 100 ans) la "conférence sur le câble" a rapporté son attachement à la paire torsadée (pour les systèmes téléphoniques) et a avancé ses avantages. Mais plus de 80 ans après cette approbation par la "conférence du câble", RS-232 n'a pas réussi à l'utiliser. Puisque RS-232 était au départ faite pour connecter un terminal à un modem basse vitesse situé tout près, le besoin de grande vitesse et d'une longueur de transmission plus élevée n'a apparemment pas été perçu.

21.5.2 Successeurs de EIA-232

Un certain nombre de normes EIA ont été établies pour des vitesses plus élevées et des distances plus grandes en utilisant la technologie en paire torsadée (équilibrée). Une transmission équilibrée peut parfois être une centaire de fois plus rapide que EIA-232 non équilibrée. Pour une vitesse donnée, la distance (longueur maximale du câble) peut être beaucoup de fois plus grande avec de la paire torsadée. Mais les PC continuent d'être fabriqués avec l'EIA-232 "obsolète" puisque que cela fonctionne correctement avec les modems reliés aux lignes téléphoniques lentes, et cela fonctionne correctement avec les souris.

Une exception reste l'ordinateur Macintosh d'Apple avec son GeoPort EIA-232/EIA-422 qui fournit de la paire torsadée (équilibrée) pour la transmission et la réception. Il utilise un petit connecteur rond "mini-DIN". Il fournit aussi du EIA-232 traditionnel mais seulement à 5 volts (ce qui reste du EIA-232 légal). Cependant, à cause du fait que les Mac coûtent plus cher que les PC, on les utilise rarement comme ordinateur hôte pour des terminaux. Certains terminaux récents utilisent l'EIA-423 mais cela reste comme de l'EIA-232 non équilibré et on peut les relier à un port EIA-232. Cet EIA-423 ne fait que 5 volts, mais les spécifications donnent des vitesses plus élevées que pour EIA-232 (qui ne sera d'aucune aide sur une grande distance où c'est le non-équilibrage qu cause les interférences).

L'EIA-530-A (équilibré mais peut aussi être non équilibré) à 2 Mbits/s (équilibré) était fait pour remplacer EIA-232 mais on en a peu installé. Elle utilise le même connecteur à 25 broches que EIA-232. L'interface série à grande vitesse (HSSI = EIA-612/613, High Speed Serial Interface) utilise un connecteur à 50 broches et monte à peu près à 50 Mbits/s mais la distance est limitée à seulement quelques mètres. Le Bus Série Universel (USB, Universal Serial Bus) est construit dans des puces PCI. Il fait 12 Mbits/s sur une paire torsadée avec un connecteur à 4 broches (2 câbles fournissent le courant) mais il est aussi limité à des distances courtes d'au plus 5 mètres (cela dépend de la configuration).

21.5.3 Pilotes de lignes

Pour un terminal texte, les vitesses de EIA-232 sont suffisamment rapides mais la longueur de câble utilisable est souvent trop courte. La technologie équilibrée pourrait résoudre ce problème. La méthode courante pour obtenir une communication équilibrée avec un terminal texte est d'installer deux pilotes de ligne dans la liaison série pour convertir du non équilibré en équilibré (et vice-versa). Ce sont des appareils spécialisés et ils sont chers si on les achète neufs.

21.6 Synchronisation et synchrone

21.6.1 Comment on synchronise l'"asynchrone"

Dans EIA-232 il n'y a que deux états sur le fil de transmission (ou de réception) : marque (-12 V) ou espace (+12 V). Il n'y a pas d'état à 0 V. Ainsi une séquence de bits à 1 est tranmise avec uniquement du -12 V stable sans marqueur d'aucune sorte entre les bits. Pour que le récepteur détecte les bits individuels il doit toujours disposer d'un signal d'horloge qui est synchronisé avec l'horloge de l'émetteur. De telles horloges génèrent un "top" synchronisé avec chaque bit transmis (ou reçu).

En transmission asynchrone, la synchronisation est faite en encadrant chaque octet d'un bit de départ et d'un bit de stop (fait par le matériel). Le récepteur attend sur la ligne un bit de départ et quand il en détecte un il lance son top d'horloge. Il utilise ce top d'horloge pour mesurer le temps de lecture des 7, 8 ou 9 prochains bits. (C'est en fait un petit peu plus compliqué que cela puisqu'on prend en général plusieurs mesures pour un bit, ce qui demande des tops supplémentaires.) Ensuite le bit de stop est lu, l'horloge s'arrête et le récepteur attend le bit de départ suivant. Ainsi l'asynchrone est en fait synchronisé pendant la réception d'un seul octet mais il n'y a pas de synchronisation entre un octet et l'octet suivant.

21.6.2 Définir l'asynchrone par rapport au synchrone

L'asynchrone signifie "non synchrone". En pratique, un signal asynchrone représente ce que le port série asynchrone envoie et reçoit qui est un flux d'octets, chacun d'entre eux étant délimité par un bit de départ et un bit de stop. Le synchrone est à peu près tout le reste. Mais ceci n'explique pas les concepts de base.

En théorie, synchrone veut dire que les octets sont envoyés à vitesse constante l'un après l'autre en accord sur un top d'horloge. Il y a souvent un fil ou un canal séparé pour envoyer le top d'horloge. Les octets asynchrones peuvent être envoyés n'importe quand avec des intervalles de temps variés entre les octets (comme quelqu'un qui tape des caractères sur un clavier).

Il y a des situations limites qu'on doit classer comme synchrones ou asynchrones. Le port série asynchrone envoie souvent des octets dans un flux constant qui en ferait un cas synchrone mais comme il y aura encore les bits de départ et de stop (ce qui permet de les envoyer de manière indéterminée) on l'appelle asynchrone. Un autre cas est quand les octets de données (sans auncun bit de départ ou de stop) forment des paquets avec un espacement erratique entre un paquet et le suivant. On l'appelle synchrone puisque les octets à l'intérieur de chaque paquet doit être transmis de manière synchrone.

21.6.3 Communication synchrone

Ne vous-êtes vous jamais demandé ce qu'on faisait de toutes les broches inutilisées sur un connecteur 25 broches pour le port série ? La plupart d'entre eux sont utilisés dans une communication synchrone qu'on implémente rarement sur les PC. Il y a des broches pour les signaux de temporisation de synchronisation ainsi que pour un canal inverse synchronisé. La spécification EIA-232 est donnée à la fois pour les communications synchrones et asynchrones mais les PC utilisent une puce UART (Émetteur Récepteur Asynchrone Universel) comme un 16450, un 16550A ou un 16550 et ne peuvent faire de la synchronisation. Pour faire du synchrone on a besoin d'une puce USART ou équivalente où le "S" veut dire synchrone. Puisque le synchrone est un marché de niche, un port série synchrone est sûrement assez cher.

À côté de la partie synchrone de EIA-232, il y a plusieurs autres normes EIA synchrones. Pour EIA-232, trois broches sur le connecteur sont réservées pour les signaux d'horloge (de temporisation). Parfois c'est le rôl du modem de générer certains signaux de temporisation rendant l'utilisation de communications synchrones impossibles sans un modem synchrone (ou sans appareil qu'on appelle "éliminateur de modem synchrone" qui fournit les signaux de temporisation).

Bien que peu de ports série soient synchrones, la communication synchrone prend souvent place sur les lignes téléphoniques en utilisant des modems qui utilisent la correction d'erreurs V.42. Ceci enlève les bits de départ et de stop et place les octets de données dans des paquets ce qui donne une opération synchrone sur la ligne téléphonique.

21.7 Mode par blocs

21.7.1 Introduction au mode par blocs

Le mode par blocs est rarement utilisé sous Linux. En mode par blocs, quand quelqu'un tape sur un terminal, le résultat est sauvé dans la mémoire du terminal et n'est pas envoyé immédiatement à l'ordinateur hôte. De tels terminaux possèdent souvent des possibilités d'édition intégrées. Quand l'utilisateur appuie sur certaines touches (comme la touche envoi), ce qui a été sauvegardé dans la mémoire du terminal est envoyé à l'ordinateur hôte. Cependant les éditeurs vi et emacs sous Linux réagissent instantanément à l'appui de certaines touches, mais, dans la situation ci-dessus, si on appuie sur de telles touches, rien ne se passera puisque rien n'est envoyé quand on appuie sur une touche. Ainsi l'utilisation d'un terminal en mode par blocs ne permettra pas l'utilisation de tels programmes interactifs. La vieille interface vers les minis IBM utilise le mode par blocs (voyez 22.2 (terminaux IBM)) et donc beaucoup de terminaux IBM ne fonctionnent qu'en mode par blocs et sont aussi synchrones (voyez la section 21.6 (synchronisation et synchrone)).

21.7.2 Types de modes par blocs, formulaires

Le mode par blocs peut lui-même avoir divers sous-modes comme "page" (une page à la fois) et "ligne" (une ligne à la fois). Certains terminaux possèdent à la fois les modes de transmissions par blocs et les modes traditionnels par caractères, et on peut passer d'un mode à l'autre. Les terminaux asynchrones possédant des modes par blocs sont parmi les HP2622A, VT130, VT131, VT330, VT340 et Visual500. Beaucoup de modèles de terminaux plus récents peuvent émuler le mode par blocs. Les modes par blocs peuvent comprendre une possibilité de formulaires où l'ordinateur hôte envoie un formulaire au terminal. L'utilisateur le remplit alors et envoie la touche envoi qui ne renvoie que les données du formulaire à l'ordinateur hôte. Le formulaire lui-même (pas les données) est affiché à l'écran dans des champs protégés qui ne sont pas transmis à l'hôte.

21.7.3 Efficacité

Les modes par blocs enlèvent une bonne partie de la charge sur l'ordinateur hôte, surtout si le matériel de l'ordinateur hôte est fait pour les modes par blocs (comme c'est/c'était le cas sur les minis IBM). En mode caractère, chaque caractère tapé est envoyé immédiatement sur le port série et en général génère une interruption sur l'ordinateur hôte. L'hôte qui reçoit l'octet doit arrêter tout ce qu'il fait et va chercher ce caractère depuis le matériel du port. Même avec des UART qui possèdent des tampons matériels FIFO, le délai matériel ne représente normalement que le temps de transmission de 3 octets, donc une interruption est générée à chaque caractère tapé.

En vrai mode par blocs, un bloc de caractères long est reçu en n'utilisant qu'une interruption. Si on utilise le mode par blocs avec des ports série FIFO asynchrones, une interruption n'est nécessaire que tous les 14 octets puisqu'ils ont des tampons matériels de 16 octets. Ainsi la plus grande partie de la charge et du coût de la gestion des interruptions est éliminée et l'ordinateur a plus de temps à consacrer à d'autres tâches quand on utilise le mode par blocs.

On fait des économies significatives en mode par blocs si le terminal est relié à son hôte par l'intermédiaire d'un réseau. Sans le mode par blocs, chaque caractère (octet) tapé est envoyé dans son propre paquet avec tous les octets d'enveloppe (40 dans un paquet TCP/IP comme ceux utilisés sur l'Internet). En mode par blocs, un grand nombre de caractères est envoyé dans un seul paquet.

21.8 Livres sur EIA-232 (RS-232)

(Note : le premier couvre bien plus que EIA-232 uniquement.)

- Black, Uyless D.: Physical Layer Interfaces & Protocols, IEEE Computer Society Press, Los Alamitos, CA, 1996.
- Campbell, Joe: The RS-232 Solution, 2nd ed., Sybex, 1982.
- Putnam, Byron W.: RS-232 Simplified, Prentice Hall, 1987.
- Seyer, Martin D.: RS-232 Made Easy, 2nd ed., Prentice Hall, 1991.

21.9 Logiciels série

Voyez *logiciels série* `ftp://metalab.unc.edu/pub/Linux/system/serial/` pour les logiciels Linux sur les ports série avec getty et les moniteurs de ports.

22 Annexe D : Notes classées par marque

Voici des notes classées par marque qui étaient trop spécifiques à un certain terminal pour être mises autre part dans ce HOWTO. Si vous avez des informations à apporter pour un terminal particulier qui n'est pas couvert autre part, elles pourraient se trouver ici. Divers modèles et marques ont souvent beaucoup de choses en commun qu'on ne doit écrire qu'à un endroit. Il serait bien d'avoir, pour chaque modèle de terminal, un grand nombre de liens vers les documentations se rapportant à ce modèle (avec les séquences d'échappement). Il y a tellement de modèles de terminaux qu'une telle tâche serait relativement pénible et moi, David Lawyer (en 1998), n'ai aucunement l'intention de tenter cette aventure. Si les fabricants de terminaux mettaient au moins leurs manuels à disposition sur Internet, alors tout ceci ne serait pas nécessaire.

22.1 CIT

Les terminaux CIT ont été fabriqués au Japon dans les années 1980 pour CIE Terminals. On a cessé de les importer à la fin des années 1980. La société, CIE, fabrique encore des imprimantes CItoh (en 1997) mais n'a pas de pièces détachées pour ses terminaux arrêtés. Ernie au (714) 453-9555 à Irvine, Californie, vendait (en 1997) certaines pièces pour les modèles 224, 326, etc. mais n'a rien pour les 80 et 101. (Le document que vous lisez à l'heure actuelle a été principalement écrit sur le 101e.)

Pour sauver les paramètres de configuration pressez ^S en mode Set-Up. cit80 : contraste : molette à l'arrière du terminal, cit101e : luminosité : utilisez les touches fléchées haut/bas en mode Set-Up.

22.2 Terminaux IBM

Ne confondez pas les terminaux IBM avec les moniteurs IBM PC. Beaucoup de terminaux IBM n'utilisent pas l'ASCII mais à la place un code EBCDIC sur 8 bits. On dit que dans EBCDIC l'ordre de transmission des bits est inversée par rapport à la normale avec le bit de haut rang en premier. Les normes de communication avec les minis IBM sont un type de communication synchrone en mode par blocs (envoie de grands paquets de caractères). Deux normes sont "BISYNC" et "SNA" (qui comprend des normes de réseau). Beaucoup de leurs terminaux sont reliés avec du câble coaxial (RG62A/U) et les personnes naïves peuvent penser que le connecteur "BNC" sur le terminal est pour l'Ethernet (mais ce n'est pas le cas).

Alors que ce système IBM est en réalité plus efficace que ce qui est normalement utilisé sous Linux, les terminaux possédant ces caractéristiques IBM ne fonctionneront pas sous Linux. Cependant, certains terminaux IBM sont des terminaux ASCII asynchrones et devraient fonctionner sous Linux sur des PC. Les modèles 31xx peuvent fonctionner à l'exception que 317x et 319x ne sont pas des terminaux ASCII. Avant d'acquérir un terminal IBM, assurez-vous qu'il y a une entrée termcap (ou terminfo) pour celui-ci. S'il n'y en a pas, il ne fonctionnera probablement pas sous Linux. Même s'il y a une entrée terminfo, il peut ne pas fonctionner. Par exemple, il y a une entrée termcap pour les 327x mais le 3270 est un terminal synchrone EBCDIC.

Les modèles 3270 comprennent le 3278 (fin des années 1970), le 3279 avec la couleur et les graphiques, et le contrôleur de terminal 3274 (à peu près comme le 3174). On peut les utiliser à la fois pour BISYNC et SNA. Le 3290 a un écran séparé (séparé en quartiers).

Les terminaux synchrones IBM ne sont pas directement reliés au mini IBM mais sont reliés à un "contrôleur de terminaux" (qu'on appelle parfois "contrôleur de cluster" ou "contrôleur de communications"). Certains de ces contrôleurs peuvent convertir un signal synchrone en asynchrone et donc dans ce cas on pourrait relier indirectement un terminal synchrone à un ordinateur hôte de type Unix par son port série. Mais il reste un problème majeur qui est la transmission par blocs. Voyez la section 21.7 (mode par blocs).

22.2.1 IBM 3153

On dit que le port Aux est DCE et utilise un câble droit.

22.3 Teletypes

Ce sont des antiquités et ils représentent les terminaux les plus anciens. Ils ressemblent à des machines à écrire télécommandées mais sont grands et font du bruit. Fabriqués par Teletype Corp., les premiers modèles ont été faits dans les années 1920 et devancent l'ordinateur de plus de 30 ans. Les premiers modèles utilisaient des relais électro-mécaniques et des distributeurs rotatifs au lieu d'électronique. Leur code Baudot n'avait que 5 bits par caractère comparé à l'ASCII sur 7 bits. Voyez le livre "Small Computer Systems Handbook" par Sol Libes, Hayden Books, 1978 : pp. 138-141 ("Teletypes").

22.4 VT (DEC)

Digital Equipment Corporation a fabriqué les modèles VT renommés avec le VT100 couramment émulé. En 1995 ils ont vendu leur partie terminaux à SunRiver qui s'appelle maintenant Boundless Technologies. On trouvera plus d'informations sur le *site Web de Shuford* `http://www.cs.utk.edu/~shuford/terminal_index.html` . Les informations sur les produits actuels sont disponibles sur le site Web de Boundless. Voyez 19.2.1 (informations sur Internet).

VT220 : certains possèdent un connecteur BNC pour la sortie vidéo (pas pour l'entrée). Parfois les gens croient à tort que c'est pour une liaison Ethernet.

VT520 : il supporte le contrôle de flux DTR/DSR complet.

22.5 Wyse

Wyse possède certaines FAQ pour les terminaux dont les numéros sont en-dessous de 100 (comme le WY60). Voyez `http://www.wyse.com/service/faq/wysetterl.htm` . Pour les spécifications de terminaux plus récents voyez `http://www.wyse.com/terminal/` .

22.5.1 Wyse 99-GT

Voici les menus de configuration du Wyse99GT (fin des années 1980). Notez que TERM veut dire "terminaison" (caractère) et non "terminal".

```
Configuration du terminal WYSE 99-GT telle qu'utilisée à l'université
      d'Irvine, Californie par David Lawyer, avril 1990

                        F1 DISP:
COLUMNS=80              LINES=24                CELL SIZE=10 X 13
STATUS LINE=STANDARD    BACKGROUND=DARK         SCROLL SPEED=JUMP
SCREEN SAVER=OFF        CURSOR=BLINK BLOCK      DISPLAY CURSOR=ON
ATTRIBUTE=CHAR          END OF LINE WRAP=ON     AUTO SCROLL=ON
-----------------------------------------------------------------------------
                        F2  GENERAL:
PERSONALITY=VT 100      ENHANCE=ON              FONT LOAD=OFF
COMM MODE=FULL DUPLEX   RCVD CR=CR              SEND ACK=ON
RESTORE TABS=ON         ANSWERBACK MODE=OFF     ANSWERBACK CONCEAL=OFF
```

```
WIDTH CHANGE CLEAR=OFF  MONITOR=OFF            TEST=OFF
---------------------------------------------------------------------------
                        F3 KEYBRD:
KEYCLICK=OFF            KEYLOCK=CAPS           KEY REPEAT=ON
RETURN=CR               ENTER=CR               FUNCT KEY=HOLD
XMT LIMIT=NONE          FKEY XMT LIMIT=NONE    BREAK=170MS
LANGUAGE=US             MARGIN BELL=OFF        PRINTER RCV=OFF
---------------------------------------------------------------------------
                        F4 COMM:
DATA/PRINTER=AUX/MODEM    MDM RCV BAUD RATE=9600  MDM XMT BAUD RATE=9600
MDM DATA/STOP BITS=8/1    MDM RCV HNDSHAKE=NONE   MDM XMT HNDSHAKE=NONE
MDM PARITY=NONE           AUX BAUD RATE=9600      AUX DATA/STOP BITS=8/1
AUX RCV HNDSHAKE=NONE     AUX XMT HNDSHAKE=NONE   AUX PARITY=NONE
(Il y a un port principal (Modem=MDM) et un port auxiliaire (AUX)
---------------------------------------------------------------------------
                        F5 MISC 1:
WARNING BELL=ON         FKEY LOCK=OFF          FEATURE LOCK=ON
KEYPAD=NUMERIC          DEL=DEL/CAN            XFER TERM=EOS
CURSOR KEYS=NORMAL      MARGIN CTRL=0          DEL FOR LOW Y=ON
GIN TERM=CR             CHAR MODE=MULTINATIONAL
---------------------------------------------------------------------------
                        F6 MISC 2:
LOCAL=OFF               SEND=ALL               PRINT=NATIONAL
PORT=EIA DATA           SEND AREA=SCREEN       PRINT AREA=SCREEN
DISCONNECT=60 MSEC      SEND TERM=NONE         PRINT TERM=NONE
PRINT MODE=NORMAL       VT100 ID=VT100         POUND=US
---------------------------------------------------------------------------
F7 TABS: Vous devriez voir plusieurs caractères "T" séparés par 8 points.
    Sinon, appuyez sur backspace.
F8 F/KEYS: Vous ne verrez normalement pas de définition pour les touches de
    fonction ici (sauf si quelqu'un les a définies et sauvées). Ceci veut
    dire qu'elles génèreront normalement leurs valeurs par défaut (pas
    affichées ici). <ctrl><F5> montre la "définition par l'utilisateur" de la
    touche F5, etc.  F9 A/BACK: Normally not defined: ANSWERBACK =
F10 EXIT: La sélection de "DEFAULT ALL" rendra les paramètres par défaut
    définis en usine.
```

Astuces sur l'utilisation du Guide de l'Utilisateur du WY-99GT :

Notez qu'on peut trouver une grande partie de ce qui manque dans ce guide dans le Guide du Programmeur WY-99GT. L'émulation (personnalité) VT100 s'appelle ANSI et utilise les codes de touches ANSI comme indiqué à la page A-10 et plus, bien que le clavier soit en ASCII. Un sous-titre sur la page A-13 "clavier ASCII" s'applique aussi au VT100 parce qu'il a un sur-titre "Touches ANSI..." quelques pages avant. Mais tous les titres sur le clavier ASCII ne concernent pas le VT100 puisqu'ils tombent dans un sur-titre de personnalité non ANSI qu'on peut trouver quelques pages auparavant. L'annexe H est le "guide de commandes ANSI" sauf pour la personnalité VT52 (ANSI) qu'on trouve dans l'annexe G.

22.5.2 Wyse 150

Quand on sort du menu de configuration avec F12, l'appui sur espace change de "non" à "oui" pour sauver la configuration. La phrase à gauche de ce non/oui parle d'"alignement vertical" et n'a rien à faire avec ce non/oui pour sauver la configuration (interface de menu confuse).

Fin du HOWTO Text-Terminal