Manuel de Linux/MIPS

Ralf Bächle, `ralf@gnu.org` Traduction : Robert Jacolin, `rjacolin@yahoo.fr` 24 juin 2000

Cette FAQ décrit le portage vers l'architecture MIPS du système d'exploitation Linux, ainsi que des problèmes courants et propose des solutions, des possibilités (voire plus). Il tente aussi d'être un peu utile pour les autres personnes qui lirait cette FAQ dans l'espoir de trouver des informations qui, en fait, pourraient être détaillées ailleurs.

Contents

1 Qu'est-ce-que Linux/MIPS ?

Linux/MIPS est un portage du très répandu clône d'UNIX, Linux, vers l'architecture MIPS. Linux/MIPS fonctionne sur un grand nombre de systèmes, techniquement très différents, qui va des petits systèmes embarqués et des petits serveurs jusqu'aux gros serveurs et aux grosses machines bureautiques qui, au moins à l'époque où ils ont été introduit sur le marché, étaient les meilleurs de leur classes.

Les avantages de Linux/MIPS par rapport aux autres systèmes d'exploitation actuels sont :

- Le système Linux complet se compose uniquement de Logiciels Libres,
- L'excellent rapport Prix/Performance,
- L'existance d'une grande quantité de logiciels dont, encore un fois, une large part sont des Logiciels Libres,
- Une compatibilité binaire entre un nombre croissant de plates-formes,
- Une petite taille rendant Linux/MIPS approprié pour beaucoup de systèmes embarqués.

En résumé, Linux a été conçu et porté avec plaisir. Cependant, comme d'habitude, le temps que vous allez y consacrer pourra varier et vous devrez vérifier que Linux puisse s'adapter à votre projet, ce que ce document se propose de faire.

2 Obtenir cette FAQ

Vous pouvez télécharger ce document dans plusieurs formats (et en anglais, NDT):

- Une version HTML

  `http://oss.sgi.com/mips/mips-howto.html`
- Une version texte

  `http://oss.sgi.com/mips/mips-howto.txt`
- Une version postscript

  `http://oss.sgi.com/mips/mips-howto.ps`
- Une version Linux-Doc SGML

  `http://oss.sgi.com/mips/mips-howto.sgml`

Cette FAQ est ausi accessible sous forme de code source SGML par CVS anonyme sur oss.sgi.com. L'archive contient aussi un Makefile qui le convertira en plusieurs formats. Une version ASCII est réguliérement postée sur *comp.os.linux.answers* et les autres cannaux des manuels Linux.

3 Quel est le matériel supporté par Linux/MIPS ?

3.1 Les plates-formes matérielles

Plusieurs machines sont utilisables avec un quantité variable d'options CPU qui ne sont pas encore toutes supportés. Veuillez vérifier que votre type de CPU est supporté dans la section 3.2 (Types de processeurs). C'est une liste de machines qui fonctionnent sous Linux/MIPS, de systèmes où Linux/MIPS peut être porté ou de systèmes où l'on a intérêt à faire fonctionner Linux/MIPS.

3.1.1 Le PICA d'Acer

Le *PICA d'Acer* est dérivé de l'architecture *Mips magnum 4000*. Il possède un CPU R4400PC fonctionnant à 133 Mhz ou éventuellemnt à 150 Mhz plus une mémoire cache de second niveau de 512 Ko (éventuellement de 2 Mo); la carte gfx G364 du Magnum a été remplacé par une carte basée sur le S3 968. Le système est supporté à l'éxception du serveur X.

3.1.2 Les séries Baget/MIPS

Les séries Baget comprennent plusieurs machines possédant des processeurs R3000 : le Baget 23, le Baget 63 et le Baget 83. Les Baget 23 et 63 ont des cartes mères BT23-201 ou BT23-202 avec respectivement un R3500A (qui est, à la base, un composant R3000A) à 25 Mhz et un R3081E à 50 Mhz. La carte BT23-201 possède un bus VME et des puces VIC068, VAC068 comme contrôleurs systèmes. La carte BT23-202 possède un bus PCI en interne et une bus VME en externe. Le support de la carte BT23-201 a été fait par *Gleb Raiko (rajko@mech.math.msu.su)* et *Vladimir Roganov (vroganov@msiu.ru)* avec l'aide de *Serguei Zimin (zimin@msiu.ru)* . Le support du BT23-202 est en développement avec un Baget 23B qui est composé de 3 cartes BT23-201 avec un bus VME partagé.

Le Baget 83 est mentionné ici uniquement pour être éxhaustif. Il possède uniquement 2 Mo de RAM et il est trop petit pour faire tourner Linux. Le code du Baget/MIPS a été fusionné avec le portage des stations DEC ; le source pour ces deux plates-formes est sur `http://decstation.unix-ag.org/` .

3.1.3 Le Qube et le Raq de Cobalt

Les séries de produits Qube Cobalt sont des systèmes de serveurs headless de faible coût basés sur un IDT R5230. Cobalt a développé sa propre variante de Linux/MIPS pour répondre aussi bien que possible aux besoins particuliers du Qube. Au départ, le noyau du Qube a été dérivé du noyau de Linux/MIPS 2.1.56, puis ramené à la version 2.0.30 pour la stabilité, enfin il a été optimisé. Les noyaux pour le Cobalt sont accessibles sur le site ftp de Cobalt `http://www.cobaltnet.com` . Le support du Qube de Cobalt n'a jamais été intégré dans les noyaux officiels 2.1.x de Linux/MIPS.

3.1.4 Les machines NEC

Les machines uni-processeur NEC sont des systèmes *PICA d'Acer*, voir cette section pour plus de détails. Les systèmes SMP sont diférents à cause du fait d'avoir plusieurs processeurs. Les développeurs de Linux/MIPS n'ont pas les documentations techniques nécessaires pour écrire un OS. Aussi longtemps qu'il n'y aura pas de changements, ce portage restera plus ou moins un bouchon remarqué faisant obstacle au portage vers les systèmes SMP de NEC.

3.1.5 Les plates-formes basées sur les VR41xx de NEC

Le projet VR Linux fait le portage de Linux vers du matériel basé sur les micro-processeurs VR41xx de NEC. La plupart de ces matériels étaient, à l'origine, destinés pour faire tourner Windows CE. Le projet a produit des noyaux qui fonctionnent avec des drivers de bases pour le Vadem Clio, la Casio E-105, l'Everex Freestyle, et bien d'autres. Pour de plus amples informations, veuillez consulter le site `http://linux-vr.org/` .

3.1.6 Les plates-formes TMPR39xx de Toshiba/PR31700 de Philips

Semblable aux VR41xx, le matériel avec ces processeurs ont été, à l'origine, destinés pour faire tourner Windows CE. Cependant, il y a des noyaux fonctionnels avec des drivers de base pour le *Sharp Mobilon* et la *série C de Compaq*. Le support d'autre matériels est en cours. Le code fait partie du projet VR Linux et donc de plus amples informations peuvent être trouvé sur `http://linux-vr.org/` .

3.1.7 Le Netpower 100

Le *Netpower 100* est apparamment un *PICA d'Acer* déguisé. Il devrait être, par conséquence, supporté mais cela n'a pas été testé. S'il y a un problème c'est probablement lors de la détection de la machine.

3.1.8 La Nintendo 64

La *nintendo 64* est une console de jeu basé sur un R4300 avec 4 Mb de RAM. Ses puces graphiques ont été développé par Silicon Graphics pour nintendo. A l'heure actuelle, ce portage est un réve de joueur et continuera de l'être tant que Nintendo ne décidera pas de publier les informations techniques necessaires. La question qui subsiste est de savoir si c'est une bonne idée.

3.1.9 Le Challenge S de Silicon Graphics

Cette machine est très similaire à l'Indy ; la différence est qu'elle ne possède pas de clavier ni de carte GFX mais un adaptateur basé sur un WD33C95 SCSI supplémentaire. Cet adaptateur WD33C95 n'est pas supporté pour l'instant.

3.1.10 L'Indigo de Silicon Graphics

Cette machine n'est mentionée que parce que certaine personne la confonde avec les Indys ou l'Indigo 2. L'Indigo posséde une architecture différente, basée sur un R3000 cependant, et n'est pas encore supporté.

3.1.11 L'Indigo2 de Silicon Graphics

Cette machine est le successeur de l'Indigo et elle est très semblable à l'Indy. Elle est maintenant supportée, bien qu'ellz péche en bien des points. Vous devrez utiliser une console série. Si vous avez une Indigo2 et si vous désirez encore y faire tourner Linux, contactez soit *Florian Lohoff (flo@rfc822.org)*

soit *Klaus Naumann (spock@mgnet.de)* .

3.1.12 L'Indy de Silicon Graphics

L'Indy est, en ce moment, l'unique machine supporté parmi (la plupart) des machines de Silicon Graphics. La seule carte graphique supportée est la carte Newport c'est-à-dire la "XL". L'Indy existe avec un grand nombre d'options pour le CPU à des taux d'horloge variés, tous étant supportés. Il existe aussi maintenant un serveur X écrit par *Guido Guenther (guido.guenther@gmx.net)* . Si vous pouvez utiliser la console de Newport sur votre Indy, il doit être possible aussi d'utiliser le serveur X. Il est basé sur XFree86 4.0 et il fonctionne courament à une vitesse de tortue mais semble bien fonctionner. Si vous désirez l'essayer, jetez un oeil sur `http://honk.physik.uni-konstanz.de/~agx/mipslinux/x/` .

Quantité étrange d'espace mémoire Lors du boot, le noyau de l'Indy reporte la mémoire utilisable dans un message du type :

```
Memory: 27976k/163372k available (1220k kernel code, 2324k data)
```

Cette importante différence entre la première paire de nombres vient de l'existance d'une zone de 128 Mo dans l'espace adressable de la mémoire de l'Indy qui reflète les 128 premiers Mo de mémoire. La différence entre les 2 nombres sera toujours proche de 128 Mo et n'indique pas un quelconque problème. Les noyaux depuis la version 2.3.23 ne compte plus ce trou de 128 Mo.

Problèmes avec la PROM de l'Indy Plusieurs personnes ont rapportés ces problèmes avec leurs machines après une mise à niveau typiquement à cause de parties en trop. Il existe plusieurs versions de PROM pour les Indys. Les machines avec de vieilles versions de leur PROM, qui ont été mis à niveau vers une variante plus récente d'un CPU comme un module R4600SC ou un R5000SC, peuvent se planter pendant l'auto-test avec un message d'erreur du type :

```
Exception: <vector=Normal>
Status register: 0x30004803<CU1,CU0,IM7,IM4,IPL=???,MODE=KERNEL,EXL,IE>
Cause register: 0x4000<CE=0,IP7,EXC=INT>
Exception PC: 0xbfc0b598
Interrupt exception
CPU Parity Error Interrupt
Local I/O interrupt register 1: 0x80 <VR/GIO2>
CPU parity error register: 0x80b<B0,B1,B3,SYSAD_PAR>
CPU parity error: address: 0x1fc0b598
NESTED EXCEPTION #1 at EPC: 9fc3df00; first exception at PC: bfc0b598
```

Dans ce cas, vous devez mettre à niveau la PROM de votre machine vers une version plus récente ou retourner vers un version plus ancienne du CPU. En général, les modules R4000SC ou R4400SC devraient fonctionner de cette manière. Juste pour être bien clair, ceci est un problème n'ayant aucun rapport avec Linux. C'est uniquement mentionné ici parce que plusieurs utilisateurs de Linux nous ont posé la question.

Le supprot d'ELF avec des vieilles versions de PROMs Les vieilles versions de PROM ne connaissent pas le format binaire ELF que le noyau de Linux utilise, ce qui l'empêche de booter directement sur Linux. La solution préférable à cela reste évidement une mise à niveau de la PROM. Vous pouvez aussi utiliser Sash d'IRIX 5 ou une version plus récente pour charger le noyau. Sash sait comment charger les binaires ELF et ne se préoccupe pas de savoir si c'est un noyau IRIX ou Linux. Il suffit de taper simplement "Sash" à partir du moniteur de la PROM. Vous obtiendrez un autre prompt shell, celui de Sash cette fois-ci. Maintenant lancez Linux comme d'habitude.

Sash peut lire les systèmes de fichiers EFS ou XFS ou lire le noyau avec bootp / tftp. Cela veut dire que si vous avez l'intention d'utiliser Sash pour lancer le noyau à partir d'un disque local, vous devrez encore posséder une installation minimale d'IRIX sur votre système.

Pourquoi y-a-t-il autant de mémoire réservée sur mon Indy ? Lors du démarrage, le message "Memory: ..." sur un Indy indique qu'il y a 128 Mo de RAM réservé. C'est normal ; de même que l'architecture PC a un trou dans son espace d'adressage mémoire entre 640 Ko et 1024 Ko, l'Indy possède une zone de 128 Mo dans son espace mémoire où les 128 premiers Mo de sa mémoire est dupliqué. Linux le sait et ignore simplement cette zone mémoire, ce qui explique ce message.

3.1.13 Les Silicon Graphics Origin 200 et 2000

Ralf Bächle (ralf@gnu.org)

et une équipe d'employés de SGI travaillent actuellement sur un portage vers l'Origin 200. Bien qu'il soit encore que dans les étapes initiales, il fonctionne en mode mono-processeur et multi-processeurs et possède des pilotes pour la carte Ethernet IOC3 et l'adaptateur SCSI fourni avec. Le code peut être pris dans l'arbre CVS de Linux/MIPS.

3.1.14 Les Silicon Graphics Onyx 2

L'Onyx 2 est, à la base, un Origin 2000 avec du matériel graphique supplémentaire. A partir de maintenant,le support de Linux pour le matériel graphique n'a pas été décidé. En dépit de ca, Linux devrai fonctionner aussi bien qu'une configuration headless Origin 2000.

3.1.15 Silicon Graphics Power Series

C'estune très vieilles séries des systèmes R3000 SMP. Il n'existe pas de documentation sur le matériel de ces machines, peu d'entre elles existant encore, le matériel est bizarre. Pour faire court, les chances pour que Linux tourne un jour sur l'une d'elles sont proches de zéro. Non pas que l'on veuille décourager des volontaires ...

3.1.16 Consoles Série sur les machines SGI

Assurez-vous que le noyau que vous utilisez inclus le driver approprié pour une interface série et une console série. Initialisez la variable d'environnement *console* ARC soit avec la valeur *d1* soit avec la valeur *d2* pour les Indy et les Challenge S en fonction de l'interface série que vous allez utiliser comme console.

Si vous avez le problème d'affichage de tous les messages du noyau sur la console série lors du démarrage alors que plus rien n'est affiché à partir du début de la phase d'init, alors vous avez probablement une mauvaise configuration pour votre /dev/console. Vous pourrez trouver de plus amples informations à ce sujet dans la documentation du source du noyau de Linux ; il est situé dans le répertoire /usr/src/linux/Documentation/serial-console.txt si vous avez installé le source du noyau.

3.1.17 Les autres machines de Silicon Graphics

A l'heure actuelle, aucune machine Silicon Graphics n'est supportée. Ceci s'applique aussi aux systèmes basés sur les *très* vieux Motorola 68k.

3.1.18 La Playstation de Sony

La Playstation de Sony est basée sur un R3000 dérivée et utilise un ensemble de composants graphiques dévelopé par Sony lui-même. Alors que la machine est, en théorie, capable de tourner sous Linux, un portage semble difficile, puisque Sony n'a toujours pas fourni les informations techniques nécessaires. Cela met de côté la question de l'intêret du portage. Donc, en résumé, rien ne s'est passé jusqu'ici alors que beaucoup de gens ont montré leur intérêt en essayant Linux sur une Playstation.

3.1.19 SNI RM200C

A l'inverse du RM200 (voir en-dessous), cette machine possède des slots EISA et PCI. Le RM200 est supporté à l'exception du controleur SCSI NCR53c810A intégré.

3.1.20 SNI RM200

Si votre machine possède à la fois des slots EISA et PCI, alors c'est un RM200C ; Consultez la section précédente s'il vous plaît. A cause de légères diffèrences architecturales entre le RM200 et le RM200C, cette machine n'est pas encore supportée dans les sources officiels. *Michael Engel (engel@numerik.math.uni-siegen.de)* a réussi à faire fonctionner son RM200 partiellement mais les patches n'ont pas encore été inclus dans les sources de Linux/MIPS officiels.

3.1.21 SNI RM300C

Le RM300 est techniquement très similaire au RM200C. Il devrai être supporté par le noyau courant de Linux, mais nous n'avons encore reçu aucun signalement.

3.1.22 SNI RM400

Le RM400 n'est pas supporté.

3.1.23 SNI RW320

Cette machine est une variante OEM d'un SGI Indigo et, par conséquent, elle n'est pas encore supportée.

3.1.24 Algorithmics P4032

Le portage de l'Algorithmics P4032 tourne encore, lors de la redaction de ce document, sous Linux 2.1.36.

3.1.25 Algorithmics P5064

Le P5064 est, à la base, une variante 64 bits du P4032 basé sur un R5000. Un portage est en cours.

3.1.26 DECstation series

Pendant la fin des années 80 et au début des années 90, Digital (maintenant Compaq) a construit une station de travail basée sur les MIPS appelée respectivement DECStation et DECsystem. D'autres machines basées sur des x86 ou des Alphas ont été vendu sous le nom DECstation, mais ils ne sont malheureusement pas le sujet de cette FAQ. Le support des DECstations est encore en cours de développement, débuté par Paul M. Antoine. A l'heure actuelle, la plupart du travail est fait par *Harald Koerfgen (Harald.Koerfgen@home.ivm.de)* et par d'autres personnes. Sur Internet, des informations sur les DECstations peuvent être trouvé sur le site `http://decstation.unix-ag.org/` .

La famille des DECstations couvre les DECstations 2100 avec une puce R2000/R2010 à 12 MHz jusqu'au DECstation 5000/260 avec un R4400SC à 66 MHz.

Les modèles des DECstations suivants sont activement supportés :

- 2100, nom de code PMAX
- 5000/xx (DECstation Personnel), nom de code MAXine
- 5000/1xx, nom de code 3MIN
- 5000/200, nom de code 3MAX

- 5000/2x0, nom de code 3MAX+
- 5900/2x0 (identique au 3MAX+).

Ces modèles de DECstations sont orphelins parce que personne ne travaille dessus, alors que leur support peut être relativement facile à finir.

- 3100, identique au 2100 sauf le R2000A/R2010A @ 16 MHz
- 5100, nom de code MIPSMATE, presque identique au 2100 mais avec une puce R3000/R3010.

Les autres machines de la famille des DECstations, à part ceux basés sur x86, devraient être considéré comme des VAXen avec un CPU remplacé par un CPU MIPS. Il n'y a aucune information existante sur ces machines et le support de ces machines est improbable à moins que le portage des VAXLinux renait de ses cendres. Ce sont les :

- 5400, nom de code MIPSFAIR
- 5500, nom de code MIPSFAIR2
- 5800, nom de code ISIS

3.1.27 Mips Magnum 4000 / Olivetti M700-10

Ces deux machines sont presque complétement identiques. Revenons lors de l'initiative d'ACE, Olivetti a pris une license du concept Jazz et a mis sur le marché la machine avec comme système d'exploitation Windows NT. MIPS Computer Systems, Inc a acheté lui-même le concept Jazz et l'a mis sur le marché avec la série de machines MIPS Magnum 4000. Les systèmes Magnum 4000 ont été mis sur le marché avec comem système d'exploitation Windows NT et RISC/os.

Le microcode de la machine dépend du système d'exploitation qui a été installé. Linux/MIPS supporte uniquement le microcode "little endian" sur ces deux types de machines. Puisque le M700-10 n'a été mis sur le marché uniquement en tant que machine NT, toutes ces machines ont ce matériel installé. Le cas du MIPS Magnum est quelque peu plus complexe. Si votre machine a été configuré en "big endian" pour RISC/os alors vous devez recharger le microcode "little endian". Ce microcode était, à l'origine, inclus sur une disquette lors de la livraison de chaque Magnum. Si vous ne possédez plus la disquette, vous pouvez la télécharger par ftp anonymes sur le site `ftp://ftp.fnet.fr` .

Il est possible de reconfigurer les M700 pour des opérations headless en positionnant les variables d'environnement du matériel ConsoleIn et ConsoleOut sur mluti()serial()term(). Essayez aussi la commande *listdev* qui listera les périphériques ARC existants.

Dans bien des cas, comme lorsque la carte graphique G364 est absente alors que la console est encore configurée pour l'utilisation graphique normale, il sera nécessaire de modifier le cavalier de configuration JP2 sur la carte mère. Après le prochain reset, la machine redemarrera sur la console COM2.

3.1.28 MIPS Magnum 4000SC

Le Mips Magnum 4000SC est semblable au Magnum 4000 (voir ci-dessus) sauf qu'il utilise un CPU R4000SCC.

3.2 Les types de processeur

3.2.1 La famille des R2000, R3000

Le R2000 est le processeur MIPS original. C'est un processeur 32 Bits qui avait une fréquence de 8 MHz sortie en 85 lorsque les premiers processeurs MIPS arrivérent sur le marché. Les versions suivantes furent cadencées plus rapidement : par exemple, le 53000 est un reconception du R2000 100% compatible, juste cadencé plus rapidement. A cause de leur haute compatibilité, lorsque ce document mentionne le R3000, dans bien des cas, les mêmes faits s'appliquent aussi aux R2000. Le R3000A est, à la base, un R2000 avec un FPU R3010 et 64 K de cache cadencé jusqu'à 40 MHz et intégré dans la même puce.

Harald Koerfgen (Harald.Koerfgen@home.ivm.de) et *Gleb O. Raiko (raiko@niisi.msk.ru)*

ont tous les deux, de façon indépendante, travaillé sur des patches pour les processeurs R3000. Leur travail a été fusionné et intégré dans les sources officiels de Linux/MIPS depuis juillet 1999. Actuellement, Linux supporte les processeurs R3000 ainsi que des variantes comme le R3081 et le TMPR3912/PR31700.

3.2.2 R6000

Parfois, des personnes confondent le R6000, qui est un processeur MIPS, avec le RS6000, une série de stations de travail créée par IBM. Donc, si vous lisez ces lignes en espérant trouver des informations sur l'utilisation de Linux sur des machines IBM, vous lisez le mauvais document.

Le R6000 n'est pas supporté pour l'instant. C'est un processeur MIPS 32 Bits ISA 2 et c'est un morceau de silicon plutôt intéressant et bizarre. Il a été développé et produit par une entreprise appelée *BIT Technology*. Plus tard, NEC repris la production des semiconducteurs. Il était construit avec la technologie ECL, la même technologie qui était et qui est encore utilisé pour construire des puces extrêmement rapide comme celles utilisées dans les ordinateurs Cray. Le processeur possède son propre TLB implémenté comme une partie des dernières paires de lignes du cache primaire externe, une technologie appelée *tranche TLB (TLB slice)*. Ce qui signifie que son MMU est substantiellement différent de ceux de la série des R3000 ou des R4000, ce qui est aussi une des raisons pour laquelle le processeur n'est pas supporté.

3.2.3 La famille des R4000 et R5000

Linux supporte la plupart des membres de la famille des R4000. Actuellement, ce sont le R4000PC, le R4400PC, le R4300, le R4600, le R4700, le R5000, le R5230 et le R5260. Beaucoup d'autres fonctionnent probablement aussi bien.

Ceux qui ne sont pas supportés, ce sont les CPU R4000MC et R4400MC (ce sont des systèmes multiprocesseurs), de même que les système R5000 avec un cache de second niveau controlé par le CPU. Cela signifie que le cache est contrôlé par le R5000 lui-même à la différence des controleurs de cache externe. La différence est importante car, à la différence des autres systèmes, particulièrement les PCs, sur les MIPS, le cache est visible dans l'architecture et nécessite d'être contrôlé de façon logiciel.

Remerciements particuliers pour *Ulf Carlsson (ulfc@engr.sgi.com)* qui a fourni le module CPU pour deboguer le support du R4000SC / R4400SC.

3.2.4 R8000

Le R8000 n'est pas supporté, à l'heure actuelle, d'une part parce que ce processeur est relativement rare et qu'il n'a été utilisé que dans quelques machines de SGI, d'autre part parce que les développeurs de Linux/MIPS ne possèdent pas d'une machine de ce type.

Le R8000 est un morceau de silicon plutôt intéressant. A la différence des autres membres de la famille MIPS, c'est un ensemble de 7 puces. Son cache et son architecture TLB est assez différent des autres membres de la famille MIPS. Il est né d'un rapide "hack" pour que les virgules flotantes redeviennent le fer de lance des Silicon Graphics avant que le R10000 soit terminé.

3.2.5 R10000

Le R10000 est supporté dans le noyau mips64 qui est actuellement supporté par les IP22 (l'Indy de SGI, le Challenge S et l'Indigo 2) et l'Origin.

A cause de la très grande difficulté pour gérer la manière de fonctionner de ce processeur dans des systèmes sans cache cohèrent, cela va prendre probablement encore un certain temps avant que nous supportions ce processeur pour de tels systèmes. A partir de maintenant, ces systèmes sont les SGI O2 et Indigo.

3.3 Matériel qui ne sera jamais supporté

3.3.1 IBM RS6000

Comme son nom l'indique, c'est une machine IBM qui est basé sur la série de processeur RS6000 et, en tant que tel, ils ne font pas partie du projet Linux/MIPS. Les gens confondent souvent l'IBM RS6000 avec l'architecture MIPS R6000. Cependant, le projet Linux/PPC doit s'en occuper. Consultez le site

`http://www.linuxppc.org/` pour de plus amples informations.

3.3.2 VaxStation

Comme son nom l'indique déjà, cette machine est un membre de la famille des VAX de Digital Equipment. On le mentionne ici parce qu'il est souvent confondu avec la famille des DECstation basé sur le MIPS de Digital à cause des types de numéros similaires. Malheureusement, le VaxStation, de même que la famille entière des VAX, n'est pas supporté pour l'instant.

3.3.3 SGI VisPC

C'est un système basé sur les x86, par conséquent, il n'est pas couvert par cette FAQ. Cependant, pour faciliter vos recherches, voici quelques infos. *Ken Klingman (kck@mailbox.esd.sgi.com)*

posté le 17 Janvier 1999 sur la liste de discussion Linux de SGI :

```
Nous y travaillons. Nous terminons actuellement de mettre le support
du niveau de base dans la release de la 2.2. Les logiciels uniquement basé sur
X et OpenGl devrait suivre relativement rapidement, mais le matériel accéléré
pour OpenGL n'est pas encore planifié. Voir www.precisioninsight.com pour des
nouvelles sur le matériel accéléré pour OpenGL.
```

Pour plus d'informations, voir la Documentation/ du noyau de Linux à partir de la version 2.2 ou supérieure. Il y a des informations supplémentaires sur le web à l'adresse `http://oss.sgi.com/`. Notez que le personnel de SGI/MIPS et de SGI/Intel travaille indépendemment de chacun des autres, par conséquent, les sources sur le CVS anonyme sur oss.sgi.com peut ou ne peut très bien ne pas fonctionner pour les machines Intel ; nous n'avons pas testé cela.

3.3.4 Les machines basé sur le 68k de Motorola comme l'Iris 30000

Ce sont de *très* vieilles machines, probablement agés de plus de 10 ans maintenant. Comme ces machines ne sont pas basé sur des processeurs MIPS, ce document est le mauvais endroit pour y chercher des informations. Cependant, dans le but de vous faciliter les choses, ces machines ne sont pas supportées actuellement.

4 Les distributions Linux

4.1 RedHat

Pour les MIPS, il y a la Rough Cuts Linux, anciennement Hard Hat Linux, qui est plus qu'un simple portage d'une Red Hat Linux 5.1 sur MIPS. Il est possible de l'obtenir sur `ftp://oss.sgi.com/pub/hardhat` .

Il est aussi empaqueté avec le support de l'architecture M68k, UltraSparc et PowerPC dans un paquetage appelé "Rough Cuts" élaboré par Red Hat, et on peut l'obtenir là où les produits Red Hat sont vendus. C'est une manière très pratique de l'obtenir sans avoir à télécharger 280 Mo. Vous pouvez commander Rough Cuts directement chez Red Hat à l'adresse `http://www.redhat.com/product.phtml/RC1000` .

De plus, il existe une distribution basé sur la Red Hat 5.2 qui a pour cible les Qubes de Cobalt ; ses binaires fonctionneront parfaitement sur d'autres architectures MIPS et sont accessibles sur

`ftp://intel.cleveland.lug.net/pub/Mipsel` .

`ftp://bolug.uni-bonn.de/mips` possède plusieurs paquetages rpm Redhat 6.0 et 6.1.

4.2 Debian

Un portage de la Debian est en cours. Les efforts actuels ont été amorcé en utilisant SGI/Linux comme base, et dpkg compile en natif avec peu de changements. En plus de la version SGI, des personnes ont montré leur intérêt pour des plateformes "little endian". Gardez un oeil sur la page du portage Debian-MIPS,

`http://www.debian.org/ports/mips/` pour le développement.

4.3 Simple

Cette distribution a été faite, jusqu'ici, uniquement pour les systèmes "big endian". Elle est hautement expérimentale, pour une utilisation pour des développeurs pour tester les dernières versions de gcc, de binutils, de la glibc et du noyau. C'est la seule distribution basée sur la glibc 2.2 disponible pour MIPS. Vous pouvez toujours récupérer la dernière version de cette distribution et les notes de publications qui l'accompagnent sur `ftp://oss.sgi.com/pub/linux/mips/mips-linux/simple` . Un système de cross-compilation est également disponible pour aider le développement.

5 Ressources Linux/MIPS sur le net

5.1 Serveurs FTP anonymes

Les deux serveurs FTP anonymes principaux pour Linux/MIPS sont :

oss.sgi.com

Ce serveur devrait satisfaire presque tous vos désirs ftp sur Linux/MIPS. Vraiement.

ftp.fnet.fr

Ce serveur n'est actuellement plus tout à fait à jour ; il est inclus ici pour être exhaustif et pour les gens qui sont intéressés par des logiciels préhistoriques.

Sur tous ces serveurs ftp, il existe une liste de sites mirroirs que vous devrez plutôt utiliser pour avoir des accès plus rapide.

Une autre source de binaires MIPS "little endian" se situe sur *ftp://intel.cleveland.lug.net/pub/Mipsel* qui contient la plupart des versions très récentes des binaires pour la Redhat avec les binaires du Cobalt.

5.2 Serveurs CVS anonymes

Pour ceux qui veulent toujours rester au bord du goufre et qui veulent éviter d'avoir à télécharger des patches ou des archives tar complètes, nous avons aussi un serveur CVS anonymes. En utilisant CVS, vous pouvez rapatrier l'arbre des sources de Linux/MIPS avec les commandes suivantes :

```
 cvs -d :pserver:cvs@oss.sgi.com:/cvs login
 (Nécessaire uniquement lors de la première utilisation du CVS anonyme, le mot
de passe est "cvs")
 cvs -d :pserver:cvs@oss.sgi.com:/cvs co <repository>
```

où vous insérez linux, libc, gdb ou faq à la place de <repository>. L'autre archive CVS importante de la communauté Linux est vger.rutgers.edu où beaucoup de code est centralisé avant d'être envoyé à Linus pour distribution. Bien que vger lui-même n'offre plus d'accès anonyme, il y a des sites mirroirs qui fournissent un accès anonyme. Pour plus de détails sur la manière d'y accéder, voir `http://cvs.on.openprojects.net/` . Les modules qui ont de l'intérêt sont "linux", "modutils", "pciutils", "netutils".

5.3 Serveurs Web

Les deux principaux serveurs web pour Linux/MIPS sont :

`http://oss.sgi.com/mips`

Ce serveur couvre la presque totallité de Linux/MIPS ; il est plutôt centrée sur SGI mais comme Linux/MIPS essaye d'être identique sur toute les plateformes, la plupart de ces informations restent intéressantes pour tous les utilisateurs.

`http://lena.fnet.fr`

Ce serveur n'est actuellement plus très à jour ; il est inclu ici principalement pour être exhaustif.

Tous ces serveurs possèdent des mirroirs éparpillés à travers le monde ; vous devrez sans doute en utiliser un pour obtenir de meilleures performances.

5.4 Serveur Web CVS

Via `http://oss.sgi.com/mips/cvsweb` vous avez un accès direct aux sources du nouveau noyau Linux/MIPS et à quelques autres projets gérés dans la même archive CVS. L'interface intuitive vous permet de suivre le développement avec des clics de souris.

5.5 Listes de discussions

Il existe trois listes de discussions sur Linux/MIPS :

linux-mips@fnet.fr

Cette liste de discussions est utilisée pour des communications de toutes sortes autres qu'à propos de SGI. Les inscriptions sont gérés par une personne ; envoyez votre requêtes d'inscription à *linux-mips-request@fnet.fr* . Vous pouvez vous desinscrire à partir de la liste de discussions en envoyant *unsubscribe* *<votre-adresse-mail>* à la même adresse.

linux-mips@oss.sgi.com

Cette liste de discussions possède actuellement le plus de traffic. Elle est plus ou moins centrée sur SGI mais elle a pourtant de l'intérêt, spécialement pour les développeurs puisqu'un grand nombre d'ingénieurs de SGI sont inscrits à la liste. L'inscription à cette liste se fait par *Majordomo (majordomo@oss.sgi.com)* ; envoyez simplement un courrier électronique avec les mots *subscribe linux*. Pour se désinscrire, envoyez *unsubscribe linux*. Notez que vous devez vous inscrire pour pouvoir poster un courrier ; l'augmentation de spasm nous a forcé cette politique. Pour de plus amples informations, voir aussi `http://oss.sgi.com/mips/email.html` .

linux-mips@vger.rutgers.edu

Cette liste de discussions a vraiment très peu de traffic puisque la plupart des gens ont tendance à utiliser une des listes de discussions ci-dessus. Les inscriptions sont gérées par *Majordomo (majordomo@vger.rutgers.edu)* ; envoyez juste un courrier électronique avec les mots *subscribe linux-mips*. Pour se désinscrire envoyez un courrier avec *unsubscribe linux-mips*.

5.6 Canal IRC

Il existe un canal IRC nommé #mipslinux pour Linux/MIPS qui peut être trouvé sur irc.openprojects.net.

6 L'installation de Linux/MIPS et ses problèmes classiques

6.1 Le boot sur NFS échoue

En général, cela vient du fait que la personne a décompressé l'archive tar sous IRIX, et pas sous Linux. Puisque la représentation des fichiers des périphériques à travers NFS n'est pas standardisé entre les Unices, cela échoue. Le symptome est le blocage du système avec le message d'erreur "Warning: unable to open an initial console." juste après le montage du système de fichiers NFS.

Pour le moment, il faut utiliser un système Linux (pas nécessairement un MIPS) pour décompresser l'archive d'installation sur un serveur NFS. Le serveur NFS lui-même peut être n'importe quel UNIX.

6.2 Des noyaux compilés plantent au démarrage

Lorsque je construit mon propre noyau, il plante. Sur un Indy, le message lors du plantage ressemble à ce qui suit ; le même problème apparaît aussi sur d'autres machines mais ce qui est affiché diffère complétement.

```
Exception: <vector=UTLB Miss>
Status register: 0x300004803<CU1,CU0,IM4,IPL=???,MODE=KERNEL,EXL,IE>
Cause register: 0x8008<CE=0,IP8,EXC=RMISS>
```

```
Exception PC: 0x881385cc, Exception RA: 0x88002614
exception, bad address: 0x47c4
Local I/O interrupt register 1: 0x80 <VR/GIO2>
Saved user regs in hex (&gpda 0xa8740e48, &_regs 0xa8741048):
  arg: 7 8bfff938 8bfffc4d 880025dc
  tmp: 8818c14c 8818c14c 10 881510c4 14 8bfad9e0 0 48
  sve: 8bfdf3e8 8bfffc40 8bfb2720 8bfff938 a8747420 9fc56394 0 9fc56394
  t8 48 t9 8bfffee66 at 1 v0 0 v1 8bfff890 k1 bad11bad
  gp 881dfd90 fp 9fc4be88 sp 8bfff8b8 ra 88002614

PANIC: Unexpected exception
```

Ce problème vient d'une erreur non résolue encore dans le module Binutils dans les versions supérieures à la version 2.7. Pour le résoudre, changez la ligne suivante dans le fichier arch/mips/Makefile

```
LINKFLAGS       = -static -N
```

en :

```
LINKFLAGS       = -static
```

6.3 Le démarrage du noyau sur l'Indy échoue avec des messages d'erreurs de la PROM

```
>> boot bootp()/vmlinux
73264+592+11520+331680+27848d+3628+5792 entry: 0x8df9a960
Setting $netaddres to 192.168.1.5 (from server deadmoon)
Obtaining /vmlinux from server deadmoon

Cannot load bootp()/vmlinux
Illegal f_magic number 0x7f45, expected MIPSELMAGIC or MIPSEBMAGIC.
```

Ce problème survient uniquement pour des Indys avec des versions très anciennes de la PROM qui ne peuvent pas gérer le format binaire ELF que Linux utilise. Une solution à ce problème est en cours de résolution.

6.4 Où puis-je obtenir le microcode en "little endian" pour mon SNI ?

Le système SNI peut tourner à la fois en modes "big endian" et "little endian". Actuellement, Linux/MIPS supporte uniquement le microcode "little endian". C'est plutôt malchanceux car SNI n'a plus mis en place ce microcode depuis un certain temps, depuis qu'ils sont passés en NT. Lorsqu'il tourne en mode "big endian", le microcode se comporte de façon similaire à un Indy SGI qui est déjà supporté, par conséquent la résolution du support des SNI sera relativement facile. Les hackers intéressés peuvent contacter *Ralf Bächle (ralf@gnu.org)* .

6.5 ld meurt avec un signal 6

```
collect2: ld terminated with signal 6 [Aborted]
```

C'est un problème connu dans de vieilles versions de binutils. Vous devrez remettre à niveau binutils 2.8.1 plus des patches très récentes.

6.6 Ma machine ne télécharge pas mon noyau lorsque j'essaye de booter par le réseau

Votre machine répond aux paquets de BOOTP (vous devriez vérifier cela en utilisant des sniffers de paquets comme tcpdump ou ethereal) mais ne télécharge pas le noyau à partir du serveur BOOTP. Cela arrive si votre serveur de boot tourne sous un noyau de la série 2.3 ou supérieure. Le problème peut être contourné en faisant un "echo 1 > /proc/sys/net/ipv4/ip_no_pmtu_disc" sous compte administrateur (root) sur votre serveur de démarrage.

6.7 Erreur dans la version 2 de DHCP

Lors de l'utilisation de DHCP version 2, vous devrez voir apparaître le problème suivant : Votre machine a reçu sa réponse de BOOTP 3 fois mais refuse de démarrer TFTP. Vous pouvez résoudre cela en faisant un "unsetenv netaddr" dans la console de commande de la PROM avant de booter votre système. La version 3 de DHCP résoud ce problème.

7 Milo

Milo est le chargeur de démarrage utilisé pour démarrer les systèmes MIPS en mode "little endian" avec du microcode ARC, actuellement la famille Jazz et le SNI RM 200. Bien que Milo utilise le même nom et possède le même rôle que la version Alpha de Milo, ces deux Milo n'ont rien en commun. Ils sont développés par des gens différents, ne partagent pas de code, et travaillent sur des plateformes matérielles différentes. Le fait d'avoir le même nom est simplement une sorte d'"accident" historique.

Il est prévu de supprimer l'utilisation de Milo dans un futur proche.

7.1 Compiler Milo

La procédure de compilation de Milo est décrite en détail dans les fichiers README dans le paquetage de Milo. Puisque Milo dépend des fichiers d'en-tête du noyau qui ont changé plusieurs fois de suite, Milo, bien souvent, ne peut être compilé aisément ; cependant, la distribution de Milo inclus les binaires pour Milo et Pandora à la fois.

7.2 Pandora

Pandora est un simple débugeur. Il a été initialement dévelopé pour analyser les systèmes non-documentés. Pandora inclus un désassembleur, des fonctions de dump mémoire et bien plus. Si vous voulez simplement utiliser Linux, il est inutile d'installer Pandora. Cependant, il ne prend pas beaucoup de place.

8 Modules chargeables

L'utilisation des modules sous Linux/MIPS est plutôt facile ; cela devrait marcher normalement pour les personnes qui l'ont utilisé sous d'autres systèmes Linux. Si vous désirez lancer un système basé sur des modules alors il vous faut au moins un noyau de version 980919 et installer modutils avec un numéro de version plus récente que 2.1.121. Des versions plus anciennes ne fonctionneront pas.

9 Comment configurer un compilateur croisé ?

9.1 Binaires existant

La chose la plus simple pour configurer un compilateur croisé est de télécharger les binaires. Pour les machines Linux/i386 et les cibles "big endian", les paquetages nécessaires sont :

```
binutils-mips-linux-2.8.1-1.i386.rpm
egcs-c++-mips-linux-1.0.3a-1.i386.rpm
egcs-g77-mips-linux-1.0.3a-1.i386.rpm
egcs-libstdc++-mips-linux-2.8.0-1.i386.rpm
egcs-mips-linux-1.0.3a-1.i386.rpm
egcs-objc-mips-linux-1.0.3a-1.i386.rpm
```

Et voici la liste des paquetages pour les cibles "little endian" :

```
binutils-mipsel-linux-2.8.1-1.i386.rpm
egcs-c++-mipsel-linux-1.0.3a-1.i386.rpm
egcs-g77-mipsel-linux-1.0.3a-1.i386.rpm
egcs-libstdc++-mipsel-linux-2.8.0-1.i386.rpm
egcs-mipsel-linux-1.0.3a-1.i386.rpm
egcs-objc-mipsel-linux-1.0.3a-1.i386.rpm
```

Il n'est pas nécessaire de tous les installer ; la plupart des personnes peuvent oublier les compilateurs C++, Objective C et Fortran 77. Les binaires pour Intel ont été linké avec la GNU libc 2.1, donc vous devez l'installer aussi pour une mise à jour.

9.2 Construire votre propre compilateur croisé

Avant toute chose, téléchargez les paquetages des sources et les patches :

- binutils-2.8.1.tar.gz
- binutils-2.8.1-2.diff.gz
- egcs-1.0.3a.tar.gz
- egcs-1.0.3a-1.diff.gz
- glibc-2.0.6.tar.gz
- glibc-crypt-2.0.6.tar.gz

- glibc-localedata-2.0.6.tar.gz
- glibc-linuxthreads-2.0.6.tar.gz

Ce sont les versions actuellement recommandées. Des versions plus anciennes peuvent marcher comme elles ne peuvent pas marcher. Si vous essayez d'utiliser des versions plus anciennes, ne nous envoyez pas de rapport de bug s'il vous plaît ; nous n'y préterons pas attention. Lors de l'installation, veuillez procéder dans l'ordre binutils, egcs, puis glibc. A moins que vous ayez des versions plus anciennes déjà installées, le fait de changer l'ordre *fera* échouer l'installation. La description de l'installation suivante mentionne un certain nombre de patches que vous pouvez récupérer à partir des paquetages SRPM respectifs sur *oss.sgi.com* . Cependant, puisque ces paquetages SRPM sont censés être compilés dans la bonne architecture, il n'est pas possible de simplement les recompiler.

9.3 Espace disque nécessaire

Vous devez choisir un répertoire pour l'installation. Je ferais référence à ce répertoire par <prefix>. Pour éviter certain problème, il vaut mieux utiliser la même valeur pour <prefix> que pour le gcc natif. Par exemple, si votre gcc est installé dans /usr/bin/gcc alors choisissez /usr pour <prefix>. Vous devez utiliser la même valeur de <prefix> pour tous les paquetages que vous allez installer.

Pendant l'installation, il vous faudra environ 31 Mo d'espace disque pour binutils ; pour l'installation il vous faudra 7 Mo d'espace disque sur la partition contenant <prefix>. La compilation d'egcs nécesite 71 Mo et 14 Mo pour l'installation. GNU libc nécessite 149 Mo d'espace disque pendant la compilation et 33 Mo pour l'installation. Notez que ces quantités ne sont que des indications et peuvent varier significativement pour différentes architectures de processeurs et de systèmes d'exploitation.

9.4 Ordre des octets

Une des fonctionnalités particulières des architectures MIPS est que tous les processeurs sauf le R8000 peuvent être configuré pour tourner en mode gros ou petit boutien (big ou little endian). L'ordre des octets indique la manière dont le processeur stocke en mémoire les nombres sur plusieurs octets. Les machines gros boutiens stockent l'octet de poids le plus fort à l'adresse la plus basse alors que les machines petits boutiens les stockent à l'adresse la plus haute. Pensez à cela lors de l'écriture de nombres sur plusieurs octets de gauche à droite ou vice-versa.

Pour configurer correctement votre compilateur croisé, vous devez connaître l'ordre des octets du compilateur croisé cible. Si vous ne le savez pas déjà, consultez la section 3.1 (Hardware Platforms) pour l'ordre des octets des machines.

9.5 Les noms de configuration

La plupart des paquetages basés sur autoconf supportent plusieurs architectures et systèmes d'exploitation différents. Pour différencier chaque configuration, les noms sont construits selon les schémas <cpu>-<entreprise>-<SE> voire même <cpu>-<entreprise>-<noyau>-<SE>. Selon ces schémas, les noms de configuration de Linux/MIPS sont mips-unknown-linux-gnu pour les gros boutiens ou mipsel-unknown-linux-gnu pour les petits boutiens. Ces noms sont un peu long et il est permis de les abréger en mips-linux ou mipsel-linux. Vous *devez* utiliser les mêmes noms de configuration pour tous les paquetages qui comprennent votre environnement de compilateur croisé. Ainsi, bien que les autres noms comme mips-sni-linux ou mipsel-sni-linux sont les noms de configuration légaux, utilisez linux-mips ou mipsel-linux à la place ; ce sont les noms de configuration connu par les autres paquetages comme les sources du noyau Linux, autrement il devra être modifié pour les compilations croisées.

Je ferais maintenant référence au nom de configuration cible par <target>.

9.6 Installation de GNU binutils

Ceci est la première partie et la plus simple - du moins si vous essayez de faire l'installation sur un quelconque UNIX sain. Entrez simplement dans un répertoire avec suffisamment d'espace disque et faites ce qui suit :

```
gzip -cd binutils-<version>.tar.gz | tar xf -
cd binutils-<version>
patch -p1 < ../binutils-<version>-mips.patch
./configure --prefix=<prefix> --target=<target>
make CFLAGS=-O2
make install
```

Cela fonctionne très bien habituellement. Sur certaines machines qui utilise GCC 2.7.x comme compilateur, on raporte des dump core. C'est un bug connu de GCC et peut être résolu en mettant GCC à niveau à la version 2.8.1 ou en mettant à niveau egcs.

9.7 Assert.h

Des personnes possèdent un vieux fichier d'en-tête assert.h installé, probablement des restes d'une ancienne installation d'un compilateur croisé. Ce fichier a pour conséquence de faire échouer silencieuement les scripts autoconf ; il n'est jamais nécessaire et il a été uniquement installé à cause d'un bug dans des versions plus anciennes de GCC. Vérifiez si le fichier <prefix>/<target>/include/assert.h existe dans votre installation. Si c'est le cas, effacez-le : il n'aurait jamais dû être installé pour une quelconque version du compilateur croisé et causera des perturbations.

9.8 Installation des sources du noyau

L'installation des sources du noyau est simple. Placez-les simplement dans un répertoire de votre choix et configurez-les. Leur configuration est nécessaire pour que les fichiers qui sont générés par la procédure soient installés. Assurez-vous que vous ayez activé CONFIG_CROSSCOMPILE vers la fin du processus de configuration. L'unique problème que vous pourriez rencontrer est d'avoir besoin d'installer des programmes GNU nécessaires comme bash ou devoir écraser les versions de programmes fournies par le constructeur en plaçant les versions GNU plus tôt dans la variables PATH.

Maintenant, allez dans le répertoire <prefix>/<target>/include et créez deux liens symboliques nommés asm et linux pointant respectivement sur include/asm et include/linux dans vos sources du noyau qui viennent juste d'être installés et configurés. Ils sont nécessaires pour que les fichiers d'en-têtes nécessaires soient trouvés pendant l'étape suivante.

9.9 Première installation d'egcs

Maintenant la partie la moins rigolote commence : il existe un soit-disant problème d'amorçage. Dans notre cas, cela signifie que le processus d'installation d'egcs nécessite une glibc précédemment installée, mais nous ne pouvons pas compiler glibc parce que nous ne possédons pas encore de compilateur croisé. Heureusement, vous devrez uniquement passer par là lorsque vous installerez un compilateur croisé pour la première fois. Plus tard, lorsque vous aurez déjà installé la glibc les choses seront plus aisées. Mais pour l'instant faites :

```
gzip -cd egcs-1.0.3a.tar.gz | tar xf -
cd egcs-<version>
patch -p1 < ../egcs-1.0.3a-mips.patch
./configure --prefix=<prefix> --with-newlib --target=<target>
make SUBDIRS="libiberty texinfo gcc" ALL_TARGET_MODULES= \
        CONFIGURE_TARGET_MODULES= INSTALL_TARGET_MODULES= LANGUAGES="c"
```

Notez que nous n'avons pas délibérement compilé gcov, protoize, unprotoize et les bibliothèques. Gcov n'a pas de sens dans un environnement de compilateur croisé et protoize et unprotoize écraserait carrément vos programmes natifs - c'est un bug dans les makefiles de gcc. Enfin, nous ne pouvons compiler les bibliothèques parce que nous n'avons pas encore installé la glibc. Si tout ce passe avec succés, lancez l'installation avec :

```
make SUBDIRS="libiberty texinfo gcc" INSTALL_TARGET_MODULES= \
        LANGUAGES="c" install
```

Si vous désirez le compilateur croisé pour compiler le noyau, vous avez finis. La compilation croisée de la libc est maintenant nécessaire pour pouvoir compiler les applications utilisateurs.

9.10 Tester ce que vous avez fait jusqu'ici

Simplement pour s'assurer que ce que vous avez fait jusqu'ici fonctionne, vous devriez maintenant essayer de compiler le noyau. Entrez dans les sources du noyau MIPS et tapez "make clean; make dep; make". Si tout se déroule bien, faites un "make clean" une fois de plus pour nettoyer les sources.

9.11 L'installation de la GNU libc

Faites :

```
gzip -cd glibc-2.0.6.tar.gz | tar xf -
cd glibc-2.0.6
gzip -cd glibc-crypt-2.0.6.tar.gz | tar xf -
gzip -cd glibc-localedata-2.0.6.tar.gz | tar xf -
gzip -cd glibc-linuxthreads-2.0.6.tar.gz | tar xf -
patch -p1 < ../glibc-2.0.6-mips.patch
mkdir build
cd build
CC=<target>-gcc BUILD_CC=gcc AR=<target>-ar RANLIB=<target>-ranlib \
      ../configure --prefix=/usr --host=<target> \
      --enable-add-ons=crypt,linuxthreads,localedata --enable-profile
make
```

Vous avez maintenant une GNU libc compilée qui doit encore être installée. *Ne* faites *pas* un simple make install. Cela écraserait vos fichiers système de votre machine par les fichiers spécifiques de Linux/MIPS avec des effets désastreux. A la place, installez la GNU libc dans un autre répertoire choisie arbitrairement <somedir> à partir du quel nous déplacerons dans le répertoire cible actuel les parties que nous avons besoin pour la compilation croisée :

```
make install_root=<somedir> install
```

Maintenant pénétrez dans <somedir> et installez finalement la GNU libc manuellement

```
cd usr/include
find . -print | cpio -pumd <prefix>/<target>/include
cd ../../lib
find . -print | cpio -pumd <prefix>/<target>/lib
cd ../usr/lib
find . -print | cpio -pumd <prefix>/<target>/lib
```

La GNU libc contient aussi une vaste documentation en ligne. Votre système doit déjà posséder une version de cette documentation, donc si vous ne désirez pas installer les pages infos ce qui vous sauvera un peu moins d'un mega octets, ou si vous les avez déjà installés, sautez la prochaine étape :

```
cd ../info
gzip -9 *.info*
find . -name \*.info\* -print | cpio -pumd <prefix>/info
```

Si vous n'avez pas de programme d'amorce, votre installation n'est pas finis

9.12 Recompilation d'egcs

La première tentative de compilation d'egcs a été stoppée par l'absence de la GNU libc. Puisque nous avons maintenant la libc d'installée, nous pouvons reconstruire egcs mais cette fois de façon aussi complète que l'installation d'un compilateur croisé puisse l'être :

```
gzip -cd egcs-<version>.tar.gz | tar xf -
cd egcs-<version>
patch -p1 < ../egcs-1.0.3a-mips.patch
./configure --prefix=<prefix> --target=<target>
make LANGUAGES="c c++ objective-c f77"
```

Comme vous pouvez le constater, la procèdure est identique que la première fois sauf que nous avons laissé tombé l'option –with-newlib. Cette option était nécessaire pour enlever l'arrêt de la compilation de libgcc à cause de l'absence de la libc. Maintenant lancez l'installation avec :

```
make LANGUAGES="c c++ objective-c f77" install
```

Vous avez presque terminé. Si vous pensez ne pas avoir besoin des compilateurs Objective C ou F77, vous pouvez les enlever des commandes ci-dessus ; chacune vous sauvera environ 3 Mo. Ne compilez pas gcov, protoize ou unprotoize.

9.13 Dois-je compiler les compilateurs C++, Objective C ou F77 ? La réponse à cette question dépend largement de l'utilisation que vous ferez de votre environnement de compilateur croisé. Si vous avez pour unique but de recompiler le noyau Linux alors vous n'avez pas besoin d'une configuration pleine à craquer et vous pouvez ommettre en toute sécurité les compilateurs Objective C et F77. Vous devez, cependant, compiler le compilateur C++, parce que la compilation des bibliothèques inclues avec la distribution d'egcs nécessite C++.

9.14 Problème connu lors de compilation croisée

9.14.1 Plantage d'IRIX

Origin 200 tournant sous IRIX 6.5.1 peut se planter lors d'un "make depend" dans les sources de noyau Linux. IRIX 6.5 sur un Indy et IRIX 6.5.4 sur un Origin 200 sont connus pour fonctionner correctement. Tout rapports pour aider à isoler le problème de configuration est la bienvenue.

9.14.2 Ressources limitées sur les machines basées sur le standard System V

Les Unix basés sur le standard System V comme IRIX ou Solaris ont des valeurs limites maximum sur les nombres d'argument pouvant être passés à un processus enfant qui peuvent être dépassés lors de la compilation croisée de logiciels comme le noyau Linux ou la GNU libc. Pour les systèmes IRIX, la taille maximum de la liste d'arguments vaut par défaut 20 Ko alors la valeur par défaut pour Linux est d'au moins 128 Ko. Cette taille peut être modifié par la commande "systune ncargs 131072" sous root.

9.15 GDB

La compilation de GDB comme debugeur croisé a uniquement d'intérêt pour les développeurs du noyau ; leur GDB peut leur sauver la vie. Une telle configuration de debugage distant est composée de deux parties : le debugeur GDB distant fonctionnant sur une machine et la machine cible faisant tourner le noyau Linux/MIPS qui doit être débugué. Les machines sont habituellement inter-connectées par une ligne série. Le noyau de la machine cible doit être équipé d'un "embryon de débugage" qui communique avec le GDB de la machine hôte en utilisant le protocole série distant.

Suivant l'architecture cible, il vous faudra implémenter l'embryon de débugage vous-même. En général, vous devrez uniquement écrire des routines séries très simple. La tâche est, de plus, simplifiée par le fait que la plupart des machines utilisent des périphériques séries semblables, en général basé sur le 8250, le 16450 ou des dérivés.

10 Litteratures sur le même thème

10.1 Voir un MIPS tourner (See MIPS Run)

auteur Dominic Sweetman, publié par Morgan Kaufmann, ISBN 1-55860-410-3. Il a été écrit pour être un guide plutôt compréhensif pour la programmation MIPS, là où celui-ci diffère de la programmation d'autres CPU 32 bits. C'est la première fois que quelqu'un tente d'écrire une explication lisible et compréhensive et qui prend en compte l'immense gamme de CPU MIPS existant, et devrait être très utile pour toutes personnes programmant en MIPS qui n'est pas isolé par le système d'exploitation de quelqu'un d'autre. Et l'auteur est un enthousiaste pro-unix-libre qui s'est inscrit à la liste de discussions de Linux/MIPS ! John Hennessey, le père de l'architecture MIPS, a été assez aimable pour écrire dans la préface " ... ce livre est la meilleure combinaison entre l'exhaustivité et la lisibilité de tout livre sur l'architecture MIPS ... " ; Il contient une partie sur les CPU RISC, une description de l'architecture et de l'ensemble d'instructions incluant les instructions du "co-processeur 0" utilisé pour le contrôle du CPU ; des sections sur les caches, les exceptions, la gestion mémoire et les virgules flottantes. Il y a un guide détaillé sur le langage assembleur, des trucs sur le portage et des exemples de logiciels plutôt à usage industriel. Il est disponible sur :

- `http://www.algor.co.uk/algor/info/seemipsrun.html` (europe)
- `http://www.mkp.com/books_catalog/1-55860-410-3.asp` (US)

et chez tous les bonnes librairies. Il contient 512 pages et coûte environ 50 $ aux Etats-Unis et 39.95 £ au Royaume-Unis. J'aimerais parler de deux autres livres aussi, publiés chez Morgan Kaufmann et disponible sur www.mkp.com ou chez n'importe quelle librairie :

10.2 Le manuel du programmeur MIPS (The MIPS Programmer's Handbook)

auteurs Farquhar et Bunce, publié par Morgan Kaufmann, ISBN 1-55860-297-6. Une introduction lisible pour la pratique de la programmation bas niveau d'un MIPS, par l'auteur de PMON. Sa qualité : beaucoup d'exemples ; son défaut : ne parle pas d'importantes parties de l'architecture (telle que la gestion de la mémoire, des virgules flottantes et les caches avancés) parce qu'ils n'ont pas été présent dans le LSI "embarqués" ; ce livre a l'intention d'être un partenaire.

10.3 Architecture informatique - Une approche quantitative (Computer Architecture - A Quantitative Approach)

auteurs Hennessy & Patterson, publié chez Morgan Kaufmann, ISBN 1-55860-329-8. La bible de l'architecture informatique moderne et qui se doit d'être lu si vous désirez comprendre pourquoi un programme fonctionne lentement ou rapidement. Parle-t-il du MIPS ? Eh bien, cela tourne principalement autours de chose *semblable* au MIPS ... Son seul défaut est sa taille et son poids - mais à la différence des gros bouquins, il le vaut à chaque page.

10.4 Supplément pour les processeur MIPS ABI sous UNIX System V (UNIX System V ABI MIPS Processor Supplement)

par Prentice Hall, publié le 05 1991, ISBN 0-13880-170-3. Ce livre définit la plupart des standards techniques spécifiques à la technologie MIPS comme les conventions des appels, les propriétés ELF et bien d'autre choses utilisées par Linux/MIPS. Malheureusement, il est en rupture de stock. Parallèlement, le site *"http://www.mipsabi.org/"* ne répond plus.