# Le problème du signal 11

Rogier Wolff ( *R.E.Wolff@BitWizard.nl* `mailto:R.E.Wolff@BitWizard.nl` )
(traduit par Nat Makarévitch *nat@linux-france.com* `mailto:nat@linux-france.com` ) Version 19970716fr

Cette FAQ dresse liste des causes possibles d'un problème que connaissent ces derniers temps certains utilisateurs de Linux : la compilation d'un important ensemble logiciel[1] échoue à cause d'un *signal 11*. Ce problème peut être causé par un dysfonctionnement d'ordre matériel (cas le plus fréquemment observé) ou logiciel.

## Contents



## 1 À propos de ce document

La *version originale* `http://www.bitwizard.nl/sig11/` de ce document est à présent intégrée à la collection des "Mini-Howto" de Linux.

[1](par exemple le noyau)

La version publiée sur le Web était consultée 300 fois par semaine en juin 1996 (augmentation : facteur 3 en 3 mois).

La plus récente version de ce texte, librement utilisable s'il n'est pas modifié, sur trouve sur

*son site de référence <URL:http://www.linux-france.com/>* `http://www.linux-france.com/`

Tout commentaire et compte-rendu d'expérience intéresse l'auteur ( *Rogier Wolff <R.E.Wolff@BitWizard.nl>* `mailto:R.E.Wolff@BitWizard.nl` ) mais les suggestions d'ajouts techniquement sans valeur seront rejetées. Expédier à *nat@linux-france.com* `mailto:nat@linux-france.com` les commentaires concernant cette adaptation française.

Note : le problème détaillé ici concerne aussi les autres systèmes un tant soit peu exigeants : Windows 3.1, FreeBSD, Windows NT, NextStep...

Cette adaptation française doit beaucoup à J. Chion.

# 2 Question clé

Voici la question-clé traitée par ce document :

```
Lorsque je compile un noyau Linux la procédure avorte avec un message:
      gcc: Internal compiler error: program cc1 got fatal signal 11
Que se passe-t-il ?
```

## 2.1 Réponse succincte

Le problème est vraisemblablement causé par un dysfonctionnement du matériel. De nombreux composants de l'ordinateur peuvent être impliqués et diverses manières de résoudre le problème existent.

# 3 Comment s'assurer que le matériel est en cause ?

Sitôt après l'échec du `make`, invoquez-le à nouveau.

Si la machine parvient à compiler quelques autres fichiers, nous pouvons penser que le matériel est défaillant.

Si, par contre, la compilation cesse tout de suite (message "nothing to be done for xxxx" avant nouvel échec au même endroit), il faudra déterminer si le contenu de la mémoire vive est toujours bien préservé. Pour cela :

```
        dd if=/dev/DISQUE_DUR of=/dev/null bs=1024k count=MEGAS
```

*DISQUE_DUR* remplace ici le nom du fichier spécial associé au disque dur stockant les sources. Pour connaître son nom, rester dans le répertoire abritant les sources et introduire `df .` ("df" suivi d'un espace puis un point).
*MEGAS* remplace ici le nombre de Mo de mémoire vive dont la machine dispose (indiqué par `free`).

Cette commande va obliger Linux à lire les informations placées au début du disque de façon à "gaver" le contenu du cache disque ("buffer-cache"). Il devra donc, par la suite, relire les fichiers source à compiler ainsi que les binaires de gcc.

Invoquer `make`.

Si la compilation échoue toujours au même "endroit", le problème est probablement d'ordre logiciel. Étudier en ce cas la section consacrée aux 13 (autres causes possibles).

Si la compilation échoue à un autre stade, nous pouvons conclure que les transferts de données entre le disque et la mémoire vive ne sont pas assurés correctement.

# 4 Que signifie ce "signal 11" ?

Linux avorte grâce au "signal 11" tout programme tentant d'accéder à une adresse mémoire ne lui appartenant pas. Parmi les nombreuses causes possibles, nous ne pouvons retenir dans l'absolu que deux possibilités, dans le cas où cela concerne une version stable de gcc utilisée sur une machine très commune : problème matériel ou bien inadéquation de certaines composantes des utilitaires logiciels du système.

Lorsque ce problème survient sur une machine sans défaut matériel, il ne peut être causé que par une erreur de programmation ou de compilation (en l'occurrence du binaire de gcc). Mais lorsque le matériel est défaillant, et que des valeurs stockées en mémoire vive changent plus ou moins aléatoirement, un programme exigeant tel que gcc ne parviendra pas à mener à bien sa mission car il tentera tôt ou tard de déréférencer un pointeur au contenu ainsi modifié.

Un pointeur, sur une machine à processeur Intel, s'étend sur 32 bits et permet donc d'accéder à 4 Go. Peu de machines Intel disposent d'autant de mémoire vive dont la majeure partie serait allouée à gcc !
Une adresse de 32 bits aléatoire est donc très probablement illégale et Linux tuera le programme qui tente avec elle, selon lui, de manipuler des données ne lui appartenant pas.

# 5 Quel composant incriminer ?

Voici une liste des diverses causes de dysfonctionnement du matériel :

- Composants trop lents. De mauvaises surprises sont à craindre si l'une des "barrettes" de mémoire vive ne fonctionne pas convenablement. Même une carte mère capable de contrôler par test de parité ne détectera pas les erreurs survenant lors des cycles d'écriture.Inventaire des causes et solutions :
  - Latence des composants trop importante ("mémoire trop lente")
    Le contrôleur de bus ne parvient pas toujours en ce cas à obtenir à temps la donnée requise par le processeur car la mémoire "réagit" trop lentement. Solution :augmenter le nombre de cycles d'attente ("wait states") grâce au SETUP de la machine. Problème fréquent sur les machines 486 cadencé à plus de 80 MHz équipées d'un BIOS de marque AMI. (Pat V.)
    Il est parfois nécessaire de remplacer les composants pour diminuer la latence. Les systèmes ayant un bus cadencé à 33 MHz (P100, P133...) ne doivent pas employer de RAM avec plus de 60ns de latence, surtout si la carte mère est de marque ASUS. L'ensemble peut sembler fonctionner avec des composants à 70ns mais une petite erreur est alors toujours possible (Andrew Eskilsson).
  - Composant défecteux
    Démonter une barrette (ou changer temporairement la seule barrette employée) puis relancer le système et tester. Recommencer autant de fois que nécessaire afin d'isoler le (ou les !) composants défectueux. Prendre garde, le cas échéant, lors de la manipulation des mémoires statiques, car une décharge *d'électricité statique* peut les *condamner*.

    Témoignage (kettner@cat.et.tudelft.nl) : nous avons éprouvé de grandes difficultés avec une machine dont il s'avéra que les quatre barrettes étaient défectueuses et modifiaient à peu près un bit par heure de fonctionnement. La machine "plantait" environ une fois par jour et les compilations de noyau échouaient environ une fois par heure. Cette machine a pu exécuter le test mémoire durant 2300 cycles complets sans erreur, puis détecta environ 10 erreurs et continua ensuite sans problème durant plusieurs centaines de cycles. La compilation de noyau s'avéra le test le plus efficace car même le cas le plus favorable ne permettait pas de compiler plus de 14 noyaux à la suite. Nous avons donc échangé ces barrettes.

- Convertisseurs défectueux

  De nombreux supports de mémoire, permettant de monter des composants 32 bits sur des supports 72 points, ne sont pas conçus de façon correcte, en particulier les plus anciens.Témoignages : nous avons très longtemps utilisé sans problème un jeu de composants sans support de ce type. Mais ils ne furent pas utilisables avec un convertisseur (Naresh Sharma (n.sharma@is.twi.tudelft.nl)).

  Paul Gortmaker (paul.gortmaker@anu.edu.au) indique que les convertisseurs doivent tous comporter au moins quatre condensateurs de régulation du courant.

- Mode de rafraîchissement de la mémoire vive inadéquat

  Les composants "perdent" alors peu à peu les données stockées. Causes (Hank Barta (hank@pswin.chi.il.us), Ron Tapia (tapia@nmia.com)) : certaines cartes mère donnent la possiblité de raréfier les cycles de rafraîchissement en vue d'augmenter la bande passante utile du bus (option "hidden refresh" du SETUP). Un programme, souvent appelé `dram`, offre le moyen de configurer le jeu de composants ("chipset") au plus bas niveau afin d'obtenir des effets semblables.

- Trop faible nombre de cycles d'attente

  Certains composants de la carte mère peuvent ne pas fonctionner toujours correctement si le nombre de cycles d'attente ("wait states") n'est pas approprié. L'augmenter grâce au SETUP.

• Défaillance de la mémoire cache

Le contenu de la mémoire cache n'est généralement pas certifié par un test de parité et une défaillance ne sera donc pas diagnostiquée par la carte mère. Test : utiliser le SETUP pour invalider le cache externe ("L2") puis faire fonctionner le système. Si les problèmes disparaissent le cache est défectueux. Solutions :

- Vitesse ou latence de la mémoire cache inadéquate.

  Augmenter, grâce au SETUP, le nombre de cycles d'attente.

- Composant de mémoire défecteux

  Il faut alors changer de composant cache. *ATTENTION* : il s'agit très souvent de mémoire statique, donc *très* fragile (Joseph Barone (barone@mntr02.psf.ge.com)).

- Mode d'exploitation du cache inadéquat

  Le mode "écriture différée" ("write back"), par exemple, cause parfois des problèmes lorsque le jeu de composants de la carte mère n'est pas correctement conçu (cas observé sur une carte "MV020 486VL3H" (20 Mo RAM) par Scott Brumbaugh).

• Configuration incorrecte de la carte mère

Un cavalier ("jumper") détermine parfois le cache qui sera employé (le modèle monté sur une micro carte d'extension ou bien les composants de mémoire classiques). Exemple : cavalier JP16 sur les ASUS P/I-P55TP4XE version 2.4.

• Transferts de données entre disque et mémoire

Un bloc de données lu sur le disque peut se trouver stocké en mémoire avec un bit erroné.
Déterminer si c'est la cause du problème en recopiant des fichiers puis en comparant la copie à l'original.
Répéter ce test : après un `dd` (consulter à ce propos la section consacrée à l'3 (expiration du buffer cache)) la compilation avortera très vraisemblablement à un autre stade.

- Interruptions masquées durant des transferts IDE

  Certains disques IDE ne tolèrent pas le démasquage des interruptions lors des transferts, en particulier en période de forte charge système ("hdparm -u0").

  - Disque de marque Kalok
    La qualité des disques Kalok de la série 31xx laisse beaucoup à désirer, mieux vaut éviter de les employer. Ils ne sont de toutes façons pas compatibles avec Linux. Les réformer ou laisser aux utilisateurs de systèmes d'exploitation sans cache disque.
  - Disques SCSI
    Vérifier terminateurs et câbles. Un câble court peut sembler fonctionner avec une terminaison inadéquate mais les données transférées peuvent en pâtir. Essayer de valider les options de test de parité.

- Augmentation abusive de la cadence d'horloge ("overclocking")
  Le résultat est le plus souvent aléatoire. Essayer d'exploiter la machine à la cadence d'horloge normale.

  Dans un cas au moins (Samuel Ramac (sramac@vnet.ibm.com)) un processeur P120 ne tolérait pas 120 MHz mais fonctionnait à 100 MHz. La carte mère n'était pas en cause car le bus est en fait plus rapide lorsque l'horloge bat à 100 MHz[1]. Un autre processeur P120, monté en lieu et place, fonctionne d'ailleurs normalement.
  Tous les "fondeurs" (constructeurs de processeurs) produisent ainsi de rares "ratés", ce n'est en rien spécifique à Intel.

- Refroidissement du processeur
  L'élevation de la température du processeur provoquée par une augmentation de la cadence d'horloge ou par une panne du dispositif de refroidissement peut générer des dysfonctionnements. Bon révélateur : interdire au noyau d'utiliser l'instruction HALT grâce au paramètre LILO adéquat (lire le "BootPrompt HOWTO"). La température du circuit augmentera alors beaucoup plus vite, même sous faible charge système, et la fréquence d'apparition des problèmes augmentera. Le Pentium à l'instruction "FDIV" boguée est particulièrement concerné car son ventilateur n'était pas conçu au mieux. Notons aussi que la colle employée pour assujetir le radiateur au processeur doit présenter des caractéristiques de conduction thermique correcte (Arno Griffioen (arno@ixe.net), W. Paul Mills (wpmills@midusa.net), Alan Wind (wind@imada.ou.dk))

  Intel indique que la température de la surface du processeur doit être comprise entre :

  - 0 et +85° C: Intel486 SX, Intel486 DX, IntelDX2, IntelDX4
  - 0 et +95° C: IntelDX2, IntelDX4 OverDrive®
  - 0 et +80° C: 60 MHz Pentium®
  - 0 et +70° C: 66 to 166 MHz Pentium

  Consulter à ce propos les sections Q6, Q7 et Q13 de ce

  *document Intel* `http://pentium.intel.com/procs/support/faqs/iarcfaq.htm`

- Voltage de l'alimentation du processeur
  Certains processeurs 5 Volts fonctionnent sous 3,3 Volts, mais pas toujours de façon parfaite. Pis : les documentations de certains systèmes sont incorrectes et recommandent une configuration inadéquate (Karl Heyes (krheyes@comp.brad.ac.uk))

[1]CPU à 120 : bus à 60 (x 2), CPU à 100 : bus à 66 (x 1,33)

- Voltage de l'alimentation de la mémoire
  Les plus récentes cartes ne tolèrent que la mémoire 3,3 Volts. Ne jamais utiliser les composants sous un voltage inadéquat (risque de destruction).

- Surexploitation du bus local
  Le nombre de cartes connectables à un bus local décroît avec sa fréquence d'exploitation : 3 cartes à 25 MHz, 2 à 33 MHz, une seule à 40 MHz et aucune à 50 MHz (fréquence maximale). Certains systèmes tolèrent mal la surcharge et les données échangées peuvent alors en pâtir. Essayer d'augmenter les états d'attente insérés entre les cycles du bus local (Richard Postgate (postgate@cafe.net)).

- Fonctions d'économie d'énergie ("power management", "APM")
  Certains portables, en particulier, offrent une fonction de reprise immédiate (mode "resume") et des programmes pilotes ne tolèrent pas toujours cela. Débrayer ces fonctions ou bien compiler l'"APM support" dans le noyau (Elizabeth Ayer (eca23@cam.ac.uk)).

- Processeur défectueux
  Certains exemplaires des processeurs courants recèlent des bogues aux effets pervers. Aucune solution n'existe, il faut remplacer le composant. Des cas d'incompatibilité entre processeur et carte mère auraient été observés. Depuis février 1997 la première vague de problèmes, qui concernait les processeurs Intel, décroît nettement tandis que l'exploitation de processeurs Cyrix/IBM 6x86 sur certaines cartes mère s'avère difficile. Le manuel d'une carte mère précise qu'elle est incompatible avec les premières versions du 6x86. C'est regrettable car les performances de ces processeurs sont fort bonnes.

# 6 Mais tout fonctionnait correctement depuis longtemps !

Le fait que la configuration déficiente fonctionnait sans problème depuis un moment n'implique malheureusement pas que le matériel est hors de cause.

L'exemple classique concerne les composants de mémoire. Leurs fabricants ne disposent pas d'une ligne de production distincte pour chaque type de mémoire. Les circuits proviennent tous des mêmes machines et matières premières, seul le test final détermine si un composant donné sera par exemple vendu en tant que 60 ns ou bien 70 ns. Vos composants fonctionnaient peut-être à merveille depuis longtemps à la limite de leurs capacités mais un facteur quelconque (la température, par exemple, ralentit les mémoires) peut les rendre assez vite inadéquats.

Un climat estival ou bien une lourde charge de travail processeur place donc parfois le système dans des conditions où son fonctionnement correct n'est plus certain, voire plus possible (Philippe Troin (ptroin@compass-da.com)).

# 7 Mon programme de test mémoire ne révèle aucun problème

Le test mémoire effectué par le BIOS lors du démarrage de la machine n'en est le plus souvent pas un. Des conditions d'exploitation extrêmes peuvent seules permettre de lever le doute. Tester grâce à `memtest86`.

# 8 Le problème est-il limité à la compilation du noyau ?

Non, mais la compilation du noyau exige beaucoup de ressources et constitue donc un excellent test ou révélateur.

Autres cas observés :

- Certaines machines se bloquent parfois lorsqu'elles exécutent le script d'installation de la distribution Slackware (dhn@pluto.njcc.com).
- Le noyau stoppe parfois une tâche à cause d'une "general protection error" (message confié à syslog) (fox@graphics.cs.nyu.edu).

# 9 D'autres systèmes d'exploitation fonctionnent pourtant bien sur cette machine !

Linux exploite mieux le matériel que la plupart des autres systèmes, comme ses performances le laissent imaginer.

Certains autres systèmes, par exemple édités par Microsoft, se "plantent parfois" de façon incompréhensible. Peu d'utilisateurs s'en plaignent, semble-t-il, et cette société leur *répond* `http://www.cantrip.org/nobugs.html` en ce cas d'une manière quelque peu étrange.

Le mode de conception et d'utilisation de ces systèmes d'exploitation produit un ensemble le plus souvent plus "prédictible" que Linux dans la mesure ou une application donnée sera le plus souvent chargée dans la même section de la mémoire vive. Les aléas dûs à un composant défectueux sont donc parfois portés au compte d'un programme donné et non du matériel.

Une chose demeure cependant certaine : un système Linux bien installé sur une machine saine doit pouvoir compiler cent fois de suite un noyau sans aucun problème.

Témoignage : Linux et gcc testent à merveille la machine. Hors de Linux le test "Winstone" produit le même genre d'effets (Jonathan Bright (bright@informix.com))

# 10 "signal 11" est-il le seul effet ?

Ce n'est malheureusement pas le cas. Les signaux 6 et 4 peuvent aussi relever de ce genre de problème (lorsque la mémoire n'accomplit pas correctement son office n'importe quel type d'erreur peut survenir) mais le 11 est le plus commun.

Autres problèmes constatés :

- free_one_pmd: bad directory entry 00000008
- EXT2-fs warning (device X:Y): ext_2_free_blocks bit already cleared for block Z
- Internal error: bad swap device
- Trying to free nonexistent swap-page
- kfree of non-kmalloced memory ...
- scsi0: REQ before WAIT DISCONNECT IID
- Unable to handle kernel NULL pointer dereference at virtual address c0000004
- put_page: page already exists 00000046 invalid operand: 0000
- Whee.. inode changed from under us. Tell Linus
- crc error System halted (lors du démarrage)
- Segmentation fault

- "unable to resolve symbol"
- make 1] : $***...Error139make : ***...Error1XWindowavortebrusquementavecunmesage'canterminatewitha" cau$

Les premiers exemples relèvent d'arrêts provoqués par le noyau qui "suspecte" une erreur de programmation l'affectant. Les autres concernent les applications.

(S.G.de Marinis (trance@interseg.it), Dirk Nachtmann (nachtman@kogs.informatik.uni-hamburg.de))

## 11 Que faire ?

- Démonter des barrettes, les remplacer
- Débrayer (SETUP) le cache de second niveau du processeur
- Diminuer la cadence du processeur et du bus
- Restaurer la cadence de rafraîchissement de la mémoire préconisée
- Démarrer avec un noyau sous option "mem=4M" pour lui interdire d'exploiter la mémoire au-dessus des 4 premiers Mo
- Tester :

  ```
  tcsh
  cd /usr/src/linux
  make zImage
  foreach i (0 1 2 3 4 5 6 7 8 9)
    foreach j (0 1 2 3 4 5 6 7 8 9)
      make clean;make zImage > log."$i"$j
    end
  end
  ```

  Tous les contenus des fichiers de trace résultants doivent être identiques. Cela exige environ 24 heures sur un P100 / 16 Mo RAM et environ 3 mois sur un 386 / 4 Mo :-)

Le moyen le plus efficace reste de remplacer tous les composants de mémoire. Ce n'est cependant pas toujours facile.

## 12 Et le test physique ?

Même certains équipements électroniques de test des mémoires ne mettent pas toujours en évidence les problèmes dont nous traitons ici car ils peuvent par exemple dépendre du mode d'exploitation des composants par la carte mère.

## 13 Quelles sont les autres causes possibles ?

- pgcc
  Utilisation de la version de gcc "pgcc", dont le générateur de code est optimisé pour le Pentium. La compilation, avec ses options par défaut, de certains modules du noyau (par exemple floppy.c) produit un signal 11. Les causes se trouvent à la fois dans le noyau, la libc et pgcc. On constate vite qu'il ne s'agit pas d'un problème matériel car il se produit toujours au même stade de la compilation.

Solution : utiliser un gcc standard ou bien des options interdisant certaines optimisations (par exemple "-fno-unroll-loops") (Evan Cheng (evan@top.cis.syr.edu)).

- Composants de gcc hétéroclites
  Lorsque les fichiers appartenant à gcc proviennent de sources différentes des problèmes peuvent apparaître. Il faut alors tout remplacer par une version complète et correcte (Richard H. Derr III (rhd@Mars.mcs.com)).

- Édition de liens avec bibliothèque pour `SCO`
  Sous iBCS les applications dont le LDFLAGS contient -Llib/ sont exposées.

- a.out et ELF
  Compilation d'un noyau a.out au sein d'un environnement ELF (ou le contraire).. Le premier appel à "ld" causera toujours un "signal 11"(REW).

- Carte Ethernet ISA sur bus PCI mal configuré
  Cela peut causer de graves problèmes logiciels (sigsegv, arrêt du noyau...). Il faut alors utiliser le SETUP pour configurer l'"aperture" (zone de mémoire commune à la carte et à l'espace d'adressage du système).

- Contenu de la partition de mémoire virtuelle ("swap") endommagé
  Tony Nugent (T.Nugent@sct.gu.edu.au) précise qu'il a pu résoudre le problème en re-préparant la partition grâce à "mkswap".
  Louis J. LaBash Jr. (lou@minuet.siue.edu) nous rappelle qu'il faut invoquer "sync" après un "mkswap".

- Cartes Ethernet bas de gamme de type NE2000
  La qualité de certaines cartes est si médiocre qu'elles mettent en péril la stabilité du système. Les noyaux Linux postérieurs à 1.3.48 les tolèrent semble-t-il mieux (REW).

- Alimentation électrique
  Cas peu probable, même une machine très bien équipée n'approche guère les limites des alimentations 200 W. Seul un système utilisant de nombreux anciens disques (gros consommateurs de courant) peut poser un problème (Greg Nicholson (greg@job.cba.ua.edu)).
  Thorsten Kuehnemann (thorsten@actis.de) indique qu'une alimentation défectueuse peut provoquer des signaux 11.

- Compilation du code ext2
  Dans certains cas la compilation du code de gestion du système de fichiers ext2 provoque un signal 11 (Morten Welinder (terra@diku.dk)).

- Mémoire disponible insuffisante
  gcc produit alors d'étranges erreurs (Paul Brannan (brannanp@musc.edu)).

## 14 Cette liste me laisse sceptique !

Nous ne traitons ici que de cas **réels !**
(N.d.T : la version *originale* `http://www.bitwizard.nl/sig11/` de ce document propose une liste des auteurs de témoignages).