



- [Accueil](#)
- [À propos](#)
- [Nuage de Tags](#)
- [Contribuer](#)
- [Who's who](#)

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source

19 août 2008

## Le répertoire /proc

Catégorie : [Comprendre](#) Tags : [LP](#)



Retrouvez cet article dans : [Linux Pratique 32](#)

Si vous avez eu la curiosité de « fouiner » dans l'arborescence de votre système, afin d'étudier la composition de celui-ci, peut-être avez-vous aperçu certains répertoires dont vous n'avez pas encore bien saisi le rôle. Le répertoire /proc, qui se situe comme vous pouvez le constater à la racine de votre système, et ce, quelle que soit votre distribution, fait partie de ceux-là. Nous allons découvrir ensemble à quoi sert ce répertoire et ce qu'il contient...

## À quoi sert /proc ?

Le répertoire ~~/proc~~ contient des fichiers dont le noyau de votre système se sert pour envoyer des informations aux différents processus. On dit que c'est un « pseudo système de fichiers ». Contrairement aux autres répertoires, ~~/proc~~ est sauvegardé en mémoire et non sur votre disque dur. En effet, ~~/proc~~ contient une hiérarchie de fichiers spéciaux qui représentent l'état actuel du noyau, la mémoire est donc l'endroit le plus approprié pour le stockage de ces informations (puisque'elle est aussi gérée par le noyau). Cela permet aux applications ainsi qu'aux utilisateurs d'avoir un œil sur le noyau du système.

Certains fichiers tels que ~~/proc/interrupts~~, ~~/proc/meminfo~~, ~~/proc/mounts~~ et ~~/proc/partitions~~ fournissent un aperçu de l'environnement d'un système à un moment donné. D'autres, tels que ~~/proc/filesystems~~ et le répertoire ~~/proc/sys/~~ fournissent des informations sur la configuration du système.

Vous pouvez trouver dans le répertoire ~~/proc~~ de nombreuses informations relatives à la configuration matérielle du système et aux processus en cours d'exécution. De plus, certains des fichiers situés dans l'arborescence du répertoire ~~/proc~~ peuvent être manipulés par les utilisateurs ainsi que par les applications afin de transmettre des changements de configuration au noyau.

Attention ! Il est en général risqué de modifier ou d'écrire dans les fichiers contenus dans ~~/proc~~. En effet, la modification de ces fichiers peut entraîner la modification des paramètres de votre noyau à la volée. Donc, une maladresse de votre part et c'est l'erreur fatale ! Votre système est corrompu... Soyez absolument certain de ce que vous faites avant de modifier quoi que ce soit.

### Petit rappel :

- ~~cat~~ liste le contenu d'un ou plusieurs fichiers.
- ~~less~~ affiche le contenu du fichier et vous permet de le faire défiler avec les flèches du curseur. Lorsque vous désirez terminer la visualisation, il suffit de taper la touche [Q] (pour quitter).
- ~~more~~ effectue à peu près le même travail que ~~less~~, mais n'affiche le texte que page par page. Pour passer à la page suivante, il faut appuyer sur la barre d'espace.

## Que trouve-t-on dans /proc ?

Les fichiers ~~/proc~~ peuvent être utilisés pour accéder aux informations concernant l'état du noyau, les attributs du matériel, l'état de fonctionnement des processus, etc.

*Remarque : La syntaxe des fichiers est dépendante de l'architecture sur laquelle tourne le noyau.*

Lorsque l'on parle de fichier, vous pensez immédiatement à un fichier texte ou peut-être un fichier binaire. Ce sont en effet deux types de fichiers assez courants. Le répertoire ~~/proc~~, lui, contient un autre type de fichier : des fichiers « virtuels ». C'est pourquoi ~~/proc~~ est appelé « pseudo système de fichiers » ou encore « système de fichiers virtuels ».

Ces fichiers virtuels sont assez particuliers. Tout d'abord, notez que la plupart d'entre eux ont une taille égale à zéro octet (malgré la grande quantité d'informations qu'ils contiennent !). De plus, les paramètres « date » et « heure » des fichiers virtuels (que vous pouvez observer via la commande `ls -l` dans le répertoire ~~/proc~~) reflètent souvent la date et l'heure actuelles, ce qui montre qu'ils sont mis à jour régulièrement.

Bien que ces fichiers soient virtuels, il est tout à fait possible de lire et d'étudier leur contenu, à partir de n'importe quel éditeur ou à l'aide des commandes ~~more~~, ~~less~~, ~~cat~~. Au moment où vous ouvrez un fichier virtuel, celui-ci est créé immédiatement avec des informations contenues dans le noyau.

## Les sous-répertoires de /proc

Ouvrez un terminal, allez dans le répertoire ~~/proc~~ (`$ cd /proc`) puis affichez le contenu de ce répertoire via la commande `ls`. Vous devriez obtenir quelque chose de similaire à la Figure 1.

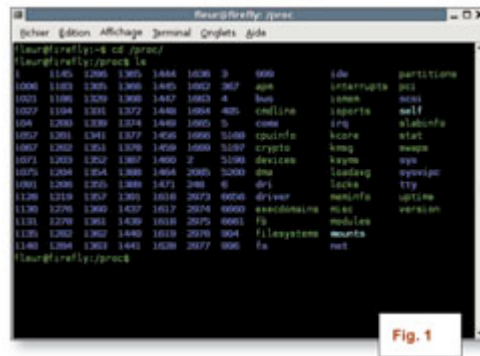


Fig. 1

Constatez que `/proc` contient de nombreux sous-répertoires, ayant des nombres pour nom. En fait, chacun de ces nombres correspond au numéro d'un processus en cours d'exécution. Le système crée automatiquement un répertoire dans `/proc` pour chaque processus en cours. Ainsi, pour chaque processus en cours d'exécution, nous avons le sous-répertoire correspondant qui porte comme nom le PID (numéro d'identification de processus).

Par exemple, j'utilise actuellement OOo Writer. Le commande `ps -ef | grep openoffice` retourne le PID du processus-père openoffice (première ligne) ainsi que les PID des processus-fils openoffice (lignes suivantes). Constatez alors qu'un répertoire a été créé dans `/proc` pour chacun de ces processus (Fig. 2).

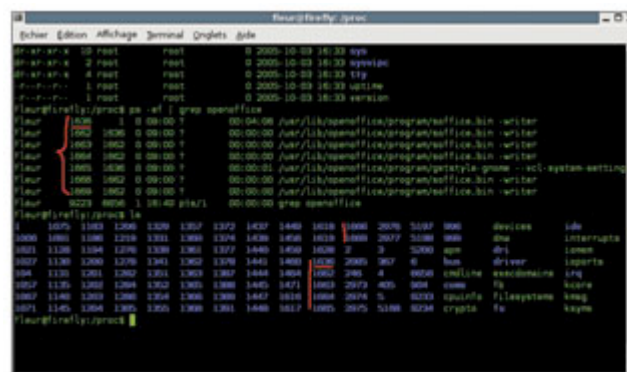


Fig. 2

Chacun des ces sous-répertoires contient diverses informations sur le processus correspondant (Fig. 3), par exemple :



Fig. 3

- Le fichier `cmdline` contient les arguments de la ligne de commande avec lesquels le processus a été lancé.
- Le fichier `environ` concerne l'environnement du processus. À noter : pour afficher ce fichier de manière lisible, tapez la commande suivante : `(cat /proc/XXX/environ; echo) | tr '\000' '\n'` où XXX représente le numéro d'un processus. Ceci permet d'afficher le contenu du fichier en remplaçant les caractères nuls par un retour à la ligne.
- Le fichier `exe` est un lien symbolique vers le chemin complet du processus (c'est-à-dire vers le binaire exécuté).
- Le fichier `statm` indique l'état de la mémoire utilisée par le processus.
- Le fichier `status` contient des informations actualisées sur l'état du processus, parmi

lesquelles : le numéro utilisateur (UID) et le numéro du groupe d'utilisateurs (GID) de l'utilisateur qui a lancé le processus, le numéro d'identification du processus parent (PPID) qui a généré le PID et l'état courant du processus : Sleeping (dormant) ou Running (en cours).

En outre, vous trouverez d'autres répertoires, nommés de façon « classique », parmi lesquels on retiendra :

- ~~driver~~ : Ce répertoire contient des informations sur des pilotes spécifiques utilisés par le noyau.
- ~~ide~~ et ~~scsi~~ : Ces répertoires contiennent des informations sur les périphériques IDE et SCSI du système.
- ~~net~~ : Ce répertoire renferme de nombreux paramètres et statistiques réseau. Chaque fichier couvre une gamme spécifique d'informations relatives à la gestion du réseau sur le système.
- ~~sys~~ : Le répertoire ~~/proc/sys/~~ est différent des autres répertoires de ~~/proc~~, car il vous permet d'apporter des modifications à la configuration du noyau. En effet, l'administrateur de l'ordinateur peut activer et désactiver immédiatement des fonctions du noyau : les modifications sont prises en compte dès qu'elles sont enregistrées.

~~/proc/sys/kernel~~ contient par exemple les informations sur le fonctionnement général du noyau. Ce répertoire renferme entre autres, le nom de domaine et le nom d'hôte pour les ordinateurs du réseau, respectivement dans les fichiers ~~domainname~~ et ~~hostname~~. Vous pouvez par exemple changer le nom de votre machine si vous le souhaitez :

```
$ cat /proc/sys/kernel/hostname
```

Cette commande retourne le nom actuel de votre machine. Pour le changer, tapez la commande :

```
$ echo nouveau_nom > /proc/sys/kernel/hostname
```

Le répertoire ~~/proc/sys/net~~ quant à lui contient des fichiers qui peuvent être modifiés pour changer les propriétés du réseau. Par exemple, nous allons modifier le noyau de façon à ne plus permettre de réponse de votre machine sur le réseau. Autrement dit, votre machine ne répondra plus à la commande ping. Il suffit d'entrer la commande :

```
$ echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Cette commande cachera votre ordinateur du réseau, car elle désactive les messages vers ~~icmp\_echo~~. Si vous lancez maintenant la commande ~~ping~~, suivie de votre adresse IP, rien ne devrait s'afficher. Pour remettre la valeur par défaut, remplacez 1 par 0.

## Les fichiers de /proc

*Remarque :*

*Toute modification de configuration effectuée à l'aide de la commande ~~echo~~ disparaîtra lorsque le système sera redémarré. Pour faire en sorte que vos modifications soient appliquées au démarrage, utilisez plutôt la commande ~~sysctl~~.*

Le répertoire ~~/proc~~ contient de nombreux fichiers. Nous allons préciser ici le rôle de certains d'entre eux.

~~apm~~ : Ce fichier fournit des informations sur l'état du système de gestion de la consommation

d'énergie (APM) (pour Advanced Power Management) et est utilisé par la commande `apm`. Ce fichier a peu d'intérêt pour les systèmes n'utilisant pas de batterie comme source d'alimentation.

~~endline~~: concerne les informations sur les paramètres donnés au noyau lors de la mise en route.

~~cpuinfo~~: contient une collection d'informations sur le ou les processeurs de la machine (modèle, famille, taille du cache, etc.)

~~devices~~: liste des différents périphériques présents classés par groupe (caractère et bloc).

~~core~~: Ce fichier représente la mémoire physique du système. Grâce à ce fichier et à une image du noyau non compressé, l'utilitaire `gdb` (acronyme de Gnu Debugger) peut être utilisé pour déboguer le noyau. La taille de ce fichier est égale à la mémoire physique plus 4 ko. Nous vous déconseillons donc de tenter d'afficher ce fichier ou de l'imprimer et encore moins de le modifier !

~~kmsg~~: Ce fichier est utilisé pour contenir des messages générés par le noyau. Il faut avoir les privilèges root pour en lire le contenu. De plus, si le système utilise `syslog`, le fichier n'est pas lisible, car un seul programme à la fois est autorisé à le lire. Ne vous inquiétez donc pas si la commande `cat /proc/kmsg` ne vous retourne rien.

~~locks~~: Ce fichier affiche les fichiers actuellement verrouillés par le noyau. Chaque verrouillage a sa propre ligne qui commence par un numéro unique.

~~meminfo~~: Ce fichier est l'un des plus utilisés, car il donne de nombreuses informations sur la mémoire vive du système. Il contient des informations sur la mémoire totale, utilisée, libre, etc. (mémoire réelle et swap). L'unité utilisée est l'octet.

~~modules~~: Ce fichier affiche une liste de tous les modules qui ont été chargés dans le noyau. Son contenu varie en fonction de la configuration et de l'utilisation du système.

~~mounts~~: renferme une liste brève de tous les montages utilisés par le système.

~~swaps~~: Ce fichier indique l'espace swap et son utilisation.

~~uptime~~: Ce fichier contient deux nombres, le temps écoulé depuis le démarrage et le temps passé à ne rien faire (en secondes).

~~version~~: Ce fichier contient les informations sur la version du noyau en cours d'utilisation.

Une des nouveautés du noyau 2.6 est le pseudo-fichier ~~/proc/config.gz~~ qui vous permet de récupérer facilement la configuration actuelle de votre noyau. Par exemple : `zcat /proc/config.gz > /usr/src/linux/.config` et le tour est joué !

### Optimiser l'utilisation de la swap (valable pour un noyau 2.6)

Le fichier ~~/proc/sys/vm/swappiness~~ contient une valeur par défaut, accessible via la commande `$ cat /proc/sys/vm/swappiness`. Ce nombre indique le pourcentage des données qui sont mises en cache dans le swap. La valeur par défaut est 60, ce qui signifie que 60% des données sont mises en cache et 40% sont stockées sur la RAM.

Ainsi, si vous faites varier ce nombre, vous influez sur l'utilisation de la swap. Par exemple, si vous fixez cette valeur à 100, toutes les données seront en cache disque. Pour cela, tapez la commande : `# echo 100 > /proc/sys/vm/swappiness`

Et concrètement, quelles sont les conséquences ? Et bien si vous utilisez beaucoup de grosses applications simultanément (OpenOffice, Gimp, client de messagerie, navigateur, etc.) et que vous retournez sur une application inutilisée depuis longtemps (après 3h d'utilisation exclusive d'OpenOffice par exemple), celle-ci va mettre beaucoup plus de temps à répondre (car elle doit être rechargée depuis le cache vers la mémoire vive).

Inversement, si vous fixez la valeur à 10 ou 20%, vos applications vont répondre très rapidement lorsque vous passerez de l'une à l'autre, même si l'une d'entre elles n'a pas été utilisée depuis longtemps. En revanche, si l'une des applications nécessite beaucoup de mémoire, elle fonctionnera plus lentement qu'avec une valeur de `swappiness` plus élevée.

Finalement, la valeur par défaut est sans doute un bon compromis. Mais n'hésitez pas à faire des

[tests...](#)

## Conclusion

Le système de fichiers ~~/proc~~ permet à l'administrateur de connaître l'état d'un système à un instant donné et lui permet de configurer des périphériques et des processus sur un ordinateur. Il est essentiel de bien comprendre la structure et l'utilité de ce système de fichiers pour bien comprendre son système d'exploitation. Avant de penser à le modifier, documentez-vous !

### Liens

- Guide de référence de Red Hat sur le sujet :  
<http://www.europe.redhat.com/documentation/rhl9/rhl-rg-fr-9/ch-proc.php3>
- Quelques astuces pour optimiser votre système :  
[http://lea-linux.org/admin/admin\\_plus/optimise.html](http://lea-linux.org/admin/admin_plus/optimise.html)

Un ouvrage sur le sujet : Olivier Daudel, /proc & /sys, Editions O'Reilly.

Retrouvez cet article dans : [Linux Pratique 32](#)

Posté par Fleur Brosseau ([Fleur](#)) | Signature : Fleur Brosseau | Article paru dans



### Laissez une réponse

Vous devez avoir ouvert une [session](#) pour écrire un commentaire.

« [Précédent](#) [Aller au contenu](#) »

[Identifiez-vous](#)

[Inscription](#)

[S'abonner à UNIX Garden](#)

### • Articles de 1ère page

- [La légalité du spam : une question ouverte](#)
- [Pear et les librairies PHP](#)
- [FreeDOS](#)
- [Mise en œuvre d'une passerelle Internet sous Linux](#)
- [Les pseudo-classes en CSS](#)
- [GNU/Linux Magazine N°108 - Septembre 2008 - Chez votre marchand de journaux](#)

- [Linux Pratique N°49 - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Donner du style à son texte : utiliser les lettrines](#)
- [Installer un serveur Syslog](#)
- [Quatre serveurs FTP hyper sécurisés avec vsftpd](#)



[Actuellement en kiosque :](#)

## • Il y a actuellement

- **719** articles/billets en ligne.

## • Catégories

- - [Administration réseau](#)
  - [Administration système](#)
  - [Agenda-Interview](#)
  - [Audio-vidéo](#)
  - [Bureautique](#)
  - [Comprendre](#)
  - [Distribution](#)
  - [Embarqué](#)
  - [Environnement de bureau](#)
  - [Graphisme](#)
  - [Jeux](#)
  - [Matériel](#)
  - [News](#)
  - [Programmation](#)



- [Réfléchir](#)
- [Sécurité](#)
- [Utilitaires](#)
- [Web](#)

## • Archives

- [août 2008](#)
- [juillet 2008](#)
- [juin 2008](#)
- [mai 2008](#)
- [avril 2008](#)
- [mars 2008](#)
- [février 2008](#)
- [janvier 2008](#)
- [décembre 2007](#)
- [novembre 2007](#)
- [février 2007](#)

## • [GNU/Linux Magazine](#)

- [Ce billet est hors ligne montrer/masquer GNU/Linux Magazine 108 - Septembre 2008 - Chez votre marchand de journaux](#)
- [Edito : GNU/Linux Magazine 108](#)
- [GNU/Linux Magazine HS 38 - Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : GNU/Linux Magazine HS 38](#)
- [GNU/Linux Magazine 107 - Juillet/Août 2008 - Chez votre marchand de journaux](#)

## • [GNU/Linux Pratique](#)

- [Linux Pratique N°49 -Septembre/Octobre 2008 - Chez votre marchand de journaux](#)
- [Edito : Linux Pratique N°49](#)
- [À télécharger : Les fichiers du Cahier Web de Linux Pratique n°49](#)
- [Linux Pratique Essentiel N°3 - Août/Septembre 2008 - Chez votre marchand de journaux](#)
- [Edito : Linux Pratique Essentiel N°3](#)

## • [MISC Magazine](#)

- [Misc 38 : Codes Malicieux, quoi de neuf ? - Juillet/Août 2008 - Chez votre marchand de journaux](#)
- [Edito : Misc 38](#)
- [Références de l'article « Détection de malware par analyse système » d'Arnaud Pilon paru dans MISC 38](#)
- [Références de l'article « La sécurité des communications vocales \(3\) : techniques numériques » d'Éric Filiol paru dans MISC 38](#)



- [Misc 37 : Déni de service - Mai/Juin 2008 - Chez votre marchand de journaux](#)

© 2007 - 2008 [UNIX Garden](#). Tous droits réservés .