

Explorer procfs

Gazette Linux n°115 — Juin 2005

Awasthi Nirendra

Copyright © 2005 Awasthi Nirendra

Copyright © 2005 Deny

Copyright © 2005 Joëlle Cornavin

Article paru dans le n°115 de la Gazette Linux de juin 2005.

Traduction française par Deny <deny CHEZ: monaco POINT net>.

Relecture de la traduction française par Joëlle Cornavin <jcornavi CHEZ club TIRET internet POINT fr>.

Article publié sous Open Publication License (<http://linuxgazette.net/copying.html>). La Linux Gazette n'est ni produite, ni sponsorisée, ni avalisée par notre hébergeur principal, SSC, Inc.

Table des matières

1. Pourquoi cet article ?.....	1
2. Qu'est ce que procfs ?.....	1
3. Le système de fichiers virtuel.....	1
4. Montage de /proc	2
5. Informations sur les processus.....	2
6. Informations générales sur le système	3
7. Quelques informations sur le noyau.....	3

1. Pourquoi cet article ?

J'essaie de rassembler quelques astuces sur la manière d'obtenir de procfs des informations de *userland*. Attention : ces informations peuvent être spécifiques à la version 2.6 du noyau.

2. Qu'est ce que procfs ?

procfs est un système de fichiers virtuel pour Linux, monté dans `/proc`. Il a de multiples objectifs, dont l'accès aux informations du noyau dans un *userland* ou le débogage. Une des fonctionnalités me font apprécier Linux est l'accès aux informations de processus sous forme de flux de texte. Beaucoup de commandes linux (`ps`, `top`, `pstree`, etc.) tirent leurs informations de ce système de fichiers.

3. Le système de fichiers virtuel

Les fichiers et répertoires du système de fichiers `/proc` sont virtuels parce que les données ne sont pas réellement enregistrées sur un quelconque support de stockage permanent comme un disque dur ; en revanche, les dossiers, fichiers et données afférents sont créés dynamiquement en mémoire depuis le noyau chaque fois que vous essayez de les lire.

4. Montage de `/proc`

Vérifiez si vous avez déjà monté `procfs` sur votre système (`grep proc /etc/mtab`), sinon montez-le à l'aide de la commande suivante :

5. Informations sur les processus

Chaque processus a une ligne dans le système de fichiers `/proc`, identifiée par son PID. Voici les fichiers importants du répertoire `/proc` :

- `pid/cmdline` contient la commande invoquée pour démarrer le processus (les caractères nuls servent à séparer les arguments).
- `/proc/pid/cwd` contient un lien vers le répertoire de travail actuel du processus.
- `/proc/pid/envIRON` contient une liste des variables d'environnement que le processus rend disponibles.
- `/proc/pid/exe` contient un lien vers le programme qui s'exécute dans le processus.
- `/proc/pid/fd/` est un répertoire contenant un lien vers chaque fichier que le processus a ouvert.
- `/proc/pid/mem` inclut le contenu mémoire d'un processus.
- `/proc/pid/stat` contient des informations sur l'état du processus.
- `/proc/pid/statm` contient des informations sur l'utilisation de la mémoire du processus.

Voici quelques exemples d'obtention des informations sur les processus :

- i. Il y a quelque temps, j'ai rencontré un problème ardu consistant à déterminer si n'importe quel processus particulier créait un *core dump*. Après quelques recherches, j'ai remarqué que le drapeau de chaque processus dans le fichier `/proc/pid/stat` (le huitième attribut) fournit énormément d'informations « personnelles » sur un processus. Il est possible d'analyser ces informations en faisant un ET logique du drapeau de chaque processus avec les valeurs suivantes : J'ai emprunté ces drapeaux au fichier `/usr/src/linux/include/linux/sched.h`.
- ii. Le dossier `/proc/[pid]/fd/` fournit des informations sur les fichiers ouverts. Pour trouver les fichiers d'entrée qu'utilise un processus, saisissez :
- iii. Pour trouver les *sockets* qu'utilise un processus, saisissez : Pour obtenir des informations sur ces *sockets*, saisissez :
- iv. Pour obtenir les arguments en ligne de commande passés à un processus quelconque, saisissez :

v. Pour obtenir l’ID du processus parent, saisissez :

6. Informations générales sur le système

procfs fournit de nombreuses informations sur le système, y compris la charge du processeur, le système de fichiers et la configuration réseau. Les exemples suivants indiquent comment visualiser ou changer les informations sur le système à l’aide de *procfs* :

- i. Pour connaître la quantité de mémoire disponible sur le système, saisissez :
- ii. Des statistiques sur le système depuis son dernier démarrage peuvent être collectées dans le fichier `/proc/stat`. Pour connaître le nombre de processus présents sur le système depuis son dernier redémarrage, saisissez :
- iii. Pour connaître la charge moyenne du système à la première, cinquième et quinzième minute, saisissez :
- iv. **`/proc/partitions`** permet également d’obtenir des informations sur les partitions du système.
- v. `/proc/net` et `/proc/sys/net` permettent de visualiser ou de modifier des informations importantes sur le réseau. Pour désactiver la commande **ping**, saisissez la commande suivante en tant que *root* : ou pour activer la redirection IP/masquerade IP, saisissez :
- vi. Les informations sur les systèmes de fichiers montés se trouvent dans le fichier `/proc/mounts`.
- vii. Pour changer le nom d’hôte à la volée, saisissez :
- viii. Pour obtenir des informations sur le processeur, saisissez :
- ix. Pour connaître l’utilisation de l’espace d’échange (*swap*), saisissez :
- x. Pour savoir depuis quand le système fonctionne, saisissez :
- xi. Pour répertorier les systèmes de fichiers actuellement partagés par NFS, saisissez :

7. Quelques informations sur le noyau

Bien que j’aie l’intention de traiter ce sujet lors d’un prochain article, voici quelques préliminaires à propos du noyau :

- i. Pour obtenir des informations sur la version du noyau actuellement utilisé, saisissez :
- ii. `klogd` utilise le fichier `/proc/kmsg` comme source d’informations pour les journaux du noyau. C’est aussi une alternative à l’interface d’appel système `syslog`.
- iii. Le fichier `/proc/kcore` permet d’accéder à la mémoire physique du système au format « fichier core » et peut être employé par `gdb` pour examiner l’état actuel de n’importe quelle structure de données du noyau.

Pour obtenir plus d'informations, jetez un coup d'œil au fichier

`/usr/src/linux/Documentation/filesystems/proc.txt` si vous avez installé les sources du noyau.

J'essaierai d'examiner ce système de fichiers du point de vue du noyau dans un prochain article.

Travaillant actuellement en Inde pour Induslogic, je suis titulaire d'une licence d'informatique. Je suis un fervent défenseur du Logiciel libre. Pendant mes loisirs, j'écris des programmes ou je lis des livres. Mes centres d'intérêt se portent sur les pilotes de périphériques, le P2P et les systèmes d'exploitation. Je maintiens mon blogue à l'adresse <http://www.nirendra.net>.