



Les processus sous Linux

Anthony Busson



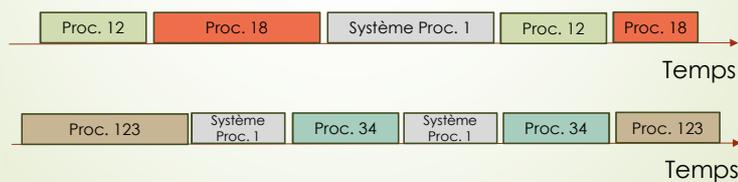
Plan

- Définition et propriété
 - Définition, Etat d'un processus
 - Propriétaire
 - Filiation
 - Signaux
- Les processus particuliers que sont les commandes
 - Background / foreground
 - Dépendance au shell
- Divers

Définition et propriété

Processus: définition

- ▶ Un processus est un programme en cours d'exécution
 - ▶ Il est l'instanciation (en terme d'exécution) d'un programme
- ▶ Plusieurs processus sont exécutés en parallèle sous Linux
 - ▶ Le temps CPU est partagé entre les différents processus
 - ▶ Scheduling/ordonnancement des processus
 - ▶ Un PC peut aussi être multi-cœur et/ou multi-CPU (exécution en parallèle)
- ▶ Plusieurs processus peuvent être issues du même binaire

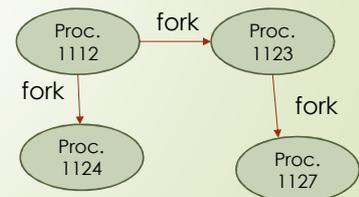


Propriétaire (owner)

- Le propriétaire du processus est celui qui le lance
 - Le propriétaire du binaire peut-être différent du propriétaire du processus
 - Exception: suid/guid
- Le processus hérite des droits du propriétaire
 - Accès aux fichiers
 - Actions demandant des privilèges

Filiation

- Le système de création des processus s'appuie sur un système de clonage (fork())
 - Un processus se duplique à l'identique
 - Il recouvre son code avec le code d'un nouveau binaire
- Créer un système de filiation père / fils
- Hiérarchie de processus
 - Tout processus à un père (sauf le premier)
 - Le fils hérite des propriétés du père lors de sa création:
 - fichiers ouverts, variables d'environnement, propriétaire
 - Possibilité pour le père de se synchroniser sur son fils
 - Attente de la terminaison (wait ou signal SIG_CHD)
 - Récupération du code de retour (return/exit)



Filiation (2)

- ▀ Le processus initial est le processus init (pid=1)
- ▀ Un fils qui a perdu son père est rattaché au processus init
- ▀ Un fils qui s'est terminé mais pour lequel le père n'a pas géré la terminaison est **un zombie**
 - ▀ Il ne faut pas avoir de zombies sur son système (pas trop en tout cas)

Etat d'un processus

- ▀ Running: en cours d'exécution
- ▀ Stopped: en pause
- ▀ Waiting: en attente d'une ressource (disque, réseau, etc.)
- ▀ Zombie: père qui ne gère pas la terminaison de son fils (qui a terminé de s'exécuter)

PID, PPID, PGID

- PID: process id (unique pour chaque processus)
- PPID: parent PID
- PGID: Process Group ID
 - Groupe de processus.
 - Un processus hérite de celui de son père
 - Possibilité de le changer: `setpgid()`.
 - Intérêt: effectuez une action sur un ensemble de processus (stop, arrêt, gestion des ressources, etc.)
- Session
 - Regroupement de processus (hérité aussi du père).
 - Tous les processus associés à une session utilisateurs ont par défaut le même numéro de session.

Propriété

- Répertoire courant
- Pid, etc.
- Utilisateur propriétaire
- Priorité:
 - Donne plus ou moins de temps d'exécution au processus
- Variables d'environnement
- Shell d'affichage (tty)

Les signaux

- Des signaux peuvent/ont échangés entre les processus ou envoyés par le système
- Notifie d'un événement
- A la réception d'un signal le processus:
 - Se termine
 - S'arrête (STOP)
 - Reprend (après un stop)
 - Ignore le signal
 - Déclenche une fonction (handler)
- Signaux usuels
 - SIGHUP, SIGKILL, SIGINT, SIGTERM: par défaut termine le processus
 - SIGUSR1, SIGUSR2: par défaut ignoré
 - SIGSTOP, SIGCONT: arrête / reprend
 - Etc.

Signaux: utilité

- Permet d'arrêter un processus
 - Ctrl-c: émission d'un signal SIGINT
 - Un shell qui se termine envoie SIGHUP à ses processus fils (les termine)
- Signaux spéciaux permettant à des processus d'effectuer une tâche spéciale
 - Recharger un fichier de configuration par exemple
 - Déclencher un handler
- Mettre en pause et reprendre
- Signaler d'un événement:
 - Terminaison d'un fils
 - Rupture d'une connexion ou d'un IPC (pipe par exemple)



Processus: les commandes



Commandes et binaire lancé du terminal

- ▀ Une commande lancé du shell est un processus
 - ▀ Le père est le shell
- ▀ Un script est lancé dans un sous shell
- ▀ Processus lancé en tâche de fond (background)
 - ▀ Permet d'accéder à la console
__ \$ commande argument &
- ▀ Processus en foreground
 - ▀ Le shell attend la terminaison du processus avant de réafficher le prompt

Commandes et binaire lancé du terminal: Dépendance commandes/shell

- ▶ Le shell envoi un SIGHUP à tous ses processus lorsqu'il se termine
- ▶ Effet: tous les processus se termine
- ▶ Possibilité d'éviter la terminaison:
 - ▶ Handler dans le code du processus
 - ▶ nohup: commande qui lance une commande mais qui ignore le signal SIGHUP

Commandes et binaire lancé du terminal: entrées / sorties

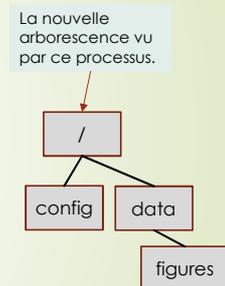
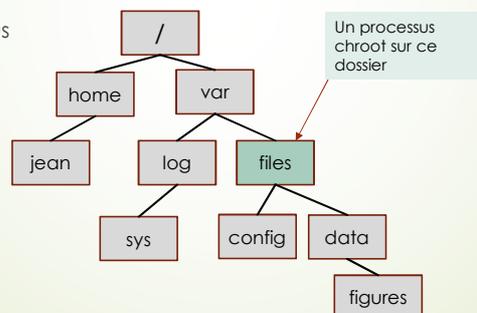
- ▶ 3 entrées sorties sont associées à un processus
 - ▶ Sortie standard → affichage sur le terminal
 - ▶ Sortie erreur -> affichage sur le terminal
 - ▶ Entrée standard -> ce qui est tapé au clavier
- ▶ D'autres entrées sorties peuvent être associées au processus:
 - ▶ Connexions réseaux
 - ▶ Fichiers
 - ▶ IPC (système de communication inter-processus: pipe, etc.)

PGID dans le shell

- Job control: chaque job utilise le pgid de la première commande (égale à son pid)
- Sans job control: toutes les commandes utilisent le même pgid hérité de son parent
- Avec le bash:
 - Commandes du shell: job control
 - Script (lancé du shell): pas de job control

Chroot et isolement

- chroot(2): appel système permettant de fixer une nouvelle racine du système de fichiers
 - Le processus a donc accès qu'à une branche du systèmes de fichiers
- Utilité:
 - Tests
 - Limiter l'accès à un processus
 - Serveur web par exemple





Divers



Daemon / démon

- Démon/Daemon: processus qui s'exécute en tâche de fond et qui rend un service.
- Les services (démarrage, etc.):
 - `init.d` ou `service serviceName stop | start | restart | reload`
 - Apparaissent dans le dossier `service` (scripts de démarrage)
- Log
 - `/var/log`

Planifier

- ▶ Crontab
 - ▶ Permet de planifier des tâches.
 - ▶ En exercice.

Commandes associés

- ▶ top / htop: liste les processus en temps réel
- ▶ kill: émission d'un signal
- ▶ ps: liste les processus
- ▶ pstree: affiche la filiation des processus
- ▶ Who
- ▶ nohup: évite la terminaison d'un processus lancé du shell
- ▶ nice: attribue une priorité au processus (autre que celle par défaut)
- ▶ chroot