

Administration d'un système GNU / Linux

02 — Processus et gestion des programmes

Anthony Labarre

上海师范大学



Plan d'aujourd'hui

- ① Processus
- ② Gestion des programmes
- ③ Gestion de la configuration
- ④ Recherche de fichiers

Processus

Programmes

N'importe quel fichier peut contenir un programme, mais il faut:

- 1 que le fichier soit exécutable;
- 2 que vous ayez la permission de l'exécuter.

Pour lancer un programme à partir du terminal, on tape:

- chemin/vers/./programme (le ./ est important!)
- ou programme si chemin/vers/ est dans \$PATH (cf.plus tard);



Lancer un programme à partir du terminal permet de voir ses messages d'erreur.

Récupérer le terminal

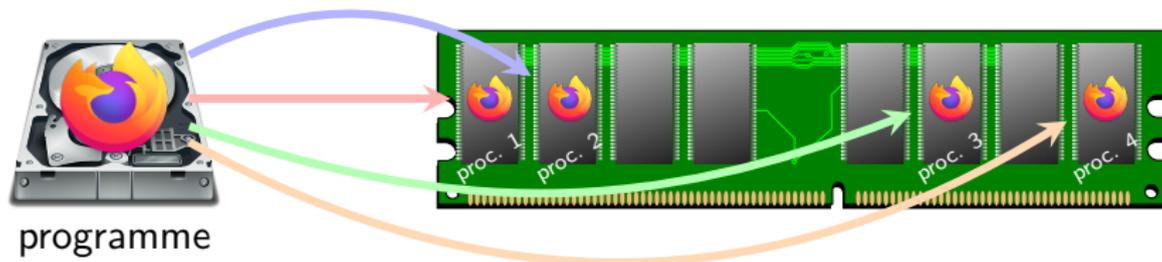


Lorsqu'on lance un programme à partir du terminal, on perd parfois l'accès au terminal (exemple: `evince`).

Solution: utiliser `programme &` au lieu de `programme`.

Identifier les processus

Un **processus** est une instance d'un programme en cours d'exécution.



On a vu que:

- 👤 un utilisateur a un nom et un numéro (UID, **u**ser **i**dentifier);
- 👥 un groupe a un nom et un numéro (GID, **g**roup **i**dentifier);

De la même façon, un processus a un nom et un numéro: le PID (**p**rocess **i**dentifier).

Obtenir le PID d'un processus

- La commande `pgrep motif` donne les PID de tous les processus dont le nom contient `motif`;
- Pour avoir aussi leurs noms, utilisez `pgrep -l`;

☰ Surveiller les processus: top

- top donne la liste des processus actifs, mise à jour en temps réel;
- Elle permet de savoir “qui” consomme le processeur ou la mémoire;

```
anthony@debian: ~$ top
top - 14:44:27 up 10 days, 16:58, 3 users, load average: 0,50, 0,78, 0,84
Tasks: 408 total, 1 running, 407 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5,4 us, 1,4 sy, 0,0 ni, 92,3 id, 0,1 wa, 0,0 hi, 0,8 si, 0,0 st
MiB Mem : 31800,8 total, 2446,5 free, 16025,7 used, 17236,7 buff/cache
MiB Swap: 48868,0 total, 39972,6 free, 8895,4 used. 15775,0 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 119540 anthony  20   0 106,7g 912500 42140 S  11,6  2,8    6,10  apostrophe
 627628 anthony  20   0 1132,1g 145704 99476 S   9,3  0,4   39:45.33 chromium
 517411 anthony  20   0 7822284 1,4g 140084 S   7,9  4,4   17:24.78 strawberry
   3495 anthony  20   0 7058992 546604 303984 S   7,6  1,7   127:34.12 gnome-shell
 182755 anthony  20   0 6154924 305772 93776 S   4,0  0,9   223:01.39 zoom
 119623 anthony  20   0 103,2g 113500 54456 S   3,0  0,3   105:07.40 WebKitWebProces
 620549 anthony  20   0 33,0g 150936 104812 S   3,0  0,5   12:00.14 chromium
 23955 anthony  20   0 714844 88540 51600 S   2,3  0,3    9:52.36 gnome-terminal-
220346 anthony  20   0 508408 2012 2012 S   2,3  0,0   85:37.40 aomhst
 619803 anthony  20   0 13,4g 827436 312664 S   2,0  2,5   27:43.58 firefox-esr
288281 anthony  20   0 8735924 2,4g 51672 S   1,3  7,6   45:28.92 pycharm
   3344 anthony  20   0 271056 22784 11136 S   1,0  0,1    0:40.48 wireplumber
 307293 anthony  20   0 2589476 57468 17848 S   1,0  0,2   19:48.72 Isolated Web Co
   3345 anthony  20   0 118144 76624 6976 S   0,7  0,2    7:16.68 pipewire-pulse
247965 anthony  20   0 2751140 118860 53292 S   0,7  0,4   15:49.91 ktorrent
 619983 anthony  20   0 3121268 330424 96056 S   0,7  1,0   16:22.85 WebExtensions
 620349 anthony  20   0 3012576 390816 122896 S   0,7  1,2    2:15.96 Isolated Web Co
```

- Critères de tri utiles: %CPU (par défaut) ou %MEM (avec  + );
- Quitter le programme: ;

Tuer les processus

- Si un processus est trop gourmand, on peut le “tuer” (l’arrêter);
- Deux moyens:
 - ① `kill -9 PID;`
 - ② `pkill -9 nom_processus;`



`pkill -9 nom` tue **tous** les processus portant ce nom!

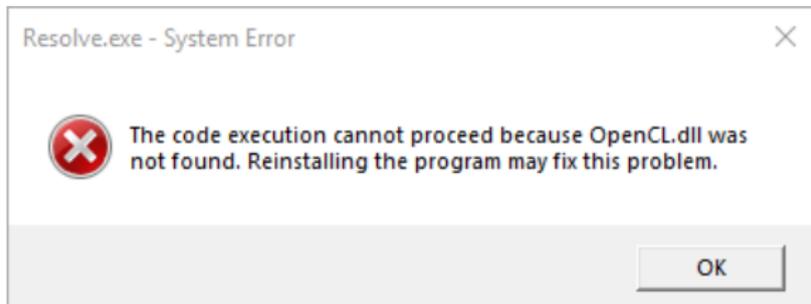
Commentaires sur `kill`

- Contrairement à son nom, `kill` sert en fait à envoyer un **signal** au processus donné;
- `-9` est le signal SIGKILL, qui tue les processus; mais il y en a d'autres:
 - ▣▣ SIGSTOP (`-19`) interrompt le processus;
 - ▶ SIGCONT (`-18`) reprend l'exécution du processus interrompu;
 - ⋮
- `kill -l` donne la liste des signaux disponibles;

Gestion des programmes

Dépendances

Une **dépendance** est un fichier dont un programme a besoin. Si elle n'est pas satisfaite, on a des erreurs. Par exemple, sous Windows:



(adapté de <https://flic.kr/p/Hz254R>)

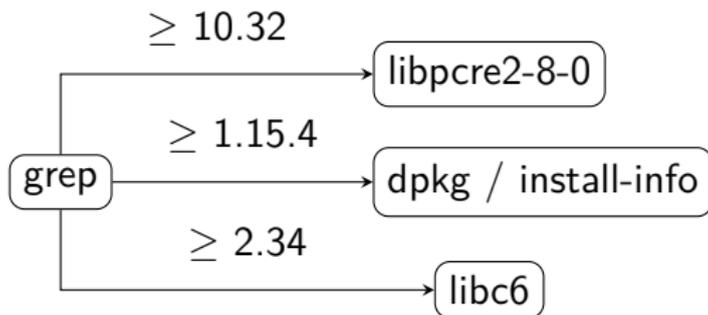
Paquets

- Les distributions GNU / Linux évitent ce problème à l'aide des *paquets*;
- Un **paquet** contient:
 - ① le programme à installer;
 - ② des fichiers auxiliaires;
 - ③ et les informations sur les dépendances éventuelles;
- Si des dépendances manquent au moment de l'installation, le système les installera automatiquement en plus du programme;

Paquets

Exemple

Voici les dépendances du paquet **grep** sous Debian:



🕒 Sous Debian, les paquets possèdent l'extension `.deb`.

Le système APT

- APT (pour **A**dvanced **P**ackage **T**ool) est une collection d'outils gérant les paquets;
- On y trouve:
 - apt: (dés)installation et recherche de paquets;
 - apt-cache: recherche de paquets avec descriptions;
 - apt-file: recherche de fichiers dans des paquets;
 - ⋮

② Fonctionnement de apt

- apt utilise des **sources**: les endroits où aller chercher les paquets à installer;
- Elles sont spécifiées dans le fichier `/etc/apt/sources.list`;

Exemple (fichier `/etc/apt/sources.list`)

```
deb http://deb.debian.org/debian/ bookworm main non-free non-free-firmware contrib
deb-src http://deb.debian.org/debian/ bookworm main non-free non-free-firmware contrib

deb http://security.debian.org/debian-security bookworm-security main non-free
↳ non-free-firmware contrib
deb-src http://security.debian.org/debian-security bookworm-security main non-free
↳ non-free-firmware contrib
```

- S'il vous faut des programmes qui ne sont pas disponibles sous Debian par défaut, il faudra:
 - configurer les sources deb (préférable si elles existent); ou
 - utiliser un autre moyen (par exemple récupérer un fichier `.deb`, ou une archive `.tar.gz` et compiler les sources);

② Installation et désinstallation



`sudo apt install monpaquet`: installe monpaquet



`sudo apt remove monpaquet`: supprime monpaquet



`sudo apt purge monpaquet`: supprime aussi les fichiers de configuration



`sudo apt autoremove`: supprime les dépendances qui ne sont plus nécessaires



`apt search mots-clés` ou `apt-cache search mots-clés`: affiche les paquets correspondant aux mots-clés;



Exemple avec le paquet `retext`.

② Trouver les programmes à installer

- Le système de paquets évite les problèmes de dépendances;
- Mais parfois, on doit exécuter ou installer un programme ne venant pas d'un paquet, et il peut manquer des fichiers;
- `apt-file` permet de trouver les paquets contenant ces fichiers;

Exemple

```
$ apt-file search flags/zh.png  
grass-gui: /usr/share/grass78/gui/icons/flags/zh.png
```

Mises à jour (un, plusieurs, tous les paquet(s))

- ① On doit mettre à jour la base de données: `sudo apt update`
- ② Et ensuite utiliser:
 - `sudo apt upgrade mon_paquet` (met à jour `mon_paquet`), ou
 - `sudo apt upgrade` (met à jour **tous** les paquets);

Gestion de la configuration

Configuration globale

- La *configuration globale* s'applique à tous les utilisateurs;
- Les fichiers correspondants se trouvent dans /etc;
- Sauf exception, vous ne pouvez pas les modifier (propriétaire et groupe: root);

Configuration locale

- La *configuration locale* s'applique uniquement à vous;
- Les fichiers correspondants se trouvent quelque part dans votre répertoire personnel ...
- ... mais ils sont "cachés": leur nom commence par .



Pour afficher les fichiers cachés, utilisez:

- `ls -a` (tout, y compris les fichiers cachés);
- `ls -d .*` (seulement les fichiers cachés);
- Vous pouvez bien sûr les modifier;

Éditeurs de texte

- Les éditeurs “célèbres” `vi` et `emacs` sont trop complexes pour un débutant;
- À la place, `nano` suffira;
- Il existe aussi des éditeurs de texte non-interactifs (`sed`, `awk`, ...) dont nous parlerons si nous avons le temps;

Visualisation de texte

Parfois, on n'a pas besoin de modifier le texte, seulement le lire. On peut utiliser:

- `cat fichier`: affiche le contenu d'un fichier en entier dans le terminal;
- `less fichier`: affiche le contenu d'un fichier de manière interactive (on peut le faire défiler);
- `head fichier`: affiche les premières lignes d'un fichier;
- `tail fichier`: affiche les dernières lignes d'un fichier;

Processus
○○○○○○○○

Gestion des programmes
○○○○○○○○○

Gestion de la configuration
○○○○○

Recherche de fichiers
●○○○○

Recherche de fichiers

🔍 Trouver des fichiers

- apt et apt-search cherchent des fichiers dans des paquets (installés ou non);
- Comment chercher des fichiers **sur le disque**?
- Il existe deux outils principaux pour trouver des fichiers sous GNU:
 - ① find, installé par défaut;
 - ② locate, à installer;

🔍 locate

En général, l'utilisation de `locate` se limite à donner le nom ou le chemin du fichier.

Exemple

```
$ locate vmlinuz # renvoie tous les chemins contenant "vmlinuz"
/boot/vmlinuz-5.14.0-0.bpo.2-amd64
/boot/vmlinuz-5.18.0-0.bpo.1-amd64
/boot/vmlinuz-6.1.0-23-amd64
/boot/vmlinuz-6.1.0-25-amd64
/usr/share/go-1.19/src/debug/pe/testdata/vmlinuz-4.15.0-47-generic
/vmlinuz
/vmlinuz.old
```

- ✓ très facile à utiliser;
- ✓ très rapide;
- × pas toujours installé par défaut;
- × utilise une base de données qui doit être à jour (ça se fait automatiquement, mais il peut y avoir des délais);

🔍 find

La syntaxe de `find` est la suivante:

```
find [options] point_de_départ expression
```

Exemple (même recherche que l'exemple précédent)

```
$ find / -name *vmlinuz* 2>/dev/null # renvoie tous les chemins  
↪ contenant "vmlinuz"  
/usr/share/go-1.19/src/debug/pe/testdata/vmlinuz-4.15.0-47-generic  
/vmlinuz  
/vmlinuz.old  
/boot/vmlinuz-5.18.0-0.bpo.1-amd64  
/boot/vmlinuz-6.1.0-23-amd64  
/boot/vmlinuz-5.14.0-0.bpo.2-amd64  
/boot/vmlinuz-6.1.0-25-amd64
```

- ✓ installé par défaut;
- ✓ permet des recherches beaucoup plus avancées;
- × plus complexe à utiliser;
- × plus lent, car `find` doit vraiment tout parcourir;

🔍 find ou locate?

- Installez et utilisez locate quand c'est possible;
- Si vous devez limiter la recherche à un répertoire particulier, find est plus simple;
- Par exemple, trouver tous les fichiers PDF dans un répertoire personnel;



`sudo updatedb` met à jour la base de données de locate.