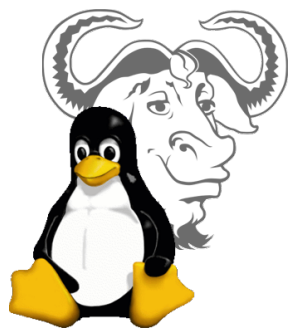
Administration d'un système GNU / Linux

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 ayiez la permission de l'exécuter.

Programmes

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

1. que le fichier soit exécutable;
2. que vous ayiez 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);

Programmes

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

1. que le fichier soit exécutable;
2. que vous ayiez 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.

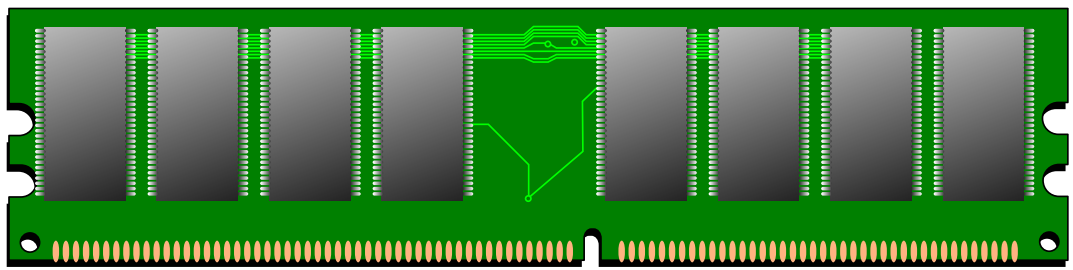
Récupérer le terminal

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

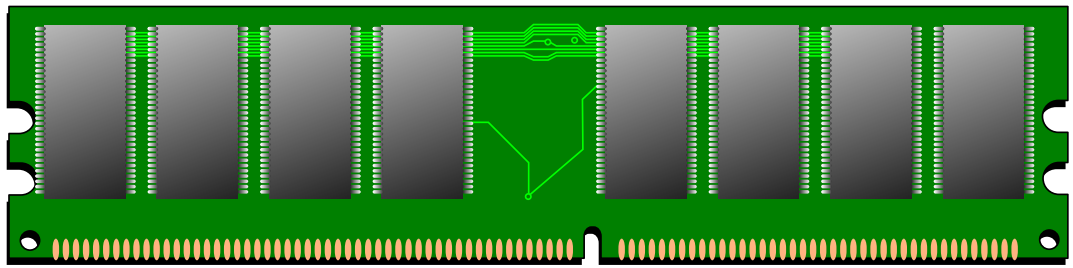
Un **processus** est une instance d'un programme en cours d'exécution.



Identifier les processus

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

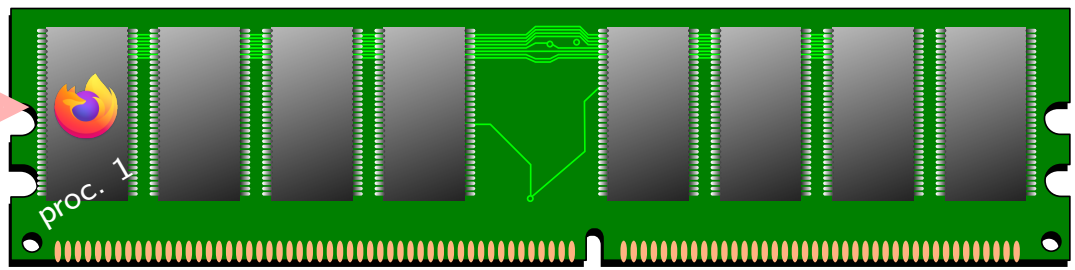
programme



Identifier les processus

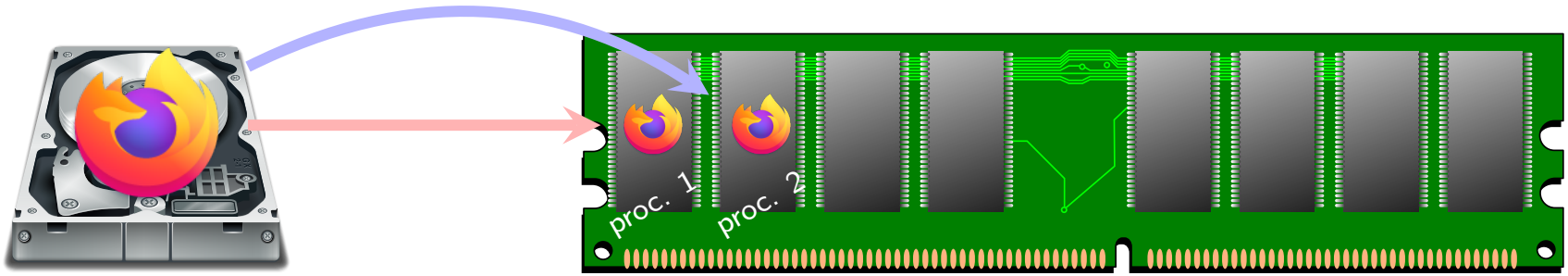
Un **processus** est une instance d'un programme en cours d'exécution.

programme



Identifier les processus

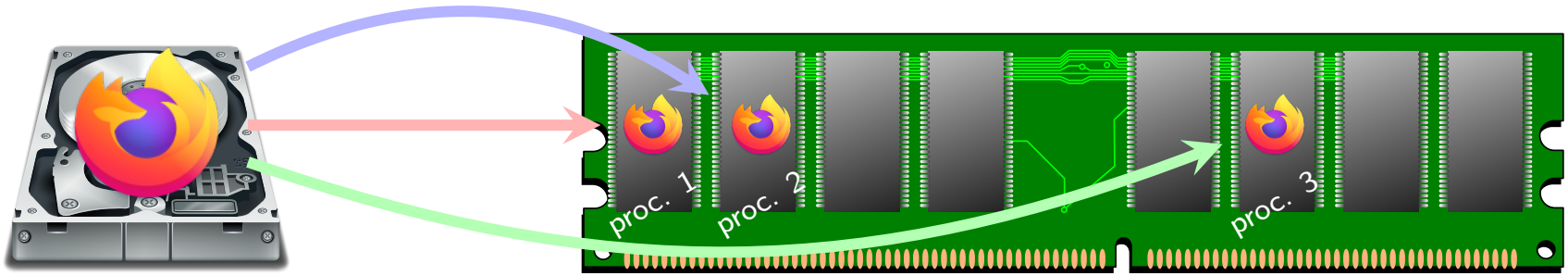
Un **processus** est une instance d'un programme en cours d'exécution.



programme

Identifier les processus

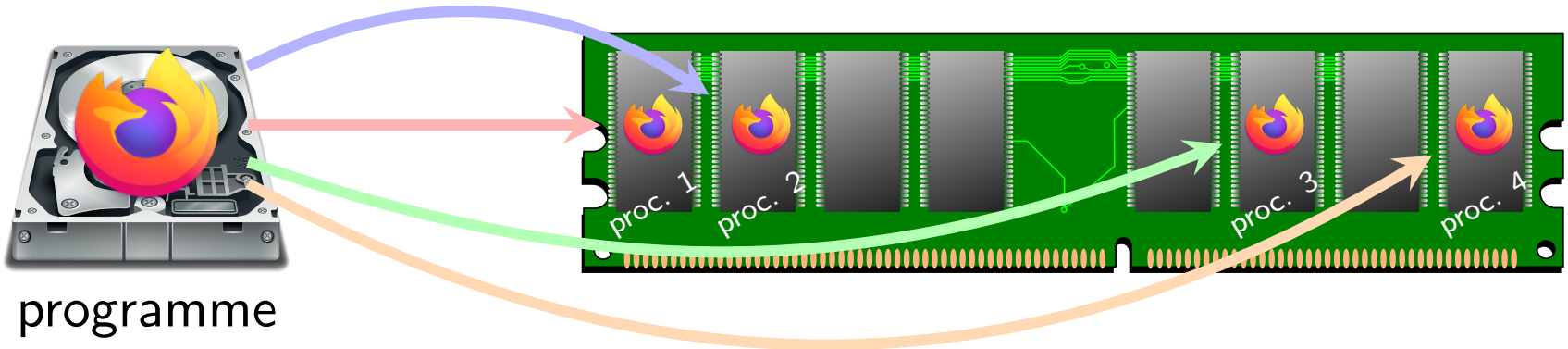
Un **processus** est une instance d'un programme en cours d'exécution.



programme

Identifier les processus

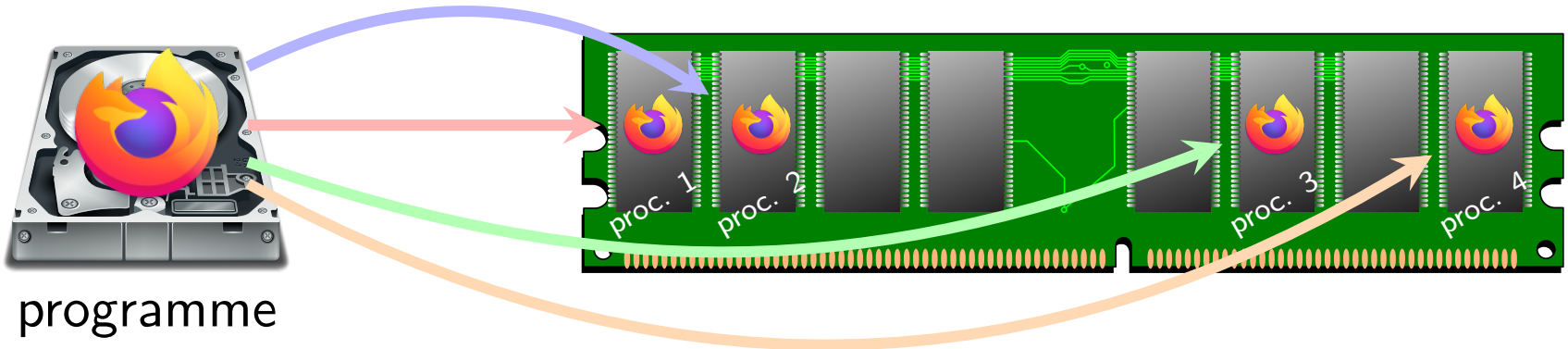
Un **processus** est une instance d'un programme en cours d'exécution.



programme

Identifier les processus

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

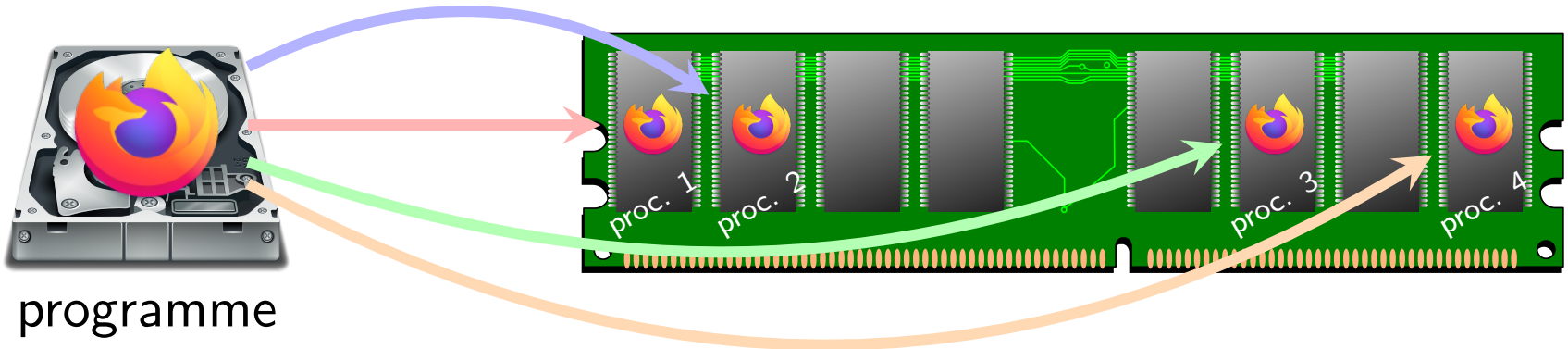


programme

On a vu que:

Identifier les processus

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



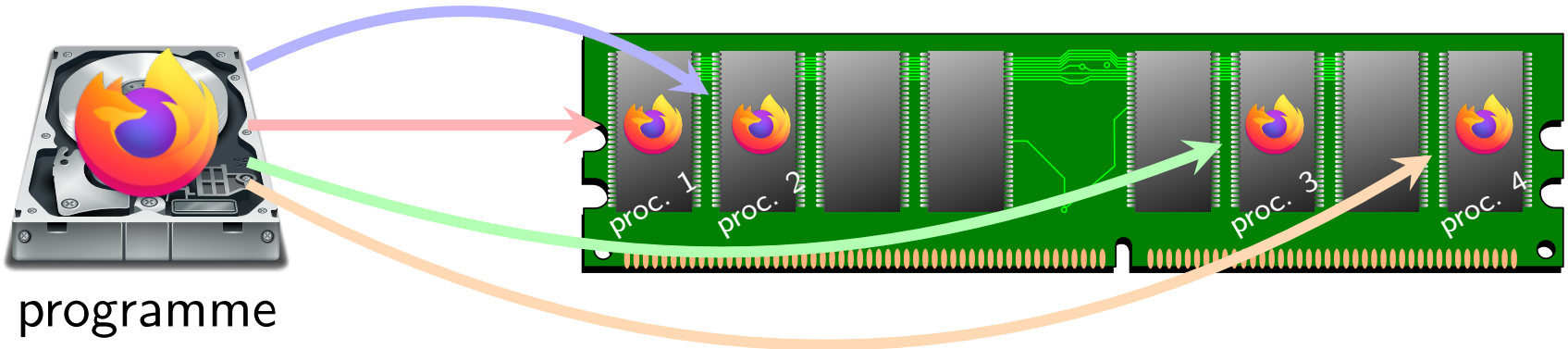
programme

On a vu que:

- un utilisateur a un nom et un numéro (UID, **u**ser **id**entifier);
- un groupe a un nom et un numéro (GID, **g**roup **id**entifier);

Identifier les processus

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



programme

On a vu que:

- un utilisateur a un nom et un numéro (UID, **u**ser **id**entifier);
- un groupe a un nom et un numéro (GID, **g**roup **id**entifier);

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

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`

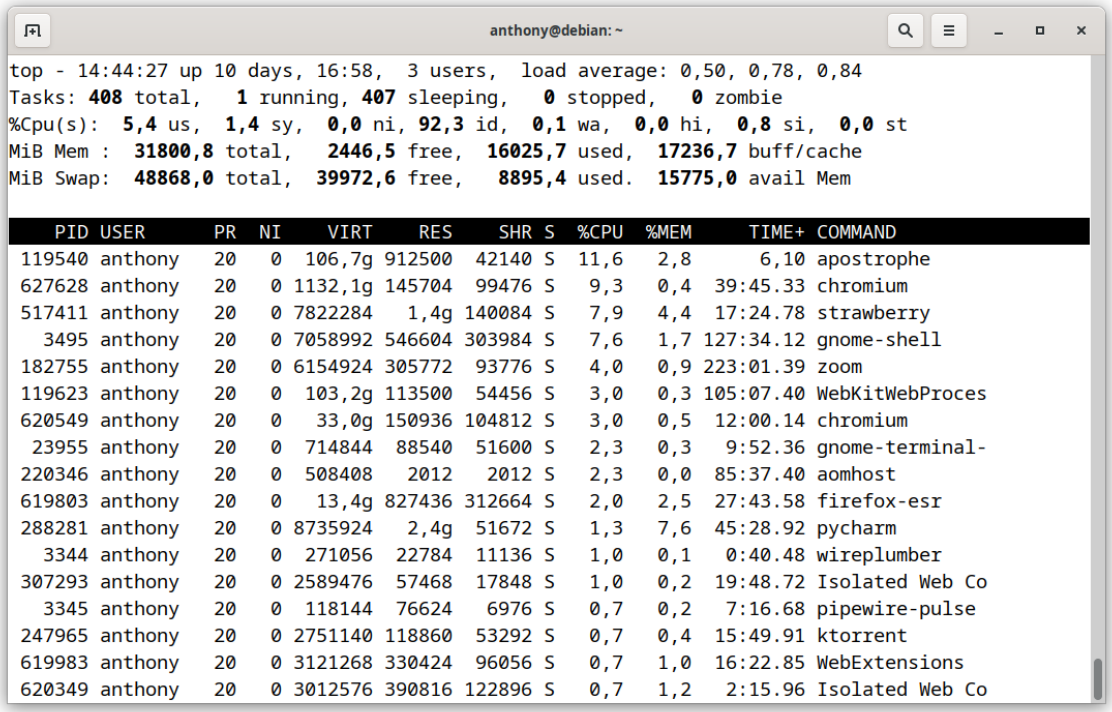
- `top` donne la liste des processus actifs, mise à jour en temps réel;

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;

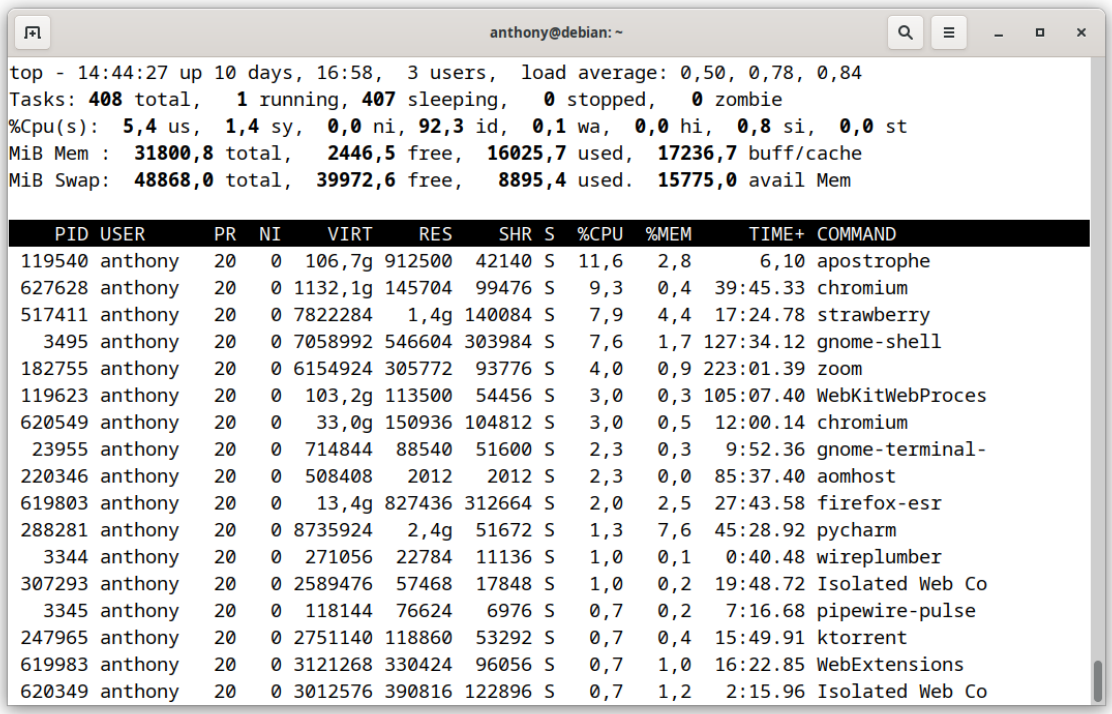
Surveiller les processus: `top`

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



Surveiller les processus: `top`

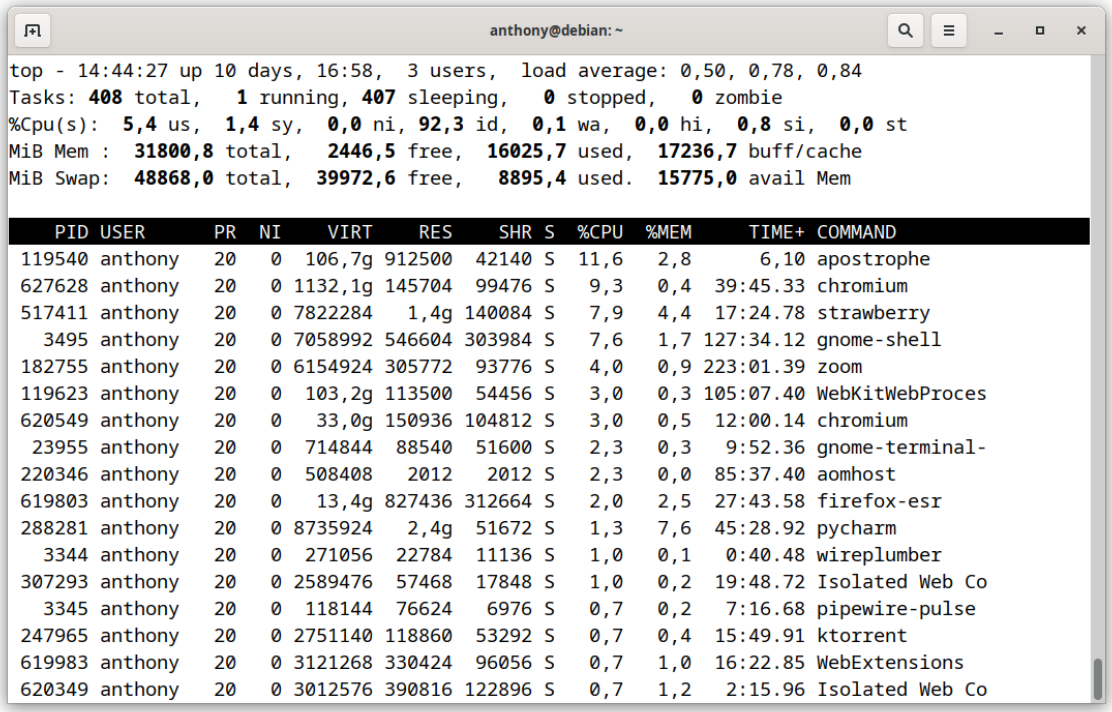
- `top` donne la liste des processus actifs, mise à jour en temps réel;
- Elle permet de savoir “qui” consomme le processeur ou la mémoire;



- Critères de tri utiles: %CPU (par défaut) ou %MEM (avec ⇧ + M);

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;



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

Tuer les processus

- Si un processus est trop gourmand, on peut le “tuer” (l’arrêter);

Tuer les processus

- Si un processus est trop gourmand, on peut le “tuer” (l’arrêter);
- Deux moyens:

Tuer les processus

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

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`;

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;

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:

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;

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;
  - ⋮

Commentaires sur `kill`

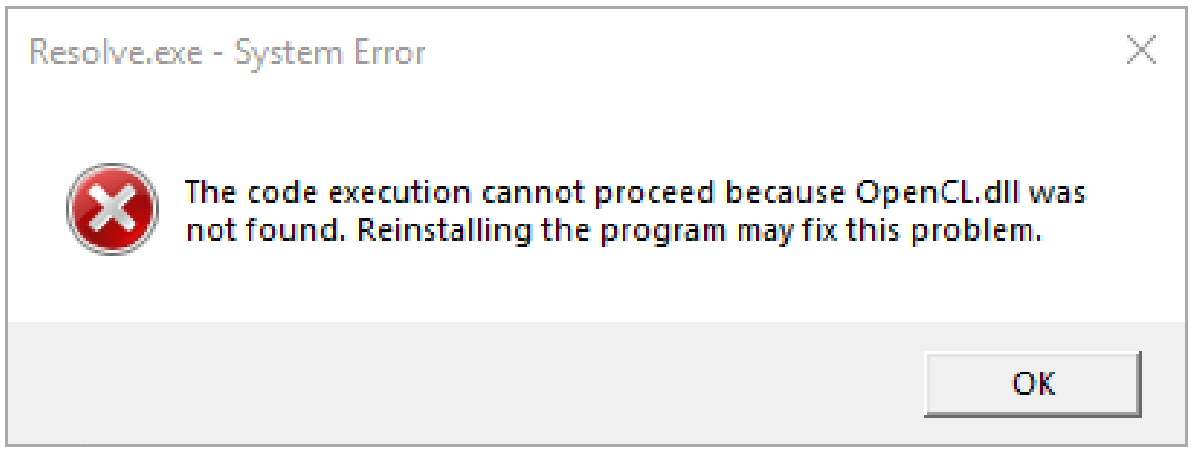
- Contrairement à son nom, `kill` sert en fait à envoyer un **signal** au processus donné;
- `-9` est le signal `SIGKILL`, qui tue les processus; mais il y en a d'autres:
  - `SIGSTOP` (`-19`) interrompt le processus;
  - `SIGCONT` (`-18`) reprend l'exécution du processus interrompu;

  ⋮
- `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*;

Paquets

- Les distributions GNU / Linux évitent ce problème à l'aide des *paquets*;
- Un **paquet** contient:

Paquets

- Les distributions GNU / Linux évitent ce problème à l'aide des *paquets*;
- Un **paquet** contient:
  1. le programme à installer;

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;

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;

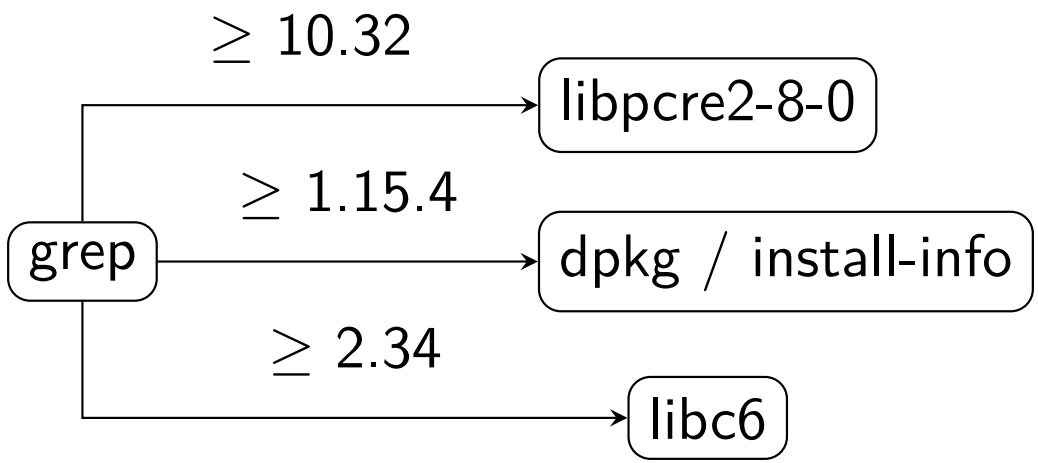
Paquets

- 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;

Le système APT

- APT (pour **A**dvanced **P**ackage **T**ool) est une collection d'outils gérant les paquets;
- On y trouve:

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;

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;

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;
  - ⋮

Fonctionnement de apt

- apt utilise des **sources**: les endroits où aller chercher les paquets à installer;

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`;

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
```

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:

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

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);

Installation et désinstallation

- `sudo apt install monpaquet`: installe `monpaquet`

Installation et désinstallation

`sudo apt install monpaquet`: installe `monpaquet`

`sudo apt remove monpaquet`: supprime `monpaquet`

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

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

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;

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`.

Trouver les programmes à installer

- Le système de paquets évite les problèmes de dépendances;

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;

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;

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;

Configuration globale

- La *configuration globale* s'applique à tous les utilisateurs;
- Les fichiers correspondants se trouvent dans `/etc`;

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;

Configuration locale

- La *configuration locale* s'applique uniquement à vous;
- Les fichiers correspondants se trouvent quelque part dans votre répertoire personnel . . .

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 .

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:

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);

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);

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;

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);

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;

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;

Recherche de fichiers

Trouver des fichiers

- `apt` et `apt-search` cherchent des fichiers dans des paquets (installés ou non);

Trouver des fichiers

- `apt` et `apt-search` cherchent des fichiers dans des paquets (installés ou non);
- Comment chercher des fichiers **sur le disque**?

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:

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;

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.

`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
```

`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
```

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
```

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`.