

Administration d'un système GNU / Linux

01 — Les bases

Anthony Labarre

上海师范大学

30 octobre 2023



Mises en garde

- Le français est plus difficile que le chinois (pour vous);
- Donc:
 - si le cours va trop vite, interrompez-moi;
 - si le français n'est pas clair: faites-moi répéter;
 - si la matière n'est pas claire: posez des questions;

Organisation du cours



“**Cours magistral**” (CM): explications sur la matière;



“**Travaux pratiques**” (TP): exercices sur ordinateur;



Slides, syllabus, exercices, scripts:

<http://igm.univ-mlv.fr/~alabarre/teaching.php#section=linux>



Anthony . Labarre @ univ-eiffel . fr (sans espaces)

- Révisez le dernier CM avant la séance de TP;
- Terminez les exercices de TP chez vous;
- Pendant les séances de TP: consultez le cours si vous ne savez pas répondre!
- Si un exercice est trop difficile: dites-le, on le corrigera ensemble;

Toute fraude sera évidemment sanctionnée.

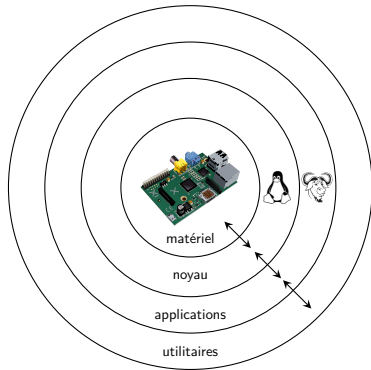
Plan d'aujourd'hui

- 1 Introduction
- 2 Le terminal
- 3 Fichiers
- 4 Utilisateurs
- 5 Processus
- 6 Variables d'environnement
- 7 Administration basique

Introduction

Qu'est-ce que GNU? Linux? GNU / Linux?

GNU 🐉 est un **système d'exploitation** (comme Windows ou OS X), qui utilise le **noyau** Linux 🐧 pour former le système GNU / Linux.



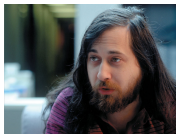
Souvent, on raccourcit “GNU / Linux” en “Linux”. Mais d'autres noyaux peuvent être utilisés (Hurd, BSD, ...).

Composantes de GNU / Linux



Le système GNU

- Créé en 1983 par Richard M. Stallman;
- Clone gratuit et “libre” (voir plus loin) de Unix;
- GNU = “**GNU's Not Unix**” (acronyme récursif);



Le noyau Linux

- Créé en 1992 par Linus B. Torvalds;
- Torvalds a également créé git;

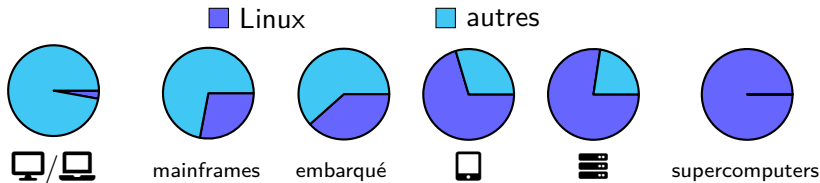


Free software

- GNU et Linux sont du **free software**; il s'agit de “logiciel libre”, et pas “logiciel gratuit” (*free as in freedom*)!
- GNU définit quatre libertés essentielles (paraphrasées):
 - ① utiliser le logiciel sans restrictions;
 - ① étudier et modifier le code source du logiciel;
 - ② redistribuer sans restrictions le logiciel ...
 - ③ ... et vos modifications!
- On peut donc étudier et améliorer le système à souhait ...
- ... et même créer son propre système GNU / Linux (sa **distribution**);

Utilisation de Linux

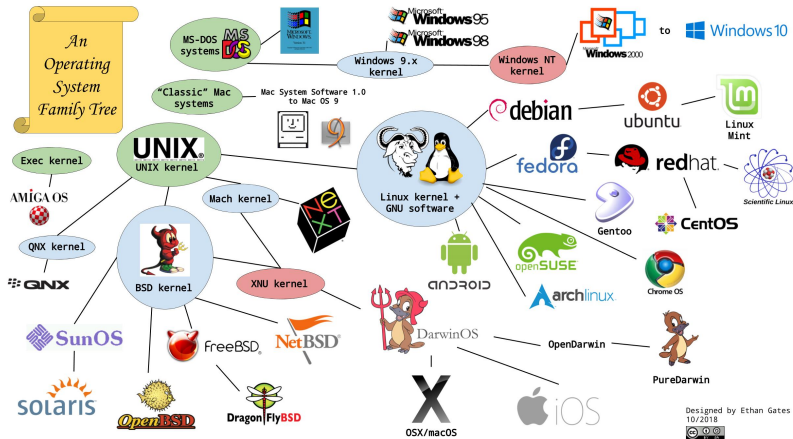
Linux est-il répandu? Les parts de marché dépendent du secteur:



(source:

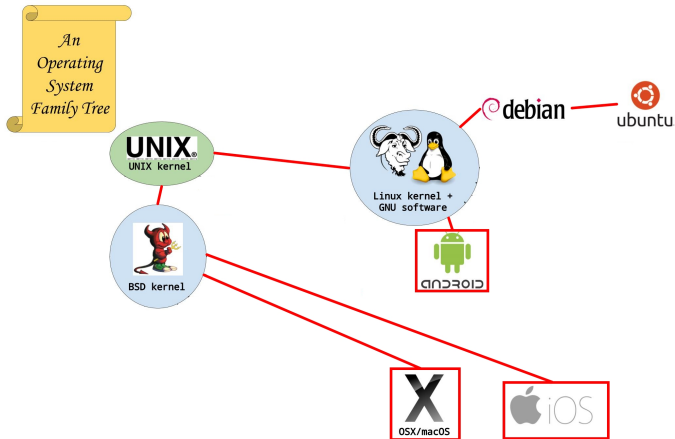
https://en.wikipedia.org/wiki/Usage_share_of_operating_systems;
on parle bien ici du **noyau** Linux, avec ou sans GNU)

GNU / Linux et le restant du monde



Designed by Ethan Gates
10/2018

GNU / Linux et le restant du monde



Quelle distribution choisir?

- Il existe beaucoup de variantes de GNU / Linux, que l'on appelle des **distributions**;
- On utilisera **Ubuntu**, plus accessible aux débutants;



Les distributions se ressemblent très fort \Rightarrow ce qu'on apprend sous Ubuntu nous servira sous les autres distributions.

Les notions spécifiques à Ubuntu seront précédées de son logo: 

Séances d'exercices

Nous utiliserons une **machine virtuelle** (voir slide suivant);

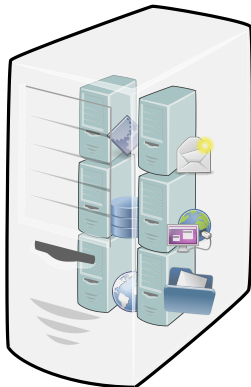
- ✓ installation très simple de GNU / Linux, sans modifier votre machine;
- × performances moins bonnes;



Quand vous serez un peu plus à l'aise avec Ubuntu, essayez de l'installer "normalement" (faites des sauvegardes avant!).

Fonctionnement des machines virtuelles

- Une **machine virtuelle** est un programme qui simule un ordinateur; on y trouve donc:
 - un disque dur virtuel;
 - de la RAM virtuelle;
 - un lecteur CD virtuel dans lequel on “insèrera” l'image ISO du système à installer;
- Votre machine (réelle) est l'**hôte** (= *host*);
- La machine virtuelle est l'**invité** (= *guest*);



Caractéristiques de GNU / Linux

GNU / Linux se distingue par:

- son utilisation intensive de la “ligne de commande”;
- sa modularité, qui rend le système très personnalisable;
- sa gestion des périphériques (“tout est un fichier”).

Nous allons maintenant voir comment GNU / Linux gère:

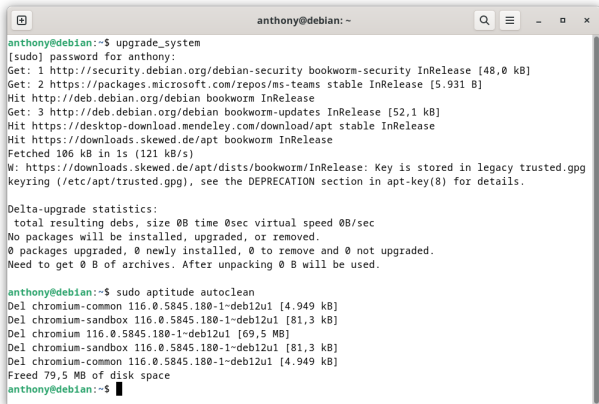
- les fichiers;
- les utilisateurs;
- les processus;
- les programmes;

Le terminal



Le terminal et la ligne de commande

- L'administration d'un système GNU / Linux se fait principalement à l'aide de la **ligne de commande**;
- On ouvre un **terminal** dans lequel on écrit des **commandes** pour réaliser les tâches voulues;



```
anthony@debian:~$ sudo upgrade_system
[sudo] password for anthony:
Get: 1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Get: 2 https://packages.microsoft.com/repos/ms-teams stable InRelease [5.931 B]
Hit http://deb.debian.org/debian bookworm InRelease
Get: 3 http://deb.debian.org/debian bookworm-updates InRelease [52,1 kB]
Hit https://desktop-download.mendeley.com/download/apt stable InRelease
Hit https://downloads.skewed.de/apt bookworm InRelease
Fetched 106 kB in 1s (121 kB/s)
W: https://downloads.skewed.de/apt/dists/bookworm/InRelease: Key is stored in legacy trusted.gpg
keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.

Delta-upgrade statistics:
 total resulting debs, size 0B time 0sec virtual speed 0B/sec
No packages will be installed, upgraded, or removed.
0 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B of archives. After unpacking 0 B will be used.

anthony@debian:~$ sudo aptitude autoclean
Del chromium-common 116.0.5845.180-1-deb12u1 [4.949 kB]
Del chromium-sandbox 116.0.5845.180-1-deb12u1 [81,3 kB]
Del chromium 116.0.5845.180-1-deb12u1 [69,5 MB]
Del chromium-sandbox 116.0.5845.180-1-deb12u1 [81,3 kB]
Del chromium-common 116.0.5845.180-1-deb12u1 [4.949 kB]
Freed 79,5 MB of disk space
anthony@debian:~$ █
```

Le terminal et la ligne de commande

- Avantages:
 - ✓ administration de systèmes à distance (via `ssh` par exemple);
 - ✓ programmation de scripts de maintenance;
 - ✓ uniformité: les commandes fonctionnent sur la plupart des systèmes malgré leurs différences;
 - ✓ flexibilité: beaucoup de moyens différents d'effectuer une même tâche;
- × Inconvénient: il faut apprendre toutes ces commandes; heureusement, des manuels très complets existent sous GNU / Linux;

Résumé des commandes



Nous verrons ensemble de nombreuses commandes; pensez à rédiger un résumé pour vous souvenir des noms des commandes et de leurs effets!

Commandes, commandes, commandes, ...

- GNU propose de très nombreux petits programmes réalisant des tâches parfois très simples;
- On combine très souvent ces outils pour obtenir des résultats de plus en plus sophistiqués;
- Il y a donc souvent beaucoup de façons différentes de réaliser une tâche particulière;

Options, options, options, ...

- Les commandes basiques possèdent des options parfois très nombreuses;
- Elles permettent de modifier le comportement du programme;
- Si la commande que vous utilisez ne fait pas exactement ce que vous voulez, consultez le manuel pour trouver l'option qui vous arrange (slide suivant);
- La plupart des programmes GNU permettent de combiner les options de manière concise: `ls -lSh` au lieu de `ls -l -S -h`
- En général, l'ordre des options n'importe pas;

man: le manuel

- Doit-on vraiment apprendre toutes ces commandes et leurs options par cœur?
- Heureusement non! Si l'on a un trou de mémoire, on peut consulter le manuel;
- Pour tout savoir sur la commande voulue, taper `man` commande dans un terminal.
- Si `man` ne fonctionne pas, essayez `help`;



Il est utile de mémoriser ce que font les commandes, mais pas leurs options dans le moindre détail.

Scripts *shell*

- Le **shell** est le programme que l'on utilise pour taper nos commandes;
- On peut changer de shell dans un même terminal;
- On peut aller jusqu'à programmer des **scripts**, de petits programmes exécutés par le *shell* qui nous permettront d'administrer notre système;

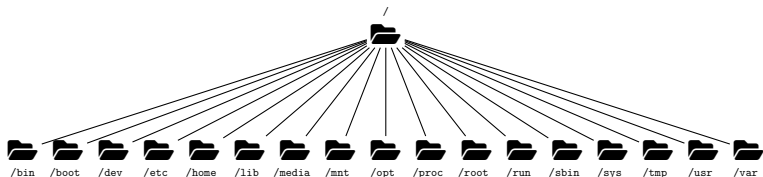
Fichiers

“(Presque) Tout est un fichier”

- Sous GNU / Linux, (presque) tout est un fichier: les fichiers bien sûr, mais aussi:
 - les répertoires;
 - les liens;
 - les périphériques . . . y compris les disques durs!
 - les entrées / sorties;
 - la mémoire;
 - . . .
- Avantage: traitement unifié de beaucoup d'objets très différents;

Structure du système de fichiers

Le **système de fichiers** structure les données sur le(s) disque(s).
C'est une **arborescence** qui suit les conventions Unix:



- / est la **racine** (à peu près comme C:\ sous Windows);
- Le caractère / sépare les répertoires (= \ sous Windows);
- Les majuscules et minuscules importent (*case sensitivity*);

Quelques répertoires importants

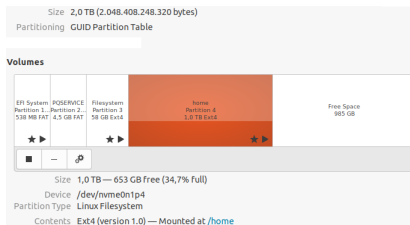
- `.` est le répertoire actuel;
- `..` est le répertoire **parent**;
- `/dev` (pour **devices**) contient le matériel (disques durs, processeurs, ...);
- `/etc` contient les fichiers de configuration globaux;
- `/home` contient les répertoires personnels des utilisateurs;
- `/mnt` et `/media` contiennent les disques “montés”;
- `/tmp` contient des fichiers temporaires: il est vidé à chaque redémarrage;
- `/var` contient diverses données (en particulier des “logs” dans `/var/logs`);

Disques et points de montage

- Les disques durs sont des fichiers situés dans `/dev`;
- Pour avoir accès à leur contenu, il faut les **monter**, c'est-à-dire les accrocher à un répertoire appelé **point de montage**;

Partitions

- En réalité, on ne monte pas directement un disque mais bien ses **partitions**;
- Tout disque est **partitionné**: une **partition** est un morceau du disque sur lequel se trouve un **système de fichiers**;






- Lorsqu'on veut accéder au contenu d'une partition, on doit monter cette partition — **pas le disque**;
- Le fichier `/etc/fstab` contient les informations de montage sur les périphériques;

Identifier les points de montage

- Pour savoir ce qui est monté et où, on peut utiliser la commande `mount`;
- Cela peut donner beaucoup de résultats;
- Si l'on veut avoir des informations sur un point de montage, on peut utiliser la commande `findmnt /point/de/montage`;

Types de systèmes de fichiers

- De très nombreux types de systèmes de fichiers existent, avec des performances différentes;
- En pratique, on retrouve surtout:
 -  apfs, hfsplus;
 -  ext2, ext3, ext4;
 -  fat, vfat, ntfs;

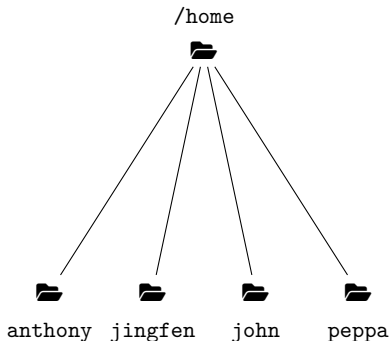


Pour pouvoir lire vos fichiers “n'importe où”, utilisez vfat ou ntfs pour vos clés USB.


Utilisateurs

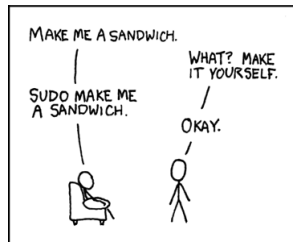
Utilisateurs

- GNU est un système **multi-utilisateurs**: plusieurs utilisateurs peuvent s'y connecter en même temps et partager des ressources;
- Chaque utilisateur a un nom (ex: anthony) et un répertoire personnel dans /home (ex: /home/anthony/);
- Le système peut théoriquement accueillir 2^{32} utilisateurs;



Super-utilisateur

- Pour des raisons de sécurité, un utilisateur ordinaire ne peut pas tout faire;
- Seul le **super-utilisateur** peut:
 - installer / supprimer des programmes;
 - accéder à des fichiers “dangereux”;
 - ...
-  La commande `sudo` permet d'exécuter une commande en tant que super-utilisateur;



<https://xkcd.com/149>

Utilisateurs

- Les utilisateurs sont identifiés par un nom et un UID (**u**ser **i**dentifier);
- Le super-utilisateur root a toujours l'UID 0;
- Il y a beaucoup d'utilisateurs "spéciaux";

Exemple (fichier /etc/passwd)

Le fichier /etc/passwd contient les utilisateurs:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
...
anthony:x:1234:1234:Anthony Labarre,,,:/home/anthony:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
...
```

Groupes

- Les groupes sont identifiés par un nom et un GID (**group identifier**);
- Ils servent à définir des **permissions** (voir plus loin) de manière plus globale et simple; par exemple:
 - faire partie de `audio` permet d'utiliser les périphériques audio;
 - faire partie de `lpadmin` permet de configurer les imprimantes;
 - faire partie de `sudo` permet d'utiliser `sudo` (!);
 - ...
- La commande `groups` affiche les groupes dont vous faites partie;
- Chaque utilisateur fait partie de son propre groupe;

Gestion des utilisateurs

Les commandes suivantes permettent de gérer les utilisateurs:

- `sudo adduser nom`: ajoute l'utilisateur avec le nom donné;
- `sudo deluser nom`: supprime l'utilisateur avec le nom donné;
 - son répertoire personnel n'est pas supprimé; utilisez l'option `--remove-home` ou `--remove-all-files` pour ce faire;
- `sudo usermod nom`: modifie l'utilisateur avec le nom donné;
- `passwd`: change le mot de passe de l'utilisateur actuel;

Gestion des groupes

Les commandes suivantes permettent de gérer les groupes:

- `sudo addgroup nom`: ajoute le groupe avec le nom donné;
- `sudo delgroup nom`: supprime le groupe avec le nom donné;
- `sudo groupmod nom`: modifie le groupe avec le nom donné;

Si l'on veut ajouter ou supprimer un utilisateur d'un groupe, on utilise `usermod`.

Permissions

- Des **permissions** sont associées à chaque fichier; elles déterminent ce que chaque utilisateur peut faire d'un fichier;
- Les trois permissions les plus courantes sont:
 - r (lecture);
 - w (écriture);
 - x (exécution);
- Si le fichier est un répertoire, x permet de le traverser;
- Un fichier exécutable se lance avec la commande `./fichier`;

Permissions: exemple avec `ls -l`

L'option `-l` de la commande `ls` affiche les informations détaillées:

Exemple

Lancer la commande `ls -l /var` donne le résultat suivant pour `/var/log`:

type	permissions				propriétaire	groupe	taille	dernière modification	nom
d	r	w	x	15	root	syslog	4096	oct 6 12:33	log

Les types les plus fréquents sont:

- `-` pour un fichier ordinaire,
- `d` pour un répertoire, ou
- `l` pour un lien.

Changer les attributs d'un fichier

- Les commandes suivantes permettent de modifier les propriétés d'un fichier:
 - `chmod`: change les permissions du fichier
 - `chown`: change le propriétaire et / ou le groupe du fichier
 - `chgrp`: change le groupe du fichier

Utilisation de `chmod`

`chmod` s'utilise comme suit:

```
chmod CATÉGORIE(S)±PERMISSION(S) fichier
```

avec: CATÉGORIE:

- u (pour **u**ser): propriétaire du fichier;
- g (pour **g**roup): groupe du fichier;
- o (pour **o**thers): les autres;
- a (pour **a**ll): tout le monde;

+ ajoute une permission (r, w, x), - la supprime.

Utilisation de chmod

Exemple

Modifions les permissions du fichier suivant:

```
$ ls -l fichier
```

```
----- 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod u+r fichier; ls -l fichier
```

```
-r----- 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod g+w fichier; ls -l fichier
```

```
-r---w---- 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod o+x fichier; ls -l fichier
```

```
-r---w---x 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod a+r fichier; ls -l fichier
```

```
-r--rw-r-x 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod ug-r fichier; ls -l fichier
```

```
-----w-r-x 1 anthony anthony 0 oct 31 03:33 fichier
```

Changer les attributs d'un fichier

Il existe également des codes correspondant aux différentes permissions:

code	signification	code	signification
0	---	4	r--
1	--x	5	r-x
2	-w-	6	rw-
3	-wx	7	rwX

Exemple

```
$ touch fichier_vider
$ chmod 123 fichier_vider
$ ls -l fichier_vider
---x-w--wx 1 anthony anthony 0 sep  2 11:27 fichier_vider
```

Processus

Processus

- Un **processus** est une instance d'un programme en cours d'exécution;
- Comme les utilisateurs, les processus sont identifiés par des numéros: les PID (**p**rocess **i**dentifier);

Lancement des programmes à partir d'un terminal

- On peut lancer un programme à partir du terminal;
- C'est utile pour voir les messages d'erreurs;
- Attention, on perd parfois l'accès au terminal!
- Solution: `programme &`

Voir les processus: `ps`

- La commande `ps` permet d'obtenir la liste des processus actifs;
- En général, on l'invoque de la façon suivante: `ps -aux`;
- Ceci donne la liste de tous les processus actifs, avec leur propriétaire et beaucoup d'autres informations;

Surveiller les processus: top

- La commande `top` donne aussi la liste des processus, mais en les “surveillant”;
- La liste est mise à jour en temps réel selon l’activité des processus;
- C’est utile entre autres pour savoir “qui” consomme le processeur ou la mémoire;

Tuer les processus

- Si un processus est trop consommateur, on peut le “tuer” (l'arrêter);
- Deux moyens:
 - ① `kill -9 PID;`
 - ② `pkill -9 nom_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;
 - ...
- `kill -l` donne la liste des signaux disponibles;

Codes de retour

- Quand un processus se termine, il renvoie un **code de retour**;
- C'est un nombre entier qui permet de vérifier si tout s'est bien passé;
- Le code de retour du dernier processus qui s'est terminé se trouve dans la variable `?`, dont on affiche le contenu avec `echo $?;`





Le code de retour “normal” est 0 et indique que tout s'est bien passé (= “0 problème”).

Variables d'environnement

Variables d'environnement

- Les **variables d'environnement** sont des variables utilisées par le système pour effectuer diverses tâches;
- Elles sont toutes en majuscules; par exemple:
 - HOME: le répertoire personnel de l'utilisateur actuel;
 - USERNAME: le nom de l'utilisateur actuel;
 - PATH: des chemins utiles au système pour exécuter des commandes;
 - ...
- Pour afficher leur contenu, on utilise la commande `echo "$VARIABLE"`;
- La commande `printenv` les affiche toutes;

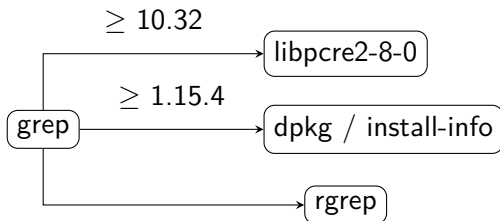
Variables d'environnement

- Pour obtenir la liste des variables d'environnement disponibles, il suffit de taper \$ suivi de la touche  deux fois;
- De manière générale, la touche  sert dans le terminal à compléter automatiquement le texte;
- Pour modifier la valeur d'une variable d'environnement, on utilise la commande `export VARIABLE=VALEUR` (pas d'espaces autour de "="!);

Administration basique

🌀 Gestion des programmes sous Ubuntu

- Les programmes sont gérés sous forme de **paquets** avec l'extension `.deb`;
- Ces paquets contiennent les informations sur les **dépendances**; par exemple, pour le programme **grep**:



- Cela permet de les installer automatiquement avec le programme désiré;

Le système APT

- APT (pour **A**dvanced **P**ackage **T**ool) est une collection d'outils gérant les programmes;
- On y trouve:
 - `apt-get`: (dés)installation de programmes ou paquets;
 - `apt-cache`: recherche de programmes ou paquets;
 - `apt-file`: recherche de fichiers dans des paquets;
- D'autres outils utiles sont basés sur apt (notamment `aptitude`)

⋮

Fonctionnement de apt-get

- apt-get utilise des **sources**, spécifiées dans le fichier `/etc/apt/sources.list`;
- Chaque source permet d'accéder à une liste de programmes;
- On les configure avec la syntaxe:
deb URL composantes
- S'il vous faut des programmes qui ne sont pas disponibles sous Ubuntu 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

- Pour installer le paquet monpaquet:
`sudo apt-get install monpaquet`
- Pour supprimer le paquet monpaquet:
`sudo apt-get remove monpaquet`
`sudo apt-get purge monpaquet` retire aussi les fichiers de configuration
- Si on veut trouver le nom d'un paquet:
`apt-cache search mots-clés`

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;
- Comment trouver les dépendances à satisfaire?
- Solution: utiliser `apt-file`

Exemple

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

Mises à jour

- ① On doit mettre à jour la base de données:

```
sudo apt-get update
```

- ② Et ensuite utiliser:

- `sudo apt-get upgrade mon_paquet` (met à jour `mon_paquet`),
ou
- `sudo apt-get upgrade` (met à jour **tous** les paquets), ou
- `sudo apt-get dist-upgrade` (comme `upgrade` mais retire parfois des paquets pour satisfaire des dépendances);

Visualisation de texte

- `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 10 premières lignes d'un fichier;
- `tail fichier` affiche les 10 dernières lignes d'un fichier;

Éditeurs de texte

- Deux éditeurs très complets et très connus: `vi` et `emacs`;
- Ils sont assez complexes à utiliser pour un débutant;
- Nous allons donc utiliser `nano` à la place, qui suffira (et nous évitera aussi de choisir un camp);
- Il existe aussi des éditeurs de texte non-interactifs (`sed`, `awk`, ...) dont nous parlerons si nous avons le temps;

Devoir pour la fois prochaine

Installez Ubuntu sur VirtualBox!

(suivez les instructions dans la vidéo sur la page de cours)

Si vous avez des problèmes, posez vos questions **avant la séance de TP**. Fournissez des informations (screenshots des messages d'erreur, etc.).