

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 ;

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` ;

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 ;

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 ;
 - ...

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 -1` 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” !).

Codes de retour

- 1 - Catchall for general errors
- 2 - Misuse of shell builtins (according to Bash documentation)
- 126 - Command invoked cannot execute
- 127 - “command not found”
- 128 - Invalid argument to exit
- 128+n - Fatal error signal “n”
- 130 - Script terminated by Control-C
- 255* - Exit status out of range

Variables d'environnement

Variables d'environnement

- Les **variables d'environnement** sont des variables utilisées par le système pour effectuer diverses tâches ;

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 :

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 ;

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 ;

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 ;

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 ;
 - ...



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"` ;

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` ;

Gestion des programmes sous Ubuntu

- Les programmes sont gérés sous forme de **paquets** avec l'extension `.deb` ;
- On les (dés)installe dans le terminal (voir slide suivant) ;

Gestion des programmes sous Ubuntu

- Les programmes sont gérés sous forme de **paquets** avec l'extension `.deb` ;
- On les (dés)installe dans le terminal (voir slide suivant) ;
- Avantages :

Gestion des programmes sous Ubuntu

- Les programmes sont gérés sous forme de **paquets** avec l'extension `.deb` ;
- On les (dés)installe dans le terminal (voir slide suivant) ;
- Avantages :
 - mises à jour centralisées ;

Gestion des programmes sous Ubuntu

- Les programmes sont gérés sous forme de **paquets** avec l'extension `.deb` ;
- On les (dés)installe dans le terminal (voir slide suivant) ;
- Avantages :
 - mises à jour centralisées ;
 - les **dépendances** (= les paquets supplémentaires nécessaires) sont installées automatiquement ;

Gestion des programmes sous Ubuntu

- Les programmes sont gérés sous forme de **paquets** avec l'extension `.deb` ;
- On les (dés)installe dans le terminal (voir slide suivant) ;
- Avantages :
 - mises à jour centralisées ;
 - les **dépendances** (= les paquets supplémentaires nécessaires) sont installées automatiquement ;
- La suite de programmes `apt` (**a**dvanced **p**ackage **t**ool) permet de gérer tout ce qui concerne ces paquets ;

Fonctionnement de apt-get

- apt-get utilise des **sources**, spécifiées dans le fichier `/etc/apt/sources.list` ;

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 ;

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`

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 :

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

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

- Pour installer le paquet monpaquet :
`sudo apt-get install monpaquet`

Installation

- 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

Installation

- 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 ;

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 ;
- Comment trouver les dépendances à satisfaire ?

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`

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

- On doit mettre à jour la base de données :
`sudo apt-get update`
- 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 ;