

Roberto Medina

Ubuntu Party – 13.10

16 et 17 Novembre 2013

- Thèmes abordés:
  - Un peu d'histoire.
  - Premières commandes.
  - Système de fichiers, droits.
  - Lancer de programmes.
  - Gestion des processus.
  - Gestionnaire de paquets.
  - Démarrage des services.

- Qu'est-ce qu'un **Shell**?
  - Une interface (logicielle) avec la machine.
  - Permet l'interaction avec la machine à travers des commandes.
  - Utilisé avant l'invention de la souris, jusqu'à nos jours à cause de son efficacité.
- Exemples de Shell
  - Bourne Shell (**sh**).
  - Bourne Again Shell (**bash**) – Installé par défaut dans plusieurs distributions, dont Ubuntu.
  - Z Shell (**zsh**) – Populaire entre les utilisateurs plus expérimentés

- Un **terminal** vous permet d'utiliser des commandes shell et lancer des programmes
  - Pour ouvrir un terminal: `<Ctrl + Alt + T>` sous Unity.
  - Il existe plusieurs types de terminales, en fonction de votre environnement de bureau
    - gnome-terminal sous Unity, Cinnamon et Gnome
    - konsole sous KDE
    - Il y a d'autres options comme: xterm, terminator, finalterm...

```
user@hôte: ~ $ echo 'hello world!'
```

- Se décompose en 4 parties
  - **user**: Le nom d'utilisateur
  - **hôte**: Le nom de la machine
  - **~** : Désigne le répertoire courant
  - **\$** : pour identifier le type d'utilisateur
    - **#** : indique qu'il s'agit d'un super-utilisateur
    - **%** : sous zsh

- **commande -option1 -option2 ... argument1 argument2 ...**
- **ls** : pour lister le contenu d'un répertoire.
  - **ls** : tout seul va lister le contenu du répertoire courant.
  - On peut utiliser des options pour chaque commande, identifiés par un tiret.
  - **ls -a** : montre les fichiers cachés. Càd tous ceux qui ont un '.' au début.
  - **ls -l** : montre les détails des fichiers du répertoire.
  - **ls <répertoire>** : liste les fichiers du répertoire en question.
- **cd <répertoire>** : pour changer de répertoire (on y reviendra après...)
- **mkdir** : pour créer un répertoire.
- Attention! Les commandes doivent respecter les minuscules ou les majuscules
  - **ls -r** ce n'est pas la même chose que **ls -R**

- Pour éviter les erreurs de frappe et pour aller plus vite utiliser la touche Tab (tabulation) qui permet de compléter le nom d'un fichier, d'une commande ou d'un programme.
  - Quand vous voulez compléter le nom d'un fichier
    - Ex: **cd Doc**<Tab> \*ici j'utilise la tabulation\*  
Me complète la commande à **cd Documents**
  - S'il y a plusieurs options après les caractères tapés on utilise une double tabulation pour lister les options possibles
    - Ex: **cd D**<Tab><Tab> va lister tous les répertoires qui commencent par D dans le répertoire courant

# Premières commandes (1) : exemple



```
vov@Hubble ~ > ls
Android  Documents  Dropbox    Music      Public     Videos    workspace
Desktop Downloads  Git        Pictures   Ubuntu One VirtualBox VMs
vov@Hubble ~ > mkdir UbuntuParty
vov@Hubble ~ > ls
Android  Documents  Dropbox    Music      Public     UbuntuParty VirtualBox VMs
Desktop Downloads  Git        Pictures   Ubuntu One Videos      workspace
vov@Hubble ~ > cd UbuntuParty/
vov@Hubble UbuntuParty > █
```



- **rm** : pour supprimer des fichiers
  - **rm -r** : pour supprimer des répertoires et leur contenu
    - -r veut dire récursif.
- **cp** : pour copier des fichiers
  - **cp <fichier(s) à copier> <destination>**
    - Si la destination est un répertoire il copie le fichier dans le répertoire, sinon si c'est un fichier le contenu est remplacé par celui de la source.
    - On peut copier plusieurs fichiers à la fois, le dernier argument est la destination.
      - Ex: cp fichier1 fichier2 [...] fichierN repertoire1
  - **cp -r** : pour copier un répertoire.

- **mv** : pour déplacer des fichiers vers un répertoire.
  - Pareil que **cp**, le dernier argument est la destination.
- **echo**: va afficher une ligne de texte.
- **cat**: affiche le contenu d'un fichier.

- **man**: c'est la commande qui permet de consulter le manuel pour une commande ou un programme
  - Utilisation: **man <commande>**
  - Très important et utile! Permet la consultation des différentes options pour une commande. Peut inclure quelques exemples d'utilisation.
  - S'il y a plusieurs commandes (ou fonctions) avec le même nom, on les différencie grâce à des sections du manuel.
    - Ex: **man kill** affiche la page du manuel de la commande kill du Shell et **man 2 kill** affiche le manuel de la fonction kill de POSIX.
  - Pour quitter une page du manuel utiliser la touche <Q>.
  - Pour rechercher une commande: **man -k <mot clé>**
    - Ex: **man -k remove** va lister les différentes commandes qui contiennent le mot "remove" sur leur description
  - **man man** : permet consulter le manuel du manuel ;)

# Premières commandes (3): man (exemple)



```
vov@Hubble: man kill
KILL(1)
vov@Hubble: man kill 150x37
User Commands
KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-s signal|-p] [-q signal] [-a] [--] pid...
    kill -l [signal]

DESCRIPTION
    The command kill sends the specified signal to the specified process or process group. If no signal is specified, the TERM signal is sent. The TERM signal will kill processes which do not catch this signal. For other processes, it may be necessary to use the KILL (9) signal, since this signal cannot be caught.

    Most modern shells have a builtin kill function, with a usage rather similar to that of the command described here. The '-a' and '-p' options, and the possibility to specify processes by command name are a local extension.

    If sig is 0, then no signal is sent, but error checking is still performed.

OPTIONS
    pid... Specify the list of processes that kill should signal. Each pid can be one of five things:

        n       where n is larger than 0. The process with pid n will be signaled.

        0       All processes in the current process group are signaled.

        -1      All processes with pid larger than 1 will be signaled.

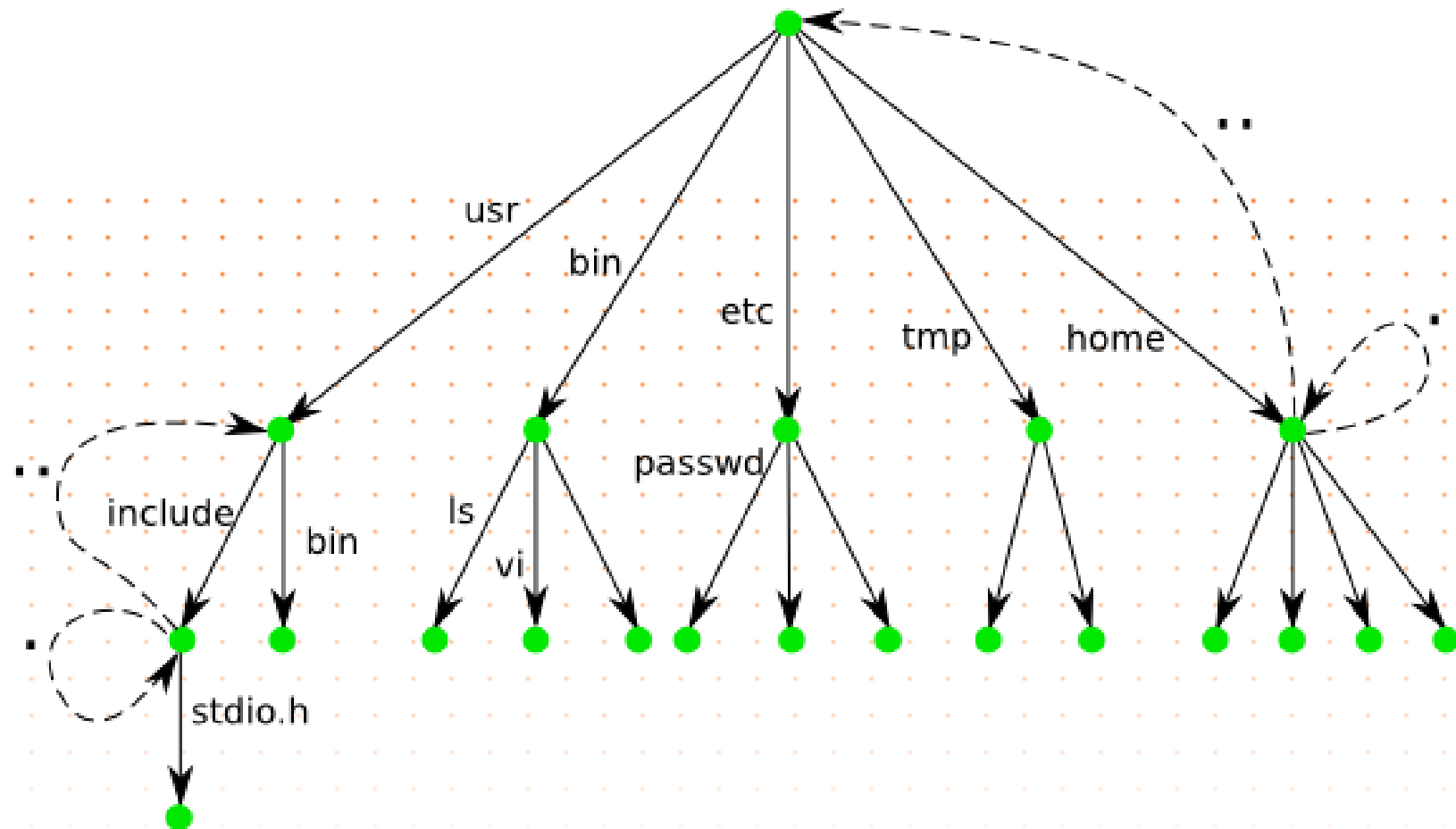
        -n       where n is larger than 1. All processes in process group n are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either the signal must be specified first, or the argument must be preceded by a '--' option, otherwise it will be taken as the signal to send.

    commandname
        All processes invoked using that name will be signaled.

Manual page kill(1) line 1 (press h for help or q to quit)
```

- Sous UNIX et donc sous Linux, TOUT est un fichier. Votre souris, clavier, écran... est un fichier.
- Il y a une arborescence qui est utilisée par le système
  - **/** : c'est la racine de tout système.
  - **/home** : c'est le dossier où se trouvent les données personnelles de chaque utilisateur.
  - **/etc** : répertoire où se trouvent les réglages de tous les programmes
  - **/bin** : où se trouvent les exécutables

- Retour sur **cd** : c'est la commande qui permet de se déplacer.
  - **cd /** : va nous emmener à la racine du système.
  - **cd** : sans paramètre nous emmène vers notre répertoire personnel.
    - **cd** est équivalent à **cd /home/utilisateur**
  - Le point simple désigne le répertoire courant. Deux points désignent le répertoire père.
    - **cd ../../..** : va me faire monter de trois répertoires.
- **pwd** : c'est la commande qui montre à quel endroit de l'arborescence on se situe.



- Retour sur ls: **ls -l** permet de lister les détails sur les fichiers du répertoire.
  - À gauche on retrouve les droits d'accès pour le propriétaire du fichier, le groupe d'utilisateur propriétaire du fichier et le reste des utilisateurs.
    - **d** (de Directory) indique que c'est un répertoire.
    - **r** (de Read) veut dire qu'on peut lire le fichiers.
    - **w** (de Write) veut dire qu'on peut écrire sur le fichier.
    - **x** (de eXecute) indique qu'on peut exécuter le fichier.



# Droits sur les fichiers (illustration) ubuntu®

```
drwxrwxr-x 5 vov plugdev 4096 Oct 31 22:38 Android/
drwxr-xr-x 2 vov plugdev 4096 Nov  9 14:30 Desktop/
drwxr-xr-x 9 vov plugdev 4096 Nov 10 21:58 Documents/
drwxr-xr-x 8 vov plugdev 4096 Nov 10 21:51 Downloads/
drwx----- 13 vov plugdev 4096 Nov 10 20:00 Dropbox/
drwxr-xr-x 3 vov users 4096 Nov  2 01:19 Git/
drwxr-xr-x 5 vov plugdev 4096 Nov 10 11:33 Music/
drwxr-xr-x 9 vov plugdev 4096 Nov 10 21:42 Pictures/
drwxr-xr-x 2 vov plugdev 4096 Jul 13 14:34 Public/
drwxrwxr-x 2 vov plugdev 4096 Jul 16 10:52 Ubuntu\ One/
drwxr-xr-x 3 vov users 4096 Nov 10 21:16 UbuntuParty/
drwxr-xr-x 5 vov plugdev 4096 Nov  4 00:50 Videos/
drwxr-xr-x 3 vov users 4096 Oct 20 23:06 VirtualBox\ VMs/
drwxrwxr-x 4 vov plugdev 4096 Nov  3 18:25 workspace/
```

- **chmod** : vous permet changer les droits pour un fichier.
  - Deux modes d'utilisation: le mode octal et le mode avec des caractères.
    - Ex mode octal: **chmod 755 mon\_fichier** va donner tous les droits à "mon\_fichier" pour l'utilisateur, les droits d'écriture et d'exécution pour les utilisateur de mon groupe et pour tous les autres utilisateurs.
    - Ex mode avec caractères: **chmod u+rw mon\_fichier** donne tous les droits à l'utilisateur sur "mon\_fichier".  
**chmod +x mon\_fichier** : donne le droit d'exécution à tous les utilisateurs  
**chmod -x mon\_fichier** : enlève le droit d'exécution à tous les utilisateur.
    - **chmod <droits> -R <nom\_repertoire>** : pour changer les droits sur un répertoire.

- Pareil que pour les commandes, peuvent recevoir des options avec des tirets, et des arguments.
  - Ex: **gedit fichier** va ouvrir l'éditeur texte gedit avec "fichier" comme paramètre.
  - Quand on lance un programme le terminal affiche un numéro, le PID (Process ID). Utile pour la gestion de processus (traité un peu plus loin).
  - Normalement le terminal ne rend pas la main, c'est à dire qu'on ne voit pas l'invite de commande reparaître
    - Il faut lancer le processus en arrière plan avec le symbole '&'
    - Ex: **gedit fichier &**

- Lancer **firefox** depuis le terminal, sans le '&'
  - Mais comment je récupère l'invite de commande?
- On a deux choix
  - Terminer le processus en lui envoyant un signal avec les touches <Ctrl+C>
  - Ou mettre en pause le processus en lui envoyant un signal avec les touches <Ctrl+Z> et en utilisant la commande **bg** (pour background)

- **ps** : liste les processus qui s'exécutent. A besoin d'options pour être lisible.
  - Ex: **ps -e** tous les processus      **pstree** une arborescence de processus (pas standard).
- **kill** : c'est la commande qui permet d'envoyer un signal à un processus.
  - **kill -Num\_du\_signal <PID\_programme>**
  - **kill -9 PID** : envoie le signal de terminaison (SIGINT) au processus identifié par PID (c'est violent, le processus se termine brutalement).
  - **killall <Nom\_programme>** : termine un programme en utilisant son nom.

# Gestion de processus : top

```
top - 17:55:43 up 38 min,  2 users,  load average: 0,50, 0,39, 0,34
Tasks: 189 total,   2 running, 187 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1,8 us,  0,4 sy,  0,0 ni, 97,6 id,  0,1 wa,  0,0 hi,  0,1 si,  0,0 st
KiB Mem:  8078908 total, 1892308 used,  6186600 free,   56904 buffers
KiB Swap:   0 total,    0 used,    0 free, 1085428 cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1997	vov	20	0	363m	64m	26m	S	2,0	0,8	1:14.27	compiz
1769	vov	20	0	65768	7120	4560	S	1,3	0,1	0:28.37	hud-service
1782	vov	20	0	150m	22m	10m	S	1,3	0,3	0:41.33	unity-panel-ser
1062	root	20	0	270m	92m	73m	S	1,0	1,2	2:17.76	Xorg
2009	vov	20	0	52228	9948	7688	S	1,0	0,1	0:29.05	indicator-multi
143	root	20	0	0	0	0	S	0,3	0,0	0:01.17	kworker/u16:5
1688	vov	20	0	5952	2668	856	S	0,3	0,0	0:09.65	dbus-daemon
1716	vov	20	0	4940	604	428	S	0,3	0,0	0:03.00	upstart-dbus-br
1847	vov	20	0	56552	4500	3576	S	0,3	0,1	0:10.43	indicator-appli
2210	vov	20	0	168m	20m	13m	S	0,3	0,3	0:06.38	gnome-terminal
2865	vov	20	0	863m	336m	43m	S	0,3	4,3	2:22.22	firefox
3204	vov	20	0	5340	1368	956	R	0,3	0,0	0:00.03	top
1	root	20	0	4044	2340	1340	S	0,0	0,0	0:01.74	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.60	ksoftirqd/0
4	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.10	migration/0
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0,0	0,0	0:02.46	rcu_sched
12	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	migration/1
13	root	20	0	0	0	0	S	0,0	0,0	0:00.47	ksoftirqd/1
14	root	20	0	0	0	0	S	0,0	0,0	0:01.16	kworker/1:0
15	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0H
17	root	rt	0	0	0	0	S	0,0	0,0	0:00.10	migration/2

- On peut installer, désinstaller, chercher, mettre à jour des programmes avec le terminal.
- **apt-get** c'est le gestionnaire de paquets utilisé par défaut dans Ubuntu.
- Nécessite des droits de super utilisateur (root).
  - La commande **sudo** permet de lancer un processus avec les droits de super utilisateur. Introduit par des raisons de sécurité.
    - Quand vous l'utilisez il va vous demander le mot de passe de root. Pas affiché dans le terminal par des raisons de sécurité.
    - Ex: **sudo cp /etc/bash.rc ~** copie le fichier de configuration de bash dans le répertoire personnel de l'utilisateur courant. Nécessite sudo car seulement le super utilisateur à droit de modifier ces fichiers.
- A une syntaxe un peu particulière



- **sudo apt-get update** : va mettre à jour la liste des paquets disponibles.
- **sudo apt-get upgrade** : va mettre à jour les paquets que vous avez installés.
  - Ex: on utilise **sudo apt-get update && sudo apt-get upgrade**, le '&&' veut dire qu'on va exécuter la mise à jour seulement si on réussit à récupérer la liste des nouveaux paquets.
- **sudo apt-get install <programme\_à\_installer>** : pour installer des programmes
  - On peut installer plusieurs programmes à la fois.
    - Ex: **sudo apt-get install programme1 programme2 ... programmeN**



# Gestionnaire de paquets (2) (exemple)



```
vov@Nova ~ % sudo apt-get update && sudo apt-get upgrade
Ign http://fr.archive.ubuntu.com saucy InRelease
Ign http://fr.archive.ubuntu.com saucy-updates InRelease
Ign http://security.ubuntu.com saucy-security InRelease
Ign http://archive.canonical.com saucy InRelease
Ign http://extras.ubuntu.com saucy InRelease
Ign http://dl.google.com stable InRelease
Ign http://ppa.launchpad.net saucy InRelease
Ign http://fr.archive.ubuntu.com saucy-backports InRelease
Get:1 http://security.ubuntu.com saucy-security Release.gpg [933 B]
Hit http://archive.canonical.com saucy Release.gpg
Hit http://fr.archive.ubuntu.com saucy Release.gpg
Hit http://extras.ubuntu.com saucy Release.gpg
Ign http://ppa.launchpad.net saucy InRelease
Hit http://fr.archive.ubuntu.com saucy-updates Release.gpg
Hit http://dl.google.com stable Release.gpg
Hit http://archive.canonical.com saucy Release
Hit http://fr.archive.ubuntu.com saucy-backports Release.gpg
Get:2 http://security.ubuntu.com saucy-security Release [49,6 kB]
Hit http://extras.ubuntu.com saucy Release
Ign http://ppa.launchpad.net saucy InRelease
Hit http://fr.archive.ubuntu.com saucy Release
Hit http://dl.google.com stable Release
Ign http://ppa.launchpad.net saucy InRelease
Hit http://fr.archive.ubuntu.com saucy-updates Release
Hit http://fr.archive.ubuntu.com saucy-backports Release
Hit http://archive.canonical.com saucy/partner Sources
Hit http://ppa.launchpad.net saucy Release.gpg
Hit http://extras.ubuntu.com saucy/main Sources
Hit http://fr.archive.ubuntu.com saucy/main Sources
Hit http://archive.canonical.com saucy/partner i386 Packages
Hit http://ppa.launchpad.net saucy Release.gpg
Hit http://fr.archive.ubuntu.com saucy/restricted Sources
Hit http://extras.ubuntu.com saucy/main i386 Packages
Hit http://dl.google.com stable/main i386 Packages
```

- **sudo apt-get remove <paquet>** : va désinstaller un paquet, mais pas ses fichiers de configuration.
  - Utile si on veut réinstaller le programme après.
- **sudo apt-get purge <paquet>** : fait la désinstallation et enlève les fichiers de configuration.
- **sudo apt-get autoremove** : pour enlever les paquets orphélins, c'àd ce qui sont installés pour rien.
- **sudo apt-get autoclean** : pour nettoyer le cache de apt-get.

- **apt-cache search <mot-clé>** : pour chercher des paquets qui sont installables avec apt-get.
- Le gestionnaire **aptitude** a la même plus ou moins la même syntaxe. Consulter le man.
  - **sudo apt-get install aptitude**

- **apt-cache search <mot-clé>** : pour chercher des paquets qui sont installables avec apt-get.
- Le gestionnaire **aptitude** a la même plus ou moins la même syntaxe. Consulter le man.
  - **sudo apt-get install aptitude**

- Plutôt utile quand on veut administrer un serveur.
- **/etc/init.d** : contient tous les services qui peuvent être activés au démarrage.
  - Ex: gestion d'Apache
    - **sudo /etc/init.d/apache2 start**
    - **sudo /etc/init.d/apache2 stop**
    - Pour des versions plus récentes on peut aussi utiliser l'alias:  
**sudo apache2 start**  
**sudo apache2 restart**

- Ubuntu utilise Upstart pour faire le démarrage des services
  - **initctl list** : permet de lister tous les services qui tournent sur la machine
  - **sudo initctl stop <nom\_du\_service>** : pour arrêter un service
    - Ex: sudo initctl stop bluetooth
  - **sudo initctl start <service>** : pour démarrer un service

- Ubuntu utilise Upstart pour faire le démarrage des services
  - **initctl list** : permet de lister tous les services qui tournent sur la machine
  - **sudo initctl stop <nom\_du\_service>** : pour arrêter un service
    - Ex: sudo initctl stop bluetooth
  - **sudo initctl start <service>** : pour démarrer un service

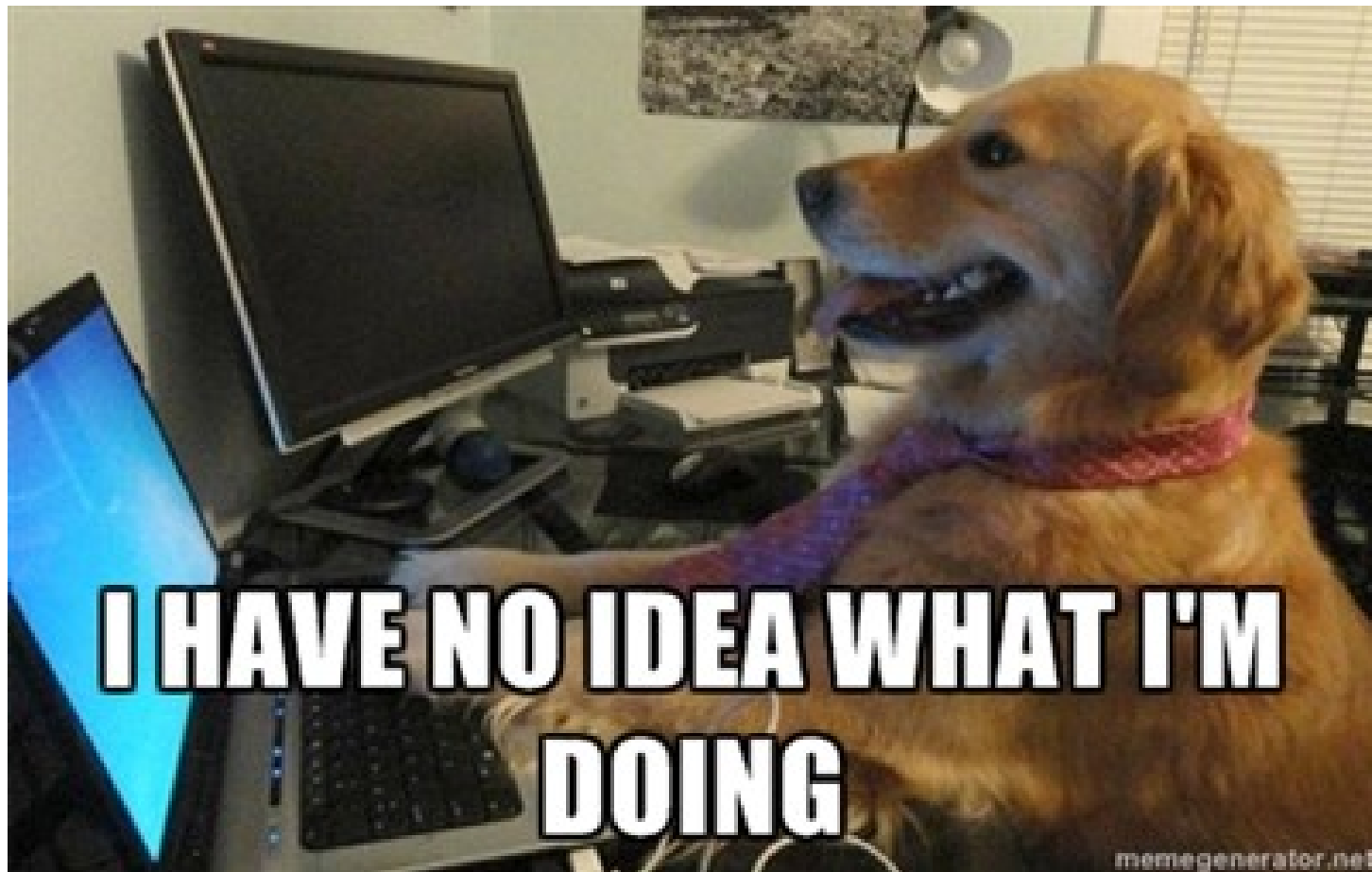
- Les boucles
  - **for, while, case, if**
- Manipulation des fichiers
  - **chown, ln, mount, umount, df**
- Recherche
  - **find, grep**
- Communication
  - **ssh, telnet, scp, wget**



- Bases sur les commandes UNIX
  - <http://perso.univ-lyon1.fr/thierry.excoffier/COURS/COURS/UNIX2/unix.html>
- Quelques cours des années précédentes
  - [http://cgit.thetys-retz.net/cli\\_lesson/](http://cgit.thetys-retz.net/cli_lesson/)
  - [http://git.thetys-retz.net/cli\\_lesson/](http://git.thetys-retz.net/cli_lesson/)

# Avez-vous des questions?

ubuntu®



# À ne pas rater juste après...

ubuntu<sup>®</sup>

- 13h30: Jouer sous Ubuntu (démonstration)