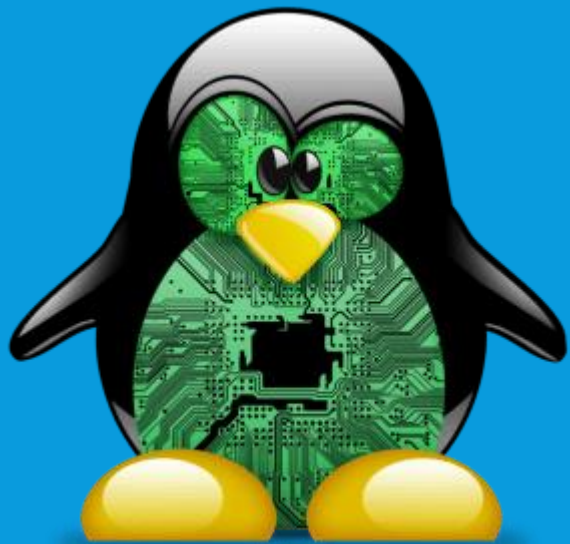


UNIVERSITÉ DE FRANCHE-COMTÉ





GNU/Linux

Créé en 1991 Par Linus Torvalds

C'est un **système d'exploitation** réunissant le noyau **Linux** et le **système GNU**

Il est **Libre** et basé sur **UNIX**

Il est présent sur les **super-calculateurs**, en version **embarqué**, sur des **serveurs** mais aussi sur des **postes de travail**

Il existe de nombreuses version qui se présente sous forme de **Distributions**



Environ 90% des serveurs fonctionnent sous Linux

Deux modes d'utilisations

Interface graphique

Plus simple

Moins puissante, change d'une distribution à l'autre, pas toujours disponible,



Ligne de commandes

Plus Puissante, standard d'une distribution à l'autre, toujours présente

Moins facile a prendre en main

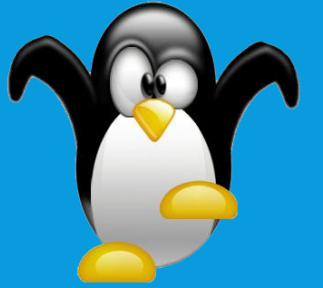
```
gerald@2017-00021-P: /mnt/c/Users/ggeniaut
root@orwell:/var/www# df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          1.5G  435M  997M  31% /
udev            10M    0   10M   0% /dev
tmpfs           397M  236K  396M   1% /run
/dev/mapper/vg_virtuel-root 1.5G  435M  997M  31% /
tmpfs           5.0M    0   5.0M   0% /run/lock
tmpfs           793M   12K  793M   1% /run/shm
/dev/sda1       228M   33M  183M  16% /boot
/dev/mapper/vg_virtuel-home 1.5G  178M  1.2G  13% /home
/dev/mapper/vg_virtuel-tmp  357M   11M  328M   3% /tmp
/dev/mapper/vg_virtuel-usr  4.1G   2.2G  1.7G  57% /usr
/dev/mapper/vg_virtuel-var  48G   39G   6.8G  86% /var
root@orwell:/var/www# df -h /dev/sda
sda  sda1  sda2  sda3  sda5
root@orwell:/var/www# df -h /dev/sda
Filesystem      Size  Used Avail Use% Mounted on
udev            10M    0   10M   0% /dev
root@orwell:/var/www#
```

Le mode console ou la ligne de commande



```
1: "/pci/@d/pci-ata@1/ata-400/@0:10,\mach_kernel", Waiting on 0
IOPathMatch</key><string ID="1">IODeviceTree:/pci/@d/pci-ata@1
ring></dict>
reWire GUID = 0x50e4ff:0
ent:0
device = IOService:/GossamerPE/pci@800000000/AppleGracklePCI
PCIBridge/pci-ata@1/CMD646Root/ata-400/CMD646ATA/ATADeviceK
eDriver/IOATABlockStorageDevice/IOBlockStorageDriver/ST34000
itions
Untitled_3010 14 minor 9 with uid=0 audit-uid=1
```

Les indispensables



Lister un répertoire (ls)

Ex : **ls** /home/myname

Changer de répertoire (cd)

Ex : **cd** /home/myname

Créer un répertoire (mkdir)

Ex : **mkdir** /home/myname/nouveau_dossier

Supprimer un fichier (rm)

Ex : **rm** /home/myname/fichier.txt

Copier un fichier (cp)

Ex : **cp** /usr/fichier /tmp/fichier_copie

Déplacer un fichier (mv)

Ex : **mv** /ancien/fichier /nouveau/fichier

Toutes les commandes Linux disposent d'arguments qui peuvent compléter la commande de base

Un argument peut être une lettre, une lettre et une valeur.

Un argument est précédé d'un tiret (parfois deux)

Ex : Lister un répertoire en colone : **ls -l**

Ex : Afficher les partitions en « human readable » : **df -h**

RTFM

(Read The Fucking Manual)

Toutes les commandes Linux disposent d'une aide courte et d'un manuel détaillé

Aide courte : -h ou --help
Manuel d'aide : man

Ex : ls -h ou ls --help ou man ls



Linux propose de base dans ses bash, l'autocomplétion

Le système vous propose les commandes sur simple demande !

Tapez le début de votre commande et appuyez sur la touche « Tab »



Pour aller plus loin, **LE** mécanisme : la redirection de flux !

Linux permet de chainer les commandes, c'est-à-dire, d'utiliser la sortie d'une commande comme l'entrée d'une autre commande

Le pipe !



Exemple :

ls : permet de lister le contenu d'un dossier

grep : permet de vérifier des expressions régulières



Trouver lister les fichiers dont le nom contiennent « text » dans le repertoire courant :

ls -l | grep "text"

Commandes utiles

Créer un fichier vide : **touch**

Ex : **touch** /tmp/nouveau_fichier

lire un fichier : **cat**

Ex : **cat** /tmp/nouveau_fichier

Trouver les fichiers qui contiennent une chaine : **grep**

Ex : **grep** 'chaine' /tmp/nouveau_fichier

Lister les processus : **ps**

Ex : **ps -aux**

Afficher le contenu d'un fichier : **tail**

Ex : affiche les 20 dernière lignes **tail -n 20** /var/log/syslog

Ex : affiche en temps réel le fichier : **tail -f** /var/log/syslog

Faire un lien symbolique : **ln**

Ex : **ln -s** /chemin/Fichier /chemin/lien

Changer d'utilisateur : **su**

Ex : **su** utilisateur

Lister les ports réseaux : **netstat**

Ex : **netstat -pant**



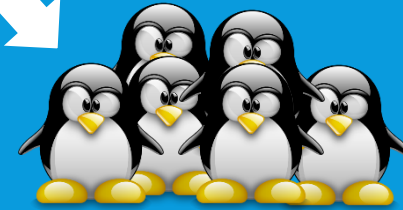
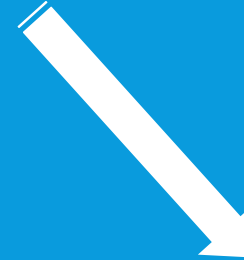
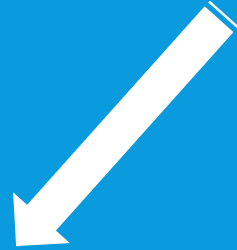
Linux est un système Multi-Utilisateur :



Plusieurs personnes peuvent utiliser l'ordinateur en même temps



Gestion des droits
+
Gestion des propriétés



Gestion des utilisateurs
+
Gestion des groupes

Droits et propriétés

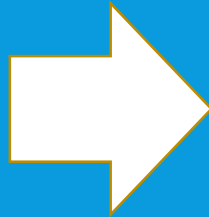


Les droits sont représentés par des chiffres. Il y a trois droits :

Lire – **R**(ead)

Ecrire – **W**(rite)

Exécuter – **X**(ecute)



4

2

1

On « compte » les permissions en additionnant

Exemples :

Lire et Ecrire : $4 + 2 = 6$

Lire, Ecrire et Exécuter : $4 + 2 + 1 = 7$



Les fichiers et les dossiers disposent de 3 catégories d'utilisateurs :

Dans l'ordre :

Propriétaire - Groupe - Tous les autres

Les droits sont définis pour chaque catégorie, de gauche à droite !



Exemples :

Propriétaire, lire, écrire et exécuter
Groupe, lire et exécuter
Autres, aucun
750

Propriétaire, lire et écrire
Groupe, lire
Autres, lire
644

Modification des droits : la commande chmod

La commande s'utilise de cette façon : **chmod** [options] droits /chemin/vers/fichier

[options] : Le plus souvent on utilise l'option -R pour appliquer les droits de manière récurrentes : dossier et sous dossiers

Exemple : modification des droits sur /home/test/fichier :

Lire, écrire et exécuter pour le propriétaire

Lire, exécuter pour le groupe

Lire pour les autres

chmod 754 /home/test/fichier



Les propriétaires ou l'appartenance pour un fichier

D'une manière général le propriétaire d'un fichier est l'utilisateur qui a créé le fichier, ainsi que son groupe.

➡ Il y a donc deux appartenances différentes pour un fichier :

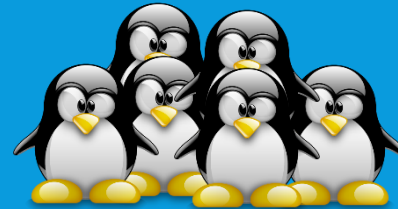
Utilisateur

■
■

Groupe



Droit pour une seule personne



Droit pour plusieurs personnes

Modification de l'appartenance : la commande chown

La commande s'utilise de cette façon : **chown** [options] propriétaire:groupe /chemin/vers/fichier

[options] : Le plus souvent on utilise l'option -R pour appliquer la modification de manière récurives : dossier et sous dossiers

Exemple : modification de l'appartenance sur /home/test/fichier :

On donne le fichier à l'utilisateur Eleve_X

On donne le fichier au groupe : Classe_Y

chown Eleve_X:Classe_Y /home/test/fichier



Les utilisateurs et les Groupes

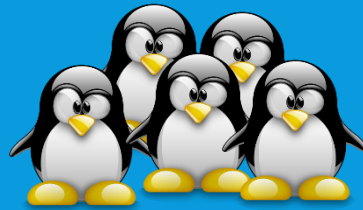
Linux conserve la liste des utilisateurs et des groupes dans deux fichiers, un pour les utilisateurs et un pour les groupes

`/etc/passwd`



Les utilisateurs

`/etc/group`



Les group

Un utilisateur c'est :

Un Login

Un mot de passe

Un **groupe**, *parfois plusieurs*

Un UID

Eventuellement un dossier



Un **utilisateur** peut appartenir à un, ou plusieurs **groupes**.

Les groupes permettent souvent de **gérer plus facilement** les droits d'accès sur un fichiers **pour plusieurs utilisateurs**

Quelques commandes pour manipuler les utilisateurs et groupes

Ajouter un utilisateur :
`useradd nouvel_utilisateur`

Ajouter un group :
`groupadd nouveau_groupe`

Supprimer un utilisateur
`deluser utilisateur`

Supprimer un groupe
`delgroup groupe`

Modifier le groupe principal d'un utilisateur
`usermod -g nouveau_groupe utilisateur`

Ajouter un groupe à un utilisateur
`usermod -a -G nouveau_groupe utilisateur`



Visualisation d'un utilisateur dans le fichier /etc/passwd

Dans le fichier des comptes utilisateurs, chaque ligne représente un utilisateur.
On y retrouve : le login:UID_user:UID_groupe_principal:Libélé du compte:dossier utilisateur

L'UID de l'utilisateur est UNIQUE sur le système.

D'un point de vue convention, on admet que les UID <1000 sont des comptes utilisateurs pour le système.

Ex : root a pour UID 0, daemon 1 etc...



Le fichier /etc/group

Dans ce fichier des groupes, chaque ligne représente un groupe.

On y retrouve : Nom_groupe:UID_groupe:login_utilisateur1,login_utilisateur2

Un groupe peut contenir de 0 à n utilisateurs

Chaque groupe dispose d'un UID unique sur l'ensemble du système.



Pour aller un peu plus loin, le fichier /etc/shadow

Nous avons dit qu'un utilisateur se caractérisait aussi par un mot de passe.
Ceux-ci sont stockés de manière crypté dans le fichier /etc/shadow

Dans ce fichier, chaque ligne représente un compte utilisateur et son mot de passe crypté.
Login:mot de passe crypté:autres infos



Ces trois fichiers passwd, group et shadow ne doivent en principe pas se modifier à la main.

On peut dans des cas précis le faire, mais il est préférable de gérer les utilisateurs et les groupes avec les commandes que l'on a vu précédemment.



Le compte Root et le groupe sudo



Root, c'est le super utilisateur.

Il a **tous les droits** sur tous les fichiers et tous les dossiers.

Son répertoire personnel est séparé des autres : **/root**

En ligne de commande, le fait d'être root est symbolisé par le signe **#**



N'utiliser qu'en cas de besoin !



SUDO

Il s'agit d'une commande, qui n'est pas forcément de base dans toutes les distribution linux (par ex, sous Debian il faut l'installer, alors que sous Ubuntu elle est présente par défaut)

Ce petit programme permet d'exécuter une commande avec les privilèges « root »

Ce programme s'accompagne d'un groupe appelé sudo (présent dans /etc/group et d'un fichier de configuration /etc/sudoers)

Exemple, **lister les fichiers dans /root/ :**

ls /root → Permission refusée

sudo ls /root/ → Affichage autorisé



SUDO

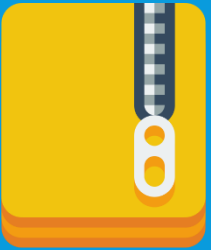
Le sudo, c'est un peu le pass-VIP des systèmes Unix. . .



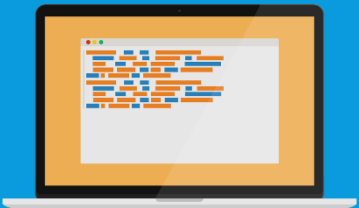
Installation & Mise à Jour



Sous Linux il existe plusieurs modes d'installation :



Téléchargement et désarchivage



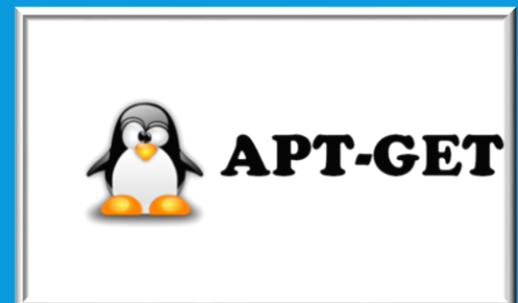
Compilation du code



Paquets et dépôts



Nous allons voir
ensemble le mode
« paquets et dépôts »



Linux repose sur des dépôts

Ces dépôts regroupent des paquets (logiciel prêt à être installés)
Chaque distribution a ses propres dépôts, souvent plusieurs
Il est possible d'ajouter d'autres dépôts
Les dépôts gèrent aussi les mises à jour

Trois grandes familles : Debian, RedHat et Slakeware
Ces grandes familles utilisent un gestionnaire de paquets différents.
Famille Debian (Ubuntu, Knopix, etc...) : APT
Famille RedHat (CentOS, Fedora, etc...) : YUM
Slakeware (SuSe, Slax, etc...) : YAST

Nous verrons ici que le gestionnaire de paquets **APT**



Mettre à jour



On utilise le gestionnaire de paquets

Le gestionnaire de paquets APT demande des droits Super Utilisateur (root ou sudo)

Mettre à jour la liste des logiciels disponibles dans les dépôts : `apt-get update`

Mettre à jour les logiciels installés : `apt-get upgrade`



Installer un logiciel



Chercher un logiciel (à partir d'une partie de son nom par exemple) : `apt-cache search bout_nom_logiciel`

Installer un logiciel (APT installera aussi les dépendances) : `apt-get install nom_logiciel`

Ex : Installation de apache :
`apt-cache search apache`
`apt-get install apache2`



Supprimer un logiciel



Désinstaller un logiciel, **apt-get remove *nom_logiciel***

Désinstaller un logiciel en supprimant les fichiers de configuration, **apt-get purge *nom_logiciel***

Désinstaller un logiciel et ses dépendances, **apt-get autoremove *nom_logiciel***



Editeur de texte

(ligne de commande)



Deux éditeurs inclus de base

VIM

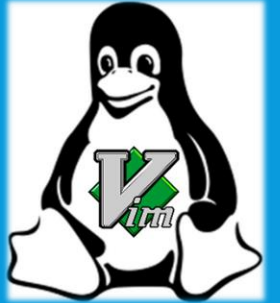


NANO

```
      :::  
iLE88Dj.  :jD88888Dj:  
.LGitE888D.f8GjjjL8888E;  
iE      :8888Et.      .G8888.  
;i      E888,          ,8888,  
        D888,          :8888:  
        D888,          :8888:  
        D888,          :8888:  
        D888,          :8888:  
        888W,          :8888:  
        W88W,          :8888:  
        W88W:          :8888:  
        DGGD:          :8888:  
                        :8888:  
                        :W888:  
                        :8888:  
                        E888i  
                        tW88D
```



Vi iMproved



Basé sur le logiciel *Vi*, *créé en 1976*

Il repose sur *un système de mode* (normal, commande, édition)

Un éditeur plutôt à *destination des programmeurs*

+

Ultra puissant
Hautement personnalisable
Extrêmement léger
Optimisé pour rester sur le clavier

-

Difficile à prendre en main
Long à apprendre

GNU Nano

```
      :::  
iLE88Dj. :jd88888Dj:  
.LGitE888D.f8GjjjL8888E;  
iE :8888Et. .G8888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W, :8888:  
DGGD: :8888:  
      :8888:  
      :W888:  
      :8888:  
      E888i  
      tW88D
```

Partie du projet GNU, **créé en 1999**

Pas de système de mode, utilise plutôt **des combinaisons de touches**

Un éditeur *pour débuter* ou faire de l'édition *occasionnelle*



+

Simple à apprendre
Relativement intuitif
Pensé pour interagir avec la souris

-

Peu puissant
Atteint rapidement ses limites