

Book Initiation — V0

Club\*NIX

<https://clubnix.fr>

# Contents

<b>1</b>	<b>Linux</b>	<b>2</b>
1.1	GNU-Linux . . . . .	2
1.1.1	UNIX . . . . .	2
1.1.2	GNU : "GNU's Not UNIX" . . . . .	2
1.1.3	GNU/LINUX . . . . .	3
1.2	different OS . . . . .	3
1.2.1	Differents OS . . . . .	3
1.2.2	differentes familles d'OS Linux . . . . .	4
<b>2</b>	<b>Language</b>	<b>4</b>
2.1	LaTeX . . . . .	4
2.2	Markdown . . . . .	4
2.2.1	Histoire . . . . .	4
2.2.2	Instruction . . . . .	5
<b>3</b>	<b>Terminal</b>	<b>5</b>
3.1	Terminal . . . . .	5
3.2	Bash . . . . .	5
3.2.1	Bash et scripts . . . . .	5
3.3	Oh My ZSH . . . . .	6
<b>4</b>	<b>Logiciel libre</b>	<b>6</b>
4.1	Git . . . . .	8
4.2	F-Droid . . . . .	8
<b>5</b>	<b>Infrastructure réseau</b>	<b>8</b>
5.1	LDAP . . . . .	8
5.1.1	Le LDAP, ou <i>Lightweigth Directory Acess Protocol</i> . . . . .	8
5.1.2	Introduction . . . . .	8
5.2	Le LDAP du club Nix ou de l'école . . . . .	8
5.2.1	Les entrées . . . . .	8
5.2.2	L'arborescence . . . . .	9
5.2.3	Chercher des informations d'un LDAP . . . . .	9
5.2.4	Les principales commandes du LDAP . . . . .	9
5.3	NFS . . . . .	10
5.3.1	Introduction . . . . .	10
5.3.2	Le NFS du Club*Nix ou de l'ESIEE . . . . .	10
5.3.3	Architecture d'un NFS . . . . .	10
5.3.4	NFS et VFS . . . . .	10
5.3.5	La sécurité du NFS . . . . .	11
5.4	Fichiers logs . . . . .	11
5.5	Virtual Machine . . . . .	11
5.6	Infra du club . . . . .	11

# 1 Linux

## 1.1 GNU-Linux

### 1.1.1 UNIX

**Unix** est un **systeme d'exploitation** multi-taches et multi-utilisateurs créé en **1969**. Il repose sur un interpreteur (le shell) et sur plusieurs petits utilisateurs (avec des actions spécifiques définies lors du code). C'est une **marque déposée de l'OpenGroup**. Son nom est un dérivé de "Unics" (*Uniplexed Information and Computing Service*). C'est un jeu de mot avec "Multics" (un autre noyau) qui vise à offrir simultanément plusieurs services à un ensemble d'utilisateur. Unix a donné naissance à une **famille de systemes** comme BSD, GNU/LINUX, iOS ou encore MacOS, eux-même divisés en variantes de système d'exploitation aux **normes POSIX**. Cette norme technique standardise les interfaces de programmation des logiciels. Il faut savoir que la quasi-totalité des systèmes pc ou mobiles (à l'exception des Windows) sont basés sur le noyau de UNIX (y compris Apple). Comme dit précédemment, Unix est un noyau, **il doit donc être utilisé avec d'autres logiciels** car il lui faut un système d'exploitation.

### 1.1.2 GNU : "GNU's Not UNIX"

GNU est un **projet des années 1990** lancé par Richard STALLMAN. C'est un **système d'exploitation libre** compatible avec le noyau UNIX. Il utilise des logiciels libres issus d'autres projets tels que le noyau Linux (*voir GNU/LINUX*). Sa **philosophie** est de maintenir intactes les **traditions hackers de partage** dans un monde de plus en plus marqué par l'empreinte du droit d'auteur. Il se bat donc pour une **libre diffusion des connaissances**. "GNU vise à ne laisser l'homme devenir ni l'esclave de la machine ni de ceux qui auraient l'exclusivité de sa programmation". Il est **utilisable et partageable librement par tous**, ainsi chacun complète petit à petit l'architecture initiale de GNU pour le performer. C'est dans ce contexte que le projet invite la communauté de hacker à le rejoindre et à participer à son développement. Il faut savoir qu'il est composé :

- d'un éditeur de texte (emacs),
- d'un compilateur très performant (gcc),
- d'un débogueur (gdb),
- d'un langage de script (bash),
- des bibliothèques de systèmes (glibc),
- (plus tard noyau ramené par un projet linux).

Malgré tout ces composants, GNU est toujours **incomplet** : son noyau est immature et incompatible avec certains de ses composants.

### 1.1.3 GNU/LINUX

GNU/LINUX a été créé en **1991**. Il est **initié par le projet Debian** et la naissance du noyau Linux. Il crédite donc à la fois Linux et GNU *mais l'usage de Linux est plus connu au grand public*. Il est alors toujours basé sur le **mouvement du logiciel libre et du mode opératoire du hacker**. Cet association a eu lieu pour combler le vide causé par le développement inachevé de GNU (*voir GNU*). Il est utilisé sur la plupart des téléphones portables comme sur les super-ordinateurs (par exemple, Android équipe 80% de ses smartphones). Ce projet fut un grand d'impacte dans le monde des serveurs informatiques. GNU/LINUX veut casser le fait qu'à l'origine il fallait des connaissances solides en informatique pour utiliser un système d'exploitation (*pas d'interface graphique et besoin d'installer toutes les applications soit même*). Il a donc été un important vecteur de **popularisation du mouvement de l'open source**. Il a eu des **centaines de milliers de redistributions** avec des versions différentes pour plaire à tout les goûts (en fonction des besoins, configuration, sécurité, ... *voir différents OS*). GNU/LINUX a remplacé d'autres systèmes de type Unix et/ou évite l'achat de licence Windows (qui est très cher à l'achat). Aujourd'hui on peut retrouver tout l'équivalent des logiciels/applications qu'il y a sous Windows mais en Open Sources.

## 1.2 différent OS

En informatique, un système d'exploitation est un ensemble de programmes qui dirige l'utilisation des capacités d'un ordinateur par des logiciels applicatifs.

### 1.2.1 Différents OS

- MacOS (série d'interfaces graphiques basé sur l'opération des systèmes développée par Apple pour leur Macintosh),
- iOS (système d'opération pour mobile développé et distribué par Apple pour iPhone et iPod),
- **Linux** (c'est un Unix. système d'opération pour ordinateur assemblé sous le modèle du "free and open source software"),
- Android (c'est un dérivé de Linux. système d'opération conçu principalement pour les mobiles tactiles comme les smartphones et les tablettes, initialement développé par Android),
- Microsoft Windows (série d'interface graphique développée et commercialisée par Microsoft),
- BSD/OS (réputé pour réhabiliter le rôle des serveurs, organisé par les programmeurs d'Unix, utilisé pour un utilisateur personnel du web).

### 1.2.2 différentes familles d'OS Linux

- **Debian/Ubuntu** : 1993

- développé par SPI (*Software in the Public Interest*),
- caractere non commercial et mode de gouvernance cooperatif,
- deja installer avec son noyau, ses pilotes, son programme d'installation de distribution, ses logiciels "utiles" (pour le WiFi, une navigateur web, ...) ,
- reuni une dizaine de sous-famille Debian avec le même noyau mais avec une architecture differente, dont les plus connus sont : KaliLinux, Kubunto, Raspbian, Ubuntu, Xubunto.

- **Red Hat/Cent OS/Fedora** : 1994

- developper et distribuer par l'entreprise Red Hat (entreprise dedier aux logiciels Open sources + distributeur de systeme d'exploitation Linux),
- plusieurs grosse distribution son issus de ce dev : Fedora, Enigma, ...
- principalement destinee aux serveurs des entreprises,
- voulait faire passer doucement les utilisateurs Windows sous Linux.

- **Arch** : 2002

- accent sur la simplicité et legerete : parfait pour les utilisateurs avancés,
- pas d'outils graphiques : pour resoudre tout les problemes du systemes,
- contribution ouverte (OpenSource) tant que cela respecte sa philosophie.

- **Suse/OpenSuse** : 1994

- distribution communautaire et commerciale,
- destine a l'utilisation en entreprise mais toujours en Open-Source,
- cycle de developpement long mais cycle de vie long,
- disponible par la vente (licence et mise a jour).

## 2 Language

### 2.1 LaTeX

### 2.2 Markdown

#### 2.2.1 Histoire

Le markdown est un **langage de balisage** crée en 2004. Il est facile à manipuler, donc simple à écrire et à lire sans connaître les balises. Le markdown peut être écrit sur n'importe quel éditeur de texte, il suffit lorsque le document est prêt à être enregistré de nommer le document puis d'inscrire **".md"**.

### 2.2.2 Instruction

Les instructions sont **très simple et peuvent être combiné**. On va voir quelques instructions de bases :

#### Police d'un texte

- Italique : encadrée le(s) mot(s) désiré(s) par \* ou \_
- Gras : le(s) mot(s) désiré(s) par \*\*
- Souligner : encadrée le(s) mot(s) désiré(s) par \_\_\_\_
- Barré : encadrée le(s) mot(s) désiré(s) par ~~
- souligner les titres : mettre sur une ligne en dessous des = ou -

#### Mise en forme du texte

- commencer un paragraphe : mettre 4 espaces
- délimiter un paragraphe : sauter une ligne
- retour a la ligne : deux espaces a la fin de phrase
- titre : # (rajouter des # par sous niveaux de paragraphes)

#### Ajout d'élément au texte

- bloc de code : encadrée le(s) mot(s) désiré(s) par ““
- citation : commercer par >
- liste : commencer par \* ou - ou +
- liste ordonnée : commencer par 1. 2. 3. ...
- cases a cocher : [ ] ou [x]
- tableau : - delimité les colonnes par : |- delimité les titres des autres lignes par : :—:
- lien : - en hypertexte : l'encadré en <et >- en bouton : [ *nom du bouton* ](#).btn
- mage : ![ *texte* ]( *url de l'image* )

## 3 Terminal

### 3.1 Terminal

### 3.2 Bash

#### 3.2.1 Bash et scripts

Le bash est un interpréteur de script natif aux systèmes d'exploitations sous environnement Unix.

**A quoi ca sert ?** Le bash permet l'exécution de scripts, petit morceaux de code, souvent exécutés au démarrage, afin d'effectuer une tâche précise. Par exemple, l'ensemble des applications s'ouvrant au démarrage sont regroupées dans un script. Ceux-ci sont généralement appelés paramètres de la commande. Bash permet l'exécution des commandes shell. Une commande shell est une chaîne de caractères en minuscules qui peut être invoquée au travers d'une invite de commande ou d'un script. Des options et des arguments peuvent la compléter.

Le bash propose différents traitements des commandes :

- Les commandes simultanées : `com1 & com2 & com3 & ... & comN`. Les commandes `com1` à `comN` sont exécutées simultanément;
- Les commandes successives : `com1 ; com2 ; com3 ; ... ; comN`. Les commandes `com 1` à `comN` sont exécutées les unes après les autres;
- Les commandes conditionnelles : `com1 && com2 && com3 && ... && comN`. La commande `N` est exécutée seulement si la commande `N-1` a elle aussi été exécutée;
- Les commandes alternatives : `com1 || com2 || com3 || ... || comN`. La commande `N` s'exécute seulement si la commande `N-1` ne s'est pas exécutée.

Il existe deux modes de fonctionnement du bash :

- Le mode interactif : Le terminal fonctionne avec le bash. lorsque l'on tape une commande manuellement dans le terminal, on crée en réalité un mini script d'une ligne, interprétée par bash puis exécutée;
- Le mode batch : bash exécute automatiquement un script contenu dans un fichier texte contenant les commandes à utiliser.

**L'interprétation d'une ligne de commande** Chaque interprétation d'une ligne de commande d'un script repose sur des codes bien stricts :

- Le premier mot de la ligne est interprété comme le nom de la commande;
- Chaque mot est séparé par un ou plusieurs caractères de séparation (espace, tabulation, tiret, underscore ...);
- La fin de la ligne se termine par un `'`, comme dans la plupart des langages de programmation, ou un retour à la ligne

Pour plus d'informations sur les lignes de commandes bash, il existe un poly sur les commandes du terminal disponible sur le git du Club\*Nix.

### 3.3 Oh My ZSH

## 4 Logiciel libre

Tout logiciel libre est défini par la **licence publique générale GNU** (*appelé GPL*) qui rend les logiciels indépendants de tout éditeurs et les encourage à l'entraide et le partage.

La licence générale est définie par 4 lois principales :

- utiliser un logiciel sans restrictions,
- etudier le logiciel,
- modifier pour l'adapter aux besoins des utilisateurs,
- redistribuer sous certaines conditions precises.

Cela a bcp d'avantages comme **correction rapide de bogues et de failles de securites**.

**Attention**, un logiciel libre n'est pas necessairement gratuit et inversement un logiciel gratuit n'est pas forcement libre !



## 4.1 Git

## 4.2 F-Droid

**F-Droid** est un magasin d'application (*au même titre que le PlayStore ou l'AppleStore*) qui met à disposition plus de **12000 applications libres et gratuites** sur les mobiles Android. Il a été créé en 2010 et est promu par la Free Software Foundation Europe. Son architecture de sécurité est basée sur le modèle de Debian. Comme toute construction OpenSource, F-Droid est tenu et développé par un grand nombre de contributeurs faisant partie de la communauté. Ainsi chaque personne peut créer sa propre application et la mettre à disposition sur ce magasin.

L'avantage à utiliser F-Droid plutôt que le magasin d'application par défaut, est principalement la préservation de vos données. Mais aussi la **sécurité** de vos applications et le fait qu'il n'est pas nécessaire de s'identifier pour pouvoir télécharger des nouvelles applications. Ce magasin facilite la découverte et l'installation de multiples applications. De plus contrairement à la plupart des applications, vous n'êtes pas obligé de faire les mises à jour, et ainsi garder une ancienne version.

# 5 Infrastructure réseau

## 5.1 LDAP

### 5.1.1 Le LDAP, ou *Lightweight Directory Access Protocol*

Lorsqu'un utilisateur tente de se connecter à un ordinateur, le LDAP reçoit une combinaison login/mot de passe et donne ensuite ou non l'autorisation de se connecter.

### 5.1.2 Introduction

Le LDAP est un protocole créé en 1995, succédant au protocole DAP et permettant l'accès et la modification de base de données sur les utilisateurs d'un réseau. Un LDAP sert notamment à se connecter ou se déconnecter d'un serveur hébergeant le LDAP afin d'y être identifié, mais aussi à chercher des informations, les comparer, ajouter des utilisateurs...

## 5.2 Le LDAP du club Nix ou de l'école

Le LDAP du club NIX ou de l'école permet de se connecter et de s'identifier afin de récupérer ses informations, quelque soit la machine utilisée.

### 5.2.1 Les entrées

Un LDAP définit l'accès aux entrées (ou utilisateurs la plus part du temps). Le LDAP ne peut gérer que des entrées. Une entrée peut être un nom d'utilisateur, un périphérique ou encore des paramètres. Il existe 2 types d'entrées, les entrées normales et les entrées opérationnelles :

- Les entrées classiques, telles le nom d'utilisateur ou la date d'anniversaires sont des entrées dites classiques.

- Les entrées opérationnelles, tels les parametres, les dates de modification, qui ne sont accessibles et utilisables uniquement par le serveur.

Une entrée est définie par son nom, ou DN pour *Distinguished Name*, composé d'une serie de clé et de valeur de ces clé. Par exemple, la clé *uid* définit le nom d'utilisateur et la clé *cn* définit le nom. Pour ces deux clé, une entrée serait sous la forme : *uid=utilisateur,cn=nix*

### 5.2.2 L'arborescence

Un serveur hébergeant un LDAP est organisé selon une arborescence, comme un systeme de fichier, dans lequel chaque branche correspond à une entrée. Une branche situé à la racine sera appelé racine ou root en anglais. Le scheme d'une clé correspond a l'ensemble des valeurs des attributs ou valeurs attribué aux clés. Les annuaires LDAP répondent à certaines règles de structure :

- Un annuaire est un arbre d'entrées;
- Une entrée est constituée d'un ensemble d'attributs;
- Un attribut possède un nom, un type et une ou plusieurs valeurs;
- Les attributs sont définis dans des schémas.

### 5.2.3 Chercher des informations d'un LDAP

Le protocole LDAP fournit un ensemble de fonction permettant d'interroger le serveur sur lequel est herbergé le serveur LDAP afin de modifier, ajouter ou supprimer des entrées. On peut citer notamment *\*add\** pour ajouter une entrée, *\*delete\** pour la supprimer, ou *\*rename\** pour la renomer, afin de modifier l'arborescence du LDAP.

### 5.2.4 Les principales commandes du LDAP

Voici une lise des principales clés d'un LDAP :

- *userid* (userid), il s'agit d'un identifiant unique obligatoire;
- *cn* (common name), il s'agit du nom de la personne;
- *givenname*, il s'agit du prénom de la personne;
- *sn* (surname), il s'agit du surnom de la personne;
- *o* (organization), il s'agit de l'entreprise de la personne;
- *u* (organizational unit), il s'agit du service de l'entreprise dans laquelle la personne travaille.
- *mail*, il s'agit de l'adresse de courrier électronique de la personne

Voici une liste des principales commandes d'un LDAP :

- *abandon* : Abandonne l'opération précédemment envoyées au serveur;

- *add* : Ajoute une entrée au répertoire;
- *bind* : Initie une nouvelle session sur le serveur LDAP;
- *compare* : Compare les entrées d'un répertoire selon des critères;
- *delete* : Supprime une entrée d'un répertoire;
- *extended* : Effectue des opérations étendues;
- *rename* : Modifie le nom d'une entrée;
- *search* : Recherche des entrées d'un répertoire;
- *unbind* : Termine une session sur le serveur LDAP.

## 5.3 NFS

### 5.3.1 Introduction

Un NFS, ou *Network File System*, à traduire par *Système de fichier en réseau*, est un protocole permettant de partager et récupérer des données via un réseau. Les réseaux utilisant un NFS permettent l'utilisation des fichiers sauvegardés sur le serveur hébergeant le NFS à distance, tant que la connection est établie. En pratique, on peut récupérer ses données sauvegardées sur n'importe quelle machine.

### 5.3.2 Le NFS du Club\*Nix ou de l'ESIEE

Le club NIX et l'ESIEE possèdent un NFS, permettant de se connecter à partir de son identifiant LDAP (voir chapitre LDAP), et de récupérer ses données personnelles, quelle que soit la machine utilisée.

### 5.3.3 Architecture d'un NFS

Un NFS suit un modèle d'architecture classique réseau : c'est à dire que chaque client ( utilisateurs ) se voit accorder un espace de stockage maximal, exactement comme pour un système de fichier local. Par la suite, les données sont sauvegardées sur le serveur hébergeant le NFS, disposant d'une bien plus grosse capacité de stockage. Chaque demande d'accès ou de sauvegarde de fichier passe par le réseau.

### 5.3.4 NFS et VFS

Sous Linux, un protocole permet de prendre en charge plusieurs systèmes de fichiers différents sur un même serveur : c'est le VFS. Le système VFS détermine le stockage auquel une demande est effectuée. Lorsqu'une demande s'avère être destinée au NFS, VFS la transmet au noyau ( appelé kernel ) du NFS. Le NFS interprète alors la requête d'entrée ou sortie et la traduit en procédure exécutable par le protocole NFS. On peut notamment citer les procédures suivantes :

- OPEN
- ACCESS

- CREATE
- READ
- CLOSE
- REMOVE
- ...

Le serveur NFS va alors satisfaire la demande de l'utilisateur en lui "renvoyant" ou "effectuant" sa demande. En soit, NFS n'est pas un système de fichier au sens propre mais un protocole réseau permettant d'accéder à des fichiers à distance via un réseau.

### **5.3.5 La sécurité du NFS**

Il existe actuellement 4 version de NFS. Dans les 3 premières versions, le protocole NFS n'était pas sécurisé et permettait l'utilisation en local, comme dans une école ou au club nix par exemple. La dernière version ( v.4 ) est doté d'un système de chiffrement comprenant la négociation du niveau de sécurité entre client et serveur, une sécurisation simple mais efficace, ainsi qu'un chiffrement des communications. La version 4.1 du système est actuellement en cours de développement et n'est pas prévu avant plusieurs années.

## **5.4 Fichiers logs**

## **5.5 Virtual Machine**

## **5.6 Infra du club**