

# Insta poste - Intro sys

## Contents

Introduction . . . . .	3
Processus en Crabe . . . . .	4
Redirections (entrées/sorties) de Zinzin mon pote à la compote . . . . .	4
Filtres . . . . .	5
tr . . . . .	5
cut . . . . .	5
Sed et son fonctionnement casse pieds . . . . .	6
Affectation des variables . . . . .	6
Gestion des users et groupe . . . . .	7
package manager . . . . .	8
Configuration . . . . .	9
Partitions . . . . .	10
Important : les Block Devices . . . . .	10
Facultatif : Faire un instantané . . . . .	10
Dual boot . . . . .	11
Fonctionnalité systeme . . . . .	11
Xorg -> wayland (mais I3 c'est mieux) . . . . .	12
gestion kernel . . . . .	12
manual install (nsm jveux pas le faire) . . . . .	12
Installation manuelle (nan gros c'est bon va prendre une douche) . . . . .	12

S1.03 -R1.04



Made by Sullivan, Adam, Ethan (Ane).

## Introduction

Ce petit guide vous permet de réaliser la SAÉ 1.03 sans problème ;)

### Voici quelques petits tips

Durant toute cette page, on utilisera une syntaxe précise pour spécifier comment exécuter une commande ou modifier un fichier.

\$ commande

Indiquera une commande à exécuter en tant qu'utilisateur normal.

# commande

Indiquera une commande à exécuter en tant que root.

### Cela revient à taper :

\$ sudo commande

(à condition que vous soyez dans le fichier /etc/sudoers (Voir la suite))

### Pour devenir root :

\$ su

Puis taper le mot de passe.

### Les modifications de fichiers

Quand nous modifierons un fichier, nous utiliserons toujours cette syntaxe :

FILE /chemin/absolu/vers/le/fichier

-----

Contenu du fichier à modifier (la plupart du temps, cela consiste à ajouter des lignes dans le fichier)

## Processus en Crabe

### PID

non-spécifique au term dans lequel on l'a lancé

```
$ ps
```

```
$ pstree
```

### Jobs

spécifique au term dans lequel on l'a lancé

pour parler des jobs il faut mettre % avant le numéro de jobs

```
$ jobs
```

### Kill/Stop/...

gère l'état des processus

```
$ kill -s <signal> <PID/%jobs>...
```

### liste signaux:

```
> * KILL : force kill
```

```
> * STOP : stop
```

```
> * TERM : kill gentil trop choupi
```

```
> * CONT : relance un processus précédemment stoppé
```

```
> * sinon y a des numéros mais flm
```

### bg/fg

fout en arrière plan (bg/'&') ou en avant plans (fg) un processus

```
$ fg/bg <PID/%jobs>...
```

## Redirections (entrées/sorties) de Zinzin mon pote à la compote

### nom term

```
$ tty
```

### n (descripteur) {0, 1, 2}:

```
> * 0 : entrée standard
```

```
> * 1 : sortie standard
```

```
> * 2 : sortie erreur
```

### afficher le contenu d'un fichier dans un autre term

```
$ cat <fichier> 1><term>
```

### afficher une erreur dans un autre term

```
$ cat <fichier> 2><term>
```

### écraser/créer un fichier en redirigeant un texte d'un autre fichier

```
$ cat <fichier> 1><fichier>
```

### append le contenu d'un fichier dans un autre fichier

```
$ cat <fichier> 1>><fichier>
```

### rediriger la sortie standard sur la sortie d'erreur

```
$ ???
```

## Filtres

### tr

- [:alpha:] : [a-zA-Z]
- [:alnum:] : [a-zA-Z0-9]
- [:digit:] : [0-9]
- [:blank:] : space
- [:graph:] : tout sauf espace
- [:print:] : tout
- [:xdigit:] : [A-F0-9]
- [:lower:] / [:upper:] : min/maj

### full maj

```
$ cat <fichier> | tr [:lower:] [:lower:]
```

### suppr avec tr

```
$ cat <fichier> | tr -d <elem>
```

### cut

N = Borne de debut M = borne de fin > \* "borne non-comprise"

> \* -M : garde tout ce qu'il y a avant la borne

> \* N- : garde tout ce qu'il y a après la borne

> \* N-M : garde ce qu'il y a entre les bornes

> \* -N,M- : garde ce qu'il y a autour des bornes

> \* "borne non-comprise"

- -b : nb bytes / ensemble de bytes
- -c : nb char / ensembles de char
- -d : détermine un char en délimiteur
- -f : détermine le field (nb de fois qu'on va passer le delim)(s'utilise avec -d)
- -complement : prend tout sauf ce qui est select par le cut (cut avec selection inversé)

### cut un elem spécifique

```
$ cat <fichier> | cut -c <borne1>-<borne2>
```

### retirer un elem spécifique

```
$ cat <fichier> | cut -c -<borne1>,<borne2>-
```

### cut du début jusqu'à un elem spécifique

```
$ cat <fichier> | cut -c -<borne1>
```

### cut d'un elem spécifique jusqu'à la fin

```
$ cat <fichier> | cut -c <borne2>-
```

### Un peu de commande chiante ? source: tp04

Utilisez cette commande connectée à d'autres filtres via des tubes pour effectuer les actions suivantes : 1. afficher le

```

nombre d'établissements;reponse: ./annuaire | cut -f 1 -d ";" | tr ";" "\n" | wc -l
2. afficher les valeurs utilisées pour la colonne Statut_public_prive ; Les valeurs à trouver sont :
> — une valeur vide
> — -
> — Public
> — Privé
Reponse: ./annuaire | cut -f 4 -d ";" | sort -d | uniq

```

### **Sed et son fonctionnement casse pieds**

expliquons comment fonctionne sed, sed a une base de ligne commande suivante TOUT COLLÉ:

```
sed [LIGNE] LETTRE [OPTION]
```

LIGNE : désigne le n° de ou des lignes sur laquelle cela doit s'exécuter :

- > rien pour toutes les lignes
- > N pour désigner LA lignes
- > N,M pour les lignes entre N et M
- > /MOT/ pour désigner la ligne content MOT

LETTRE : désigne l'action qui va demandé :

- > p pour afficher les lignes demandé. A utiliser avec sed -n par exemple **sed -n 5p**
- > d pour supprimer des lignes
- > s pour effectuer un remplacement

OPTION : va s'utiliser SEULEMENT avec LETTRE s , et s'écrit sous la forme suivante(option) : /MOT/REPLACEMENT/g

### **Exemple**

```
cat tuyaux | sed 3,4s/commande/chouette/g : dans le fichier tuyaux, remplacera dans les lignes 3 et 4 le
mots commande par chouette
```

### **Affectation des variables**

**pour crée une variable il faut suivre le schéma suivant :**

```
<nom>=<valeur>
```

**par exemple l'on veut assigner a la variable \$LOGIN le texte d'exemple suivant sullivan.cerdan.etu :**

## Gestion des users et groupe

- les infos users sont stockés dans /etc/passwd
- les passwd chiffrés sont stockés dans /etc/shadow
- les infos des groupes sont dans /etc/group

### add sudoers (add user au grp sudo)

```
$ usermod -aG sudo user
```

### La création d'utilisateur

adduser (sur debian et ubuntu) - créer un utilisateur sur la machine

### changer le mdp user

```
$ passwd <user>
```

### donner le droit d'exec tt les cmd dans /etc/sudoers

```
<user> ALL=(ALL:ALL) ALL
```

### OU ALORS, sur Debian

```
# usermod -aG sudo <utilisateur>
```

### suppr un user

```
$ deluser <user> --remove-home
```

### La création d'un groupe

sudo addgroup - créer un groupe

### ajout d'un utilisateur dans un groupe

pour exemple, quand on créer l'utilisateur tata

```
adduser tata
```

### changer le groupe d'un fichier

```
$ chown <:group> <file>
```

### changer le propriétaire d'un fichier

```
$ chown <user> <file>
```

## package manager

Sur **Debian**, en TP, on utilisera le gestionnaire de paquets **APT** (Advanced Packaging Tool). Voici une petite liste de commandes :

**Attention !** Ne pas oublier installer les lib des packages manager

### Installer un paquet

```
# apt-get install <paquet>
--- OU ---
# apt install <paquet>
```

### Supprimer un paquet

```
# apt-get purge <paquet>
--- OU ---
# apt purge <paquet>
```

NOTE : l'opération `remove` existe également, mais on préférera `purge` car elle supprime **tous** les fichiers du paquet.

### Synchroniser la liste des paquets avec le dépôt distant

Cela permet de bien être sûr que la liste des paquets est à jour et prête pour les mises à jour poussées par les développeurs de Debian.

```
# apt update
--- OU ---
# apt-get update
```

### Chercher un paquet dans la base de données

Ça pourra vous être utile pour chercher des paquets dont vous ne connaissez pas le nom exact.

```
$ apt search <texte>
```

### Chercher un fichier dans les paquets

Parfois, vous aurez besoin d'une commande, mais vous ne connaîtrez peut-être pas le nom du paquet. On peut donc utiliser `apt-file` pour chercher un fichier contenu dans un paquet.

Commençons par installer et configurer `apt-file`

```
# apt install apt-file
# apt-file update
```

On peut ensuite chercher notre fichier :

```
$ apt-file search <texte>
```

### Mettre à jour tous les paquets

```
# apt upgrade
--- OU ---
# apt-get upgrade
```

Sur Debian, APT utilise DPKG pour installer les paquets et modifier la base de données. On peut donc utiliser `dpkg` nous-même pour installer un paquet local.

### Installer un paquet local

```
# dpkg -i <chemin/vers/paquet>
```

NOTE : pour ceci, on recommande quand même **d'utiliser APT, qui pourra ainsi installer les éventuelles dépendances du paquet sans causer d'erreur et tout casser**. On utilisera donc

```
# apt install <chemin/vers/paquet>
```



**installer un packet avec snap**

```
$ snap install <packet>
```

**installer un packet avec homebrew**

```
$ brew install <packet>
```

sinon tu prend la commande pour install dans la doc de l'app que t'installe

**supprimer un packet**

```
$ apt-get remove <packet>
```

**purge(full remove) un packet**

```
$ apt-get purge <packet>
```

pour le reste tu regarde la doc de ton PM j'ai pas que ça à faire

**Installez le gestionnaire de bureau soit LXDE**

```
$ apt-get install task-ldxe-desktop
```

pour le reste tu regarde la doc de ton PM j'ai pas que ça à faire

**Configuration****changer la langue sys**

```
$ dpkg-reconfigure locales
```

**changer la timezone**

```
$ dpkg-reconfigure tzdata
```

**changer le profils de touche (azerty/qwerty)**

```
$ dpkg-reconfigure keyboard-configuration
```

## Partitions

### Important : les Block Devices

Sur les systèmes UNIX, les disques ne sont pas identifiés par des **lettres**, comme sur Windows, mais par des **fichiers spéciaux**. Ils sont tous accessibles dans le répertoire `/dev`, et peuvent être sous plusieurs formes : `*/dev/sdX` la plupart du temps, surtout pour les disques SATA `*/dev/nvmeX` pour les SSD NVME `*/dev/mmcblkX` pour les stockages flash Vous pouvez trouver une liste des disques de votre système (avec leurs noms et tailles) avec `fdisk`. # `fdisk -l` **Plus tard, on utilisera tout simplement pour parler d'un disque, ou d'une partition, donc il est important de savoir de quoi on parle.**

### Facultatif : Faire un instantané

Pour sauvegarder une image d'une partition ou d'un disque, afin de pouvoir la restaurer plus tard, vous pouvez utiliser la commande `dd`.

**\*\*Attention:** Cette commande est à haut risque et pourrait complètement nuke votre disque. À utiliser avec précaution.“

#### Pour faire une image :

```
dd if="/dev/sdXX" of="fichier_sortie.iso"
```

#### Pour pouvoir la reflash ensuite :

```
dd if="fichier_entree.iso" of="/dev/sdXX"
```

### Afficher l'espace libre sur les disques

```
$ df -h
```

### Afficher les informations sur les partitions

```
$ fdisk -x
```

### Modifier les partitions d'un disque

```
$ fdisk <disk>
```

### Ajouter une partition 4 étapes :

(dans `fdisk`)

1. `n` – Ajouter une nouvelle partition 2. `p` – En faire une partition primaire 3. – Laissez vide et appuyez sur entrée 4. `+XG` ou `+XM` – Donner la taille de la partition en Go ou en Mo (+5G créera une partition de 5 Go).

### Supprimer une partition (dans `fdisk`)

1. `d` – Supprimer une partition 2. Entrer le numéro de partition demandé (s'il y en a plusieurs)

### Modifier une partition

Vous pourrez également modifier le type d'une partition (**ici, sur la VM, vous n'en aurez pas besoin**) (dans `fdisk`)

`t` – Changer le type de partition

`L` – Lister les types de partition

`83` – Par exemple, pour une partition de type linux

### Quitter `fdisk`

`w` – Écrit les changements et quitte `fdisk`

`q` – Quitte sans écrire les changements

### Formater une partition

Une fois vos partitions créées, il faudra les formater pour pouvoir les utiliser. Les systèmes Linux utilisent quasiment tous `ext4`, mais il en existe d'autres :

`*mkfs.ext4 <partition>` – Créer une partition `ext4` (Par exemple `/dev/sda2`)

`*mkfs.vfat -F32 <partition>` – Créer une partition `FAT32`

### Monter la partition dans un dossier

```
# mount <partition> /chemin/vers/dossier (doit être vide)
```

## Dual boot

La mise en place d'un dual boot dépend beaucoup des systèmes. Si vous ne voulez pas trop vous casser les couilles, vous pouvez simplement :

1. Démarrer sur l'ISO d'une distribution comme Ubuntu, Manjaro Linux ou Solus Linux
2. Commencer l'installation comme si de rien n'était
3. Au moment où les disques sont demandés, choisir "Installer à côté de (nom de votre 1er système)", choisir oui, et installer

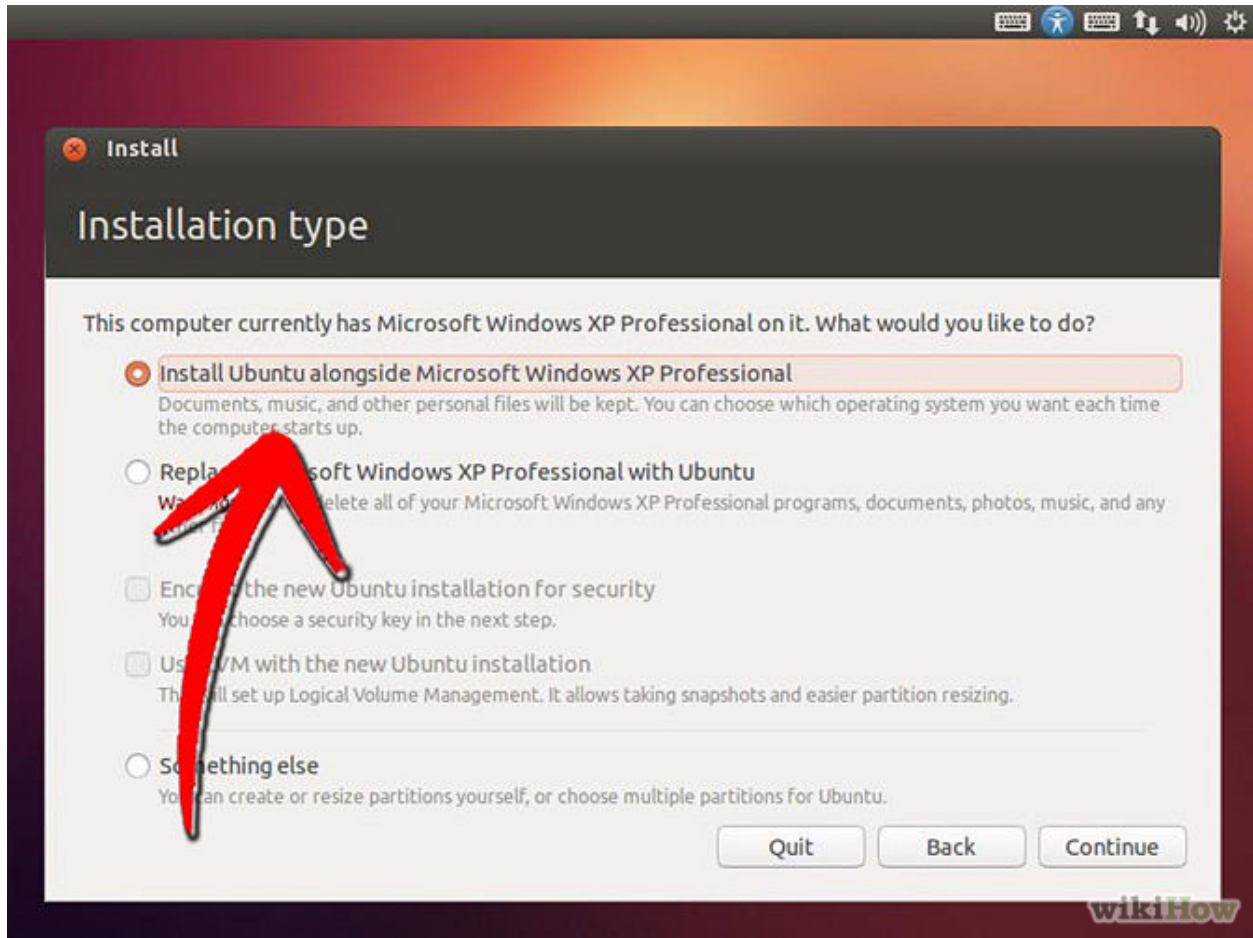


Figure 1: Install Ubuntu alongside Microsoft Windows XP Professional

## Fonctionnalité système

### Faire en sorte que Firefox s'ouvre automatiquement

Écrire dans `~/.config/autostart/firefox.desktop`

```
[Desktop Entry]
Name=Firefox
Comment=Web Browser
GenericName=Web Browser
Exec=firefox
Icon=firefox
Type=Application
```

### Faire en sorte de ne plus avoir besoin de mot de passe pour se connecter en tant que root

En modifiant `/etc/lightdm/lightdm.conf`, dans la section `[Seat: *]`

```
[Seat:*]
autologin-user=root
autologin-user-timeout=0
```

Ensuite, redémarrez ou relancez lightdm avec la commande :

```
$ sudo systemctl restart lightdm
```

### **Xorg -> wayland (mais I3 c'est mieux)**

En fait, il faut savoir que **tout n'est pas compatible wayland**. La plupart du temps, les gestionnaires de fenêtres et environnement de bureau (comme GNOME, KDE, Xfce) utilisent Xorg installé sur le système **ou** Wayland, qu'ils prennent en charge nativement sans installation.

Ainsi, si on veut installer Wayland, il suffit de trouver un environnement de bureau compatible wayland :

```
$ sudo apt install gnome # installer GNOME
$ sudo apt install plasma-workspace-wayland # installer KDE pour Wayland
```

### **gestion kernel**

Si vraiment vous n'avez aucune vie et ne prenez pas de douche, vous pouvez simplement utiliser le [magnifique guide du wiki Debian \(en Anglais\)](#).

On vous laisse vous démerder.

### **manual install (nsm jveux pas le faire)**

**Installation manuelle (nan gros c'est bon va prendre une douche)**

Va lire le livre et fais pas chier.

- [En Français](#)
- [En Anglais](#)