

2021-2022

SAE 1.03

Compte rendu



**BARRY ADONDE IBRAHIM
HERILUS ALLAN
CHEVALIER GASPARD**

Table des Matières

- 01 - Réponses aux questions**
- 03 - Installation d'une VM**
- 06 - Partie Pratique**
- 09 - Invite de commande modifiée**
- 10 - Installation de GCC et d'un IDE**
- 13 - Compilation séparée avec GCC**
- 15 - Conclusion et fin**

QUESTIONS

Définition de la virtualisation : La virtualisation est la simulation de machines sur une machine hôte. On va répartir les ressources matérielles entre les différents systèmes d'exploitation. Les systèmes sont isolés ce qui offre une certaine sécurité.

Émulateur : L'émulation c'est reproduire à l'identique un comportement logiciel et matériel.

Simulateur : La simulation c'est le fait d'imiter une situation, comme pour les simulateurs de vol où l'on va simuler un vol et où le pilote pourra s'entraîner comme dans un vrai avion.

Outil/Logiciel de virtualisation : Pour ce qui est de la virtualisation, on va faire comme de l'émulation, sauf qu'au lieu d'utiliser le matériel émulé, on utilisera directement le matériel de l'hôte. On a alors un gain de performance.

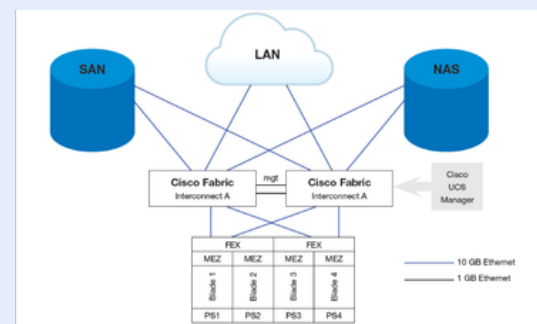
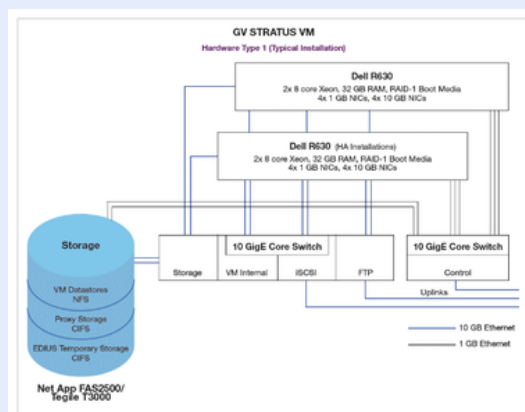
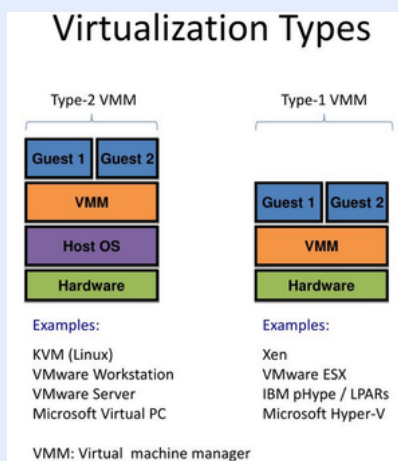
Types de virtualisation :

1) **VirtualBox :** Logiciel de virtualisation (Type 2). Besoin d'un système d'exploitation pour fonctionner.

2) **VMWare :** Version logicielle (fonctionne comme VirtualBox), et version Type-1, qui ne requiert pas de système d'exploitation pour fonctionner, seulement d'un Gestionnaire de machines virtuelles.

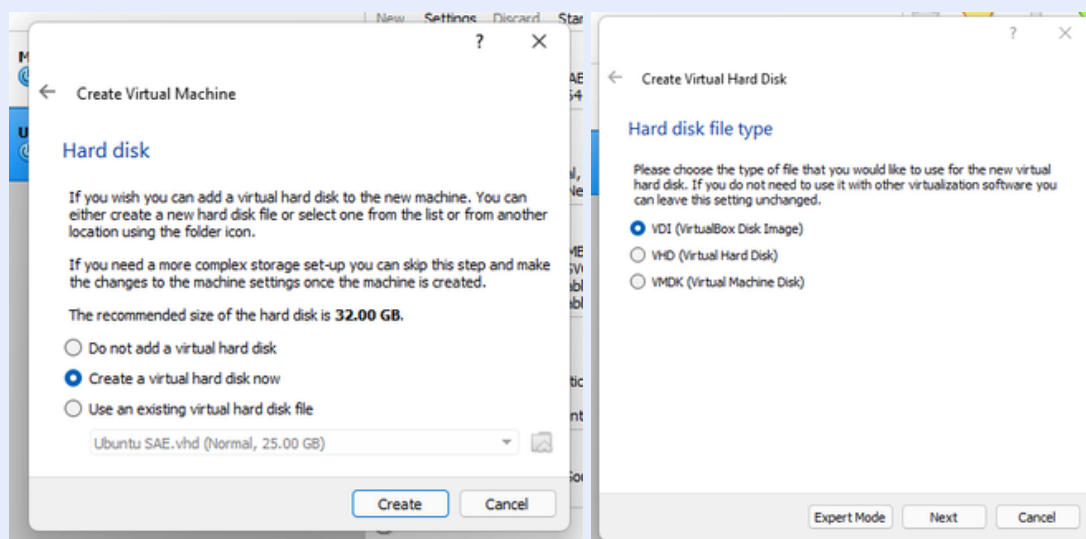
3) **Stratus :** Deux versions disponibles comme pour VMWare, on accède simplement via le navigateur à une machine virtuelle distante. (cf.

https://wwwapps.grassvalley.com/docs/Application_Notes/media_workflow/gv_stratus/GV_STRATUS_Virtualization_Guide-201901.pdf)



SUITE

Forme / Lieu de sauvegarde de la machine virtuelle créée : On peut retrouver la machine virtuelle sous différentes formes de sauvegardes. La machine virtuelle sera stockée sous la forme d'un disque virtuel comme nous le montre la capture d'écran ci-dessous :



On peut alors choisir entre 3 formats :

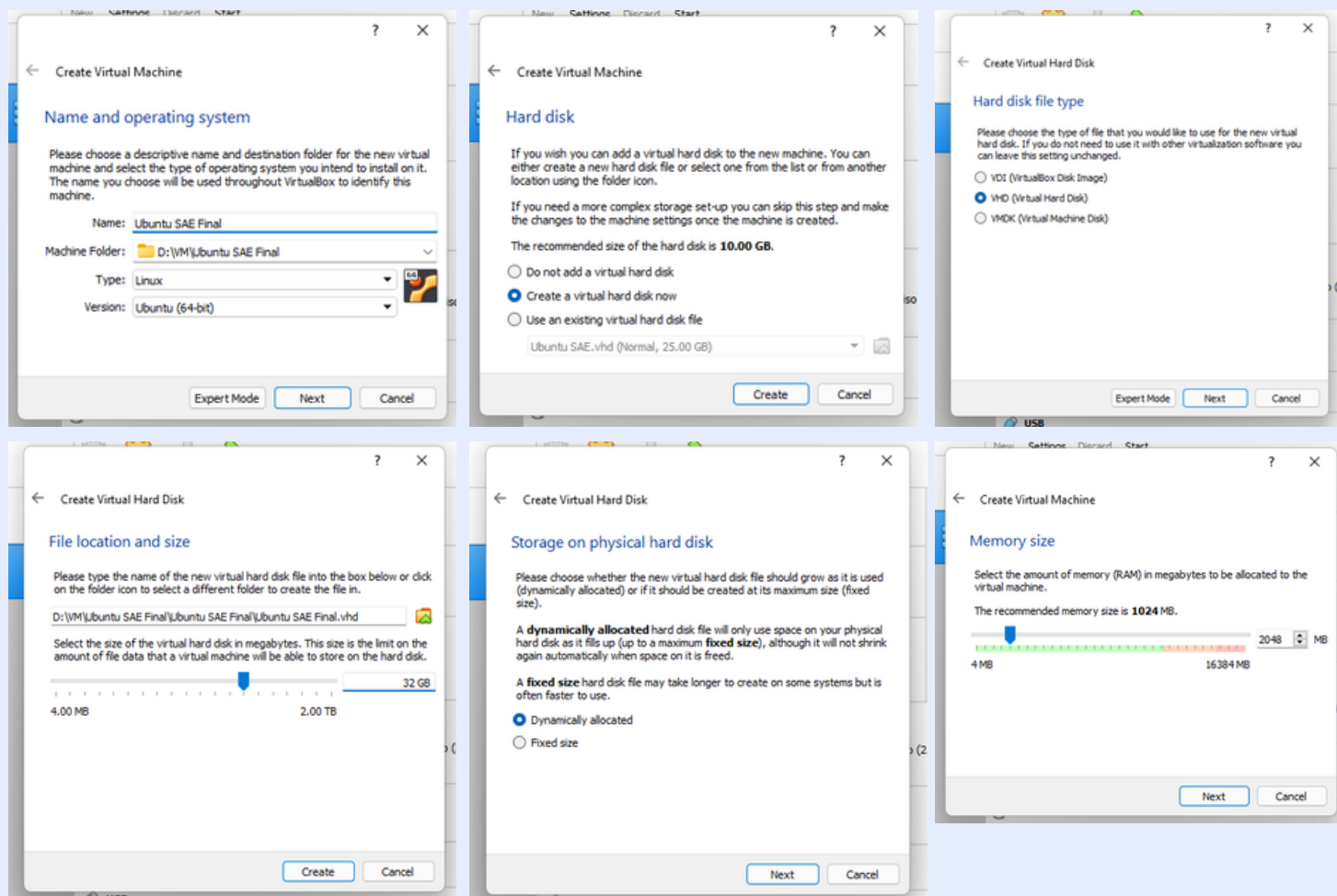
- **VDI (VirtualBox Disk Image) :** C'est le format utilisé par VirtualBox. Il est portable. C'est à dire que l'on pourra l'utiliser sur d'autres logiciels de virtualisation ou le mettre sur un autre PC.
- **VHD (Virtual Hard Disk) :** C'est le format utilisé par Microsoft. L'image reste en tant que fichiers sur l'OS host. Il est donc plus facile de faire des backups, faire des configurations et un tas d'autres choses.
- **VMDH (Virtual Machine Disk) :** C'est le format utilisé par VMWare. Le disque n'a par exemple pas de taille fixe et peut ainsi augmenter au fur et à mesure que l'on y ajoute des données. Il est aussi très facile de faire des snapshots ou de migrer vers d'autres serveurs.

Sous quelle forme retrouve-t-on la machine virtuelle créée dans la machine réelle ? :

On retrouve la machine virtuelle sous la forme d'un disque virtuel et d'un processus. Le processus (VMWare, VirtualBox...) créera des composants virtuels où le disque et le système d'exploitation seront lancés. Il sera donc en mémoire primaire d'où le fait que l'on nous demande combien de Gb allouer lors de la création de la VM.

INSTALLATION

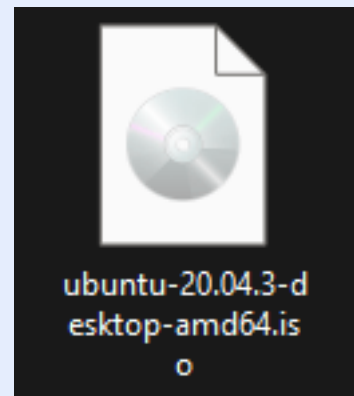
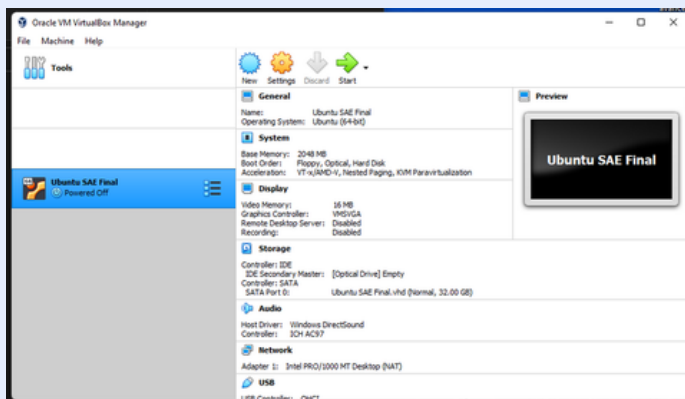
Nous avons cherché à faire le plus simple possible. Ainsi, nous avons choisi d'installer Ubuntu sur VirtualBox. (<https://ubuntu.com/download/desktop>)



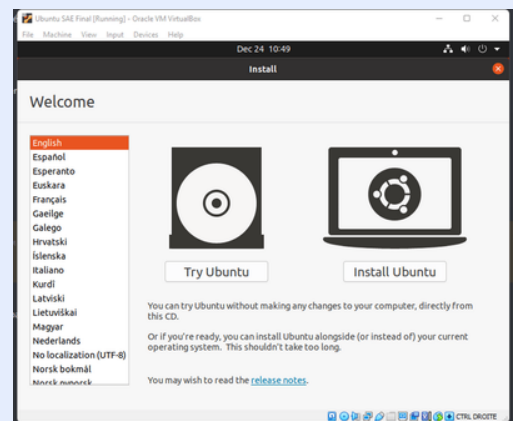
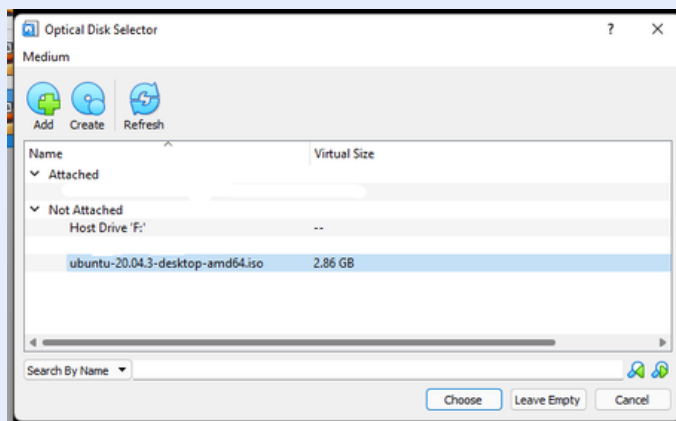
Comme vous l'avez vu sur les captures d'écran ci-dessus, nous avons fait le choix de créer un disque virtuel VHD (Virtual Hard Disk) de 32GB, qui sera dynamiquement alloué. C'est à dire qu'il ne prendra sur le disque de l'OS hôte seulement la place de l'ensemble des fichiers et dans une limite de 32GB. Ainsi, même si nous avons mis 100GB et que l'OS ne faisait que 1GB, la VM n'aurait pris qu'approximativement 1GB sur notre disque physique. Enfin, nous avons alloué à Ubuntu 2048MB de RAM (Random Access Memory), pour nous garantir une grosse marge et nous permettre de lancer plusieurs programmes en même temps sans qu'il n'y ait d'impact sur les performances de la machine virtuelle.

INSTALLATION

Dans cette seconde partie de l'installation, nous allons vous montrer comment nous avons boot sur l'Operating System "Ubuntu".



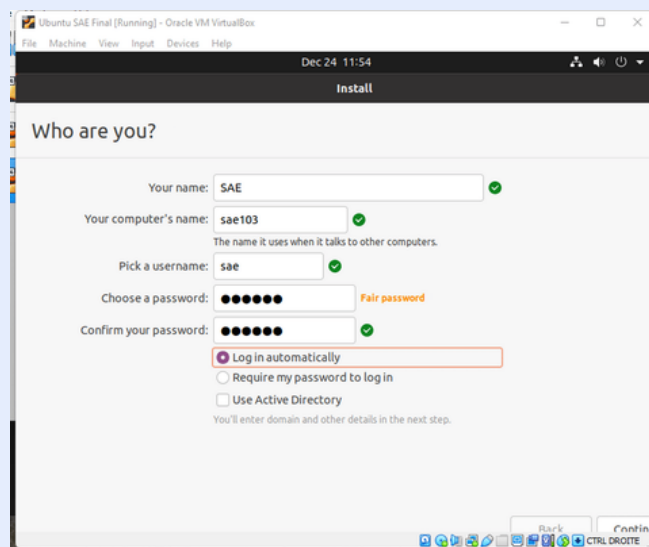
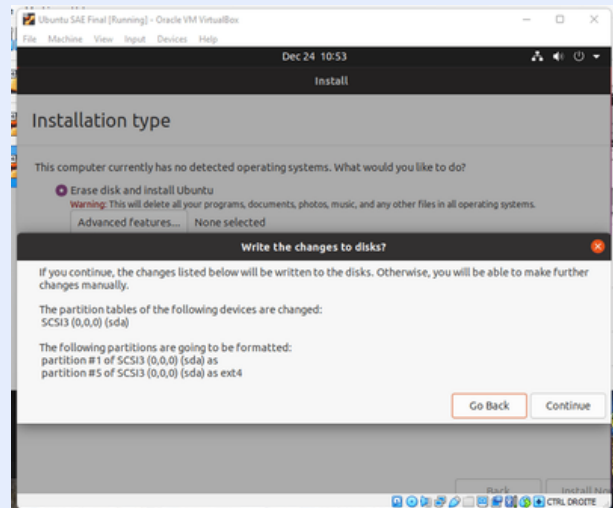
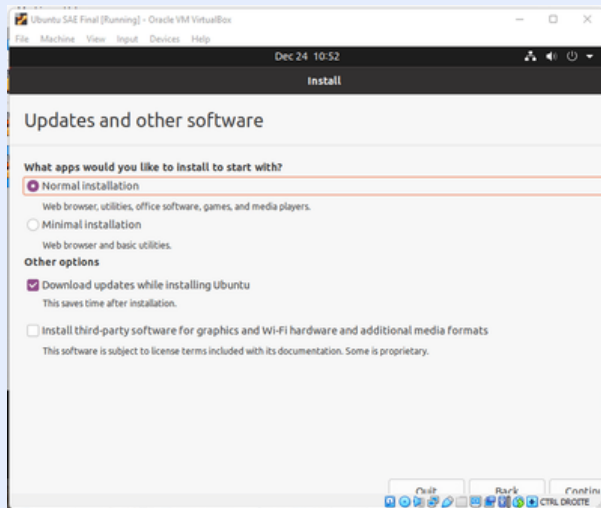
On se munit dans un premier temps d'une machine virtuelle (ici celle créée précédemment), et d'un .iso qui est l'image d'un système d'exploitation. Autrement dit, le .iso est un disque sur lequel se trouve Ubuntu dans notre cas.



On demande ensuite à VirtualBox de boot sur le disque que l'on a téléchargé. Et ça marche !

INSTALLATION

On passe maintenant à la dernière partie qui est celle de l'installation d'Ubuntu.



Nous avons choisi de faire une installation Normale, c'est à dire avec tout d'installer car nous pensions avoir besoin par la suite d'applications préinstallées. Ensuite, nous avons choisi le disque virtuel crée au début comme disque pour l'installation. Enfin, on a crée un compte utilisateur qui s'appelle sae pour ne pas l'oublier. On peut désormais commencer à travailler sur notre espace de travail pour y faire toutes les manipulations demandées!

PRATIQUE

Petit rappel sur les commandes à utiliser :

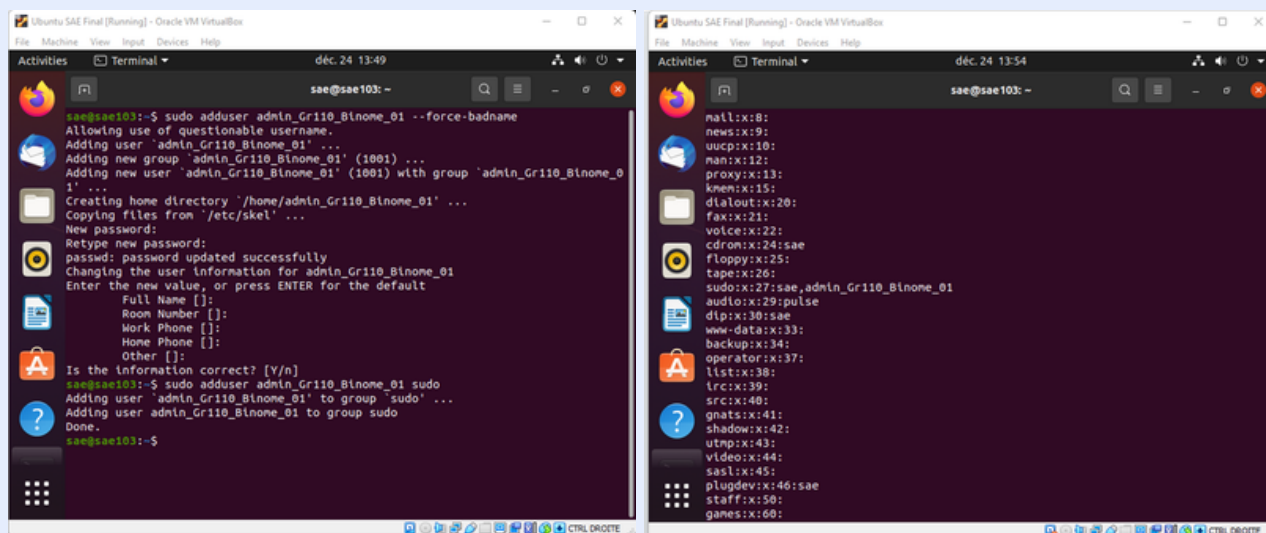
- Pour créer un utilisateur administrateur : `adduser <username> [group]`
- Pour créer les deux utilisateurs : `adduser <username> [group]`
- Pour créer un groupe : `groupadd [option] <groupe name>`

Manipulation 1 (Création d'un utilisateur avec les droits administrateurs) :

- 1) `sudo adduser admin_Gr110_Binome_01 --force-badname` (On crée un utilisateur)
- 2) Entrer le mot de passe que l'on veut pour l'utilisateur
- 3) `sudo adduser admin_Gr110_Binome_01 sudo` (Ajout au groupe sudo, les administrateurs)

Explications :

- `sudo` : On exécute la commande avec les droits administrateurs
- `adduser` : Commande pour créer un utilisateur sous Ubuntu
- `admin_Gr110_Binome_01` : Le nom de l'utilisateur que l'on va créer
- `--force-badname` : On a utilisé `_` (underscore) qui n'est pas supporté par l'expression régulière, on doit alors forcer la création de l'utilisateur



The image shows two screenshots of a terminal window in a VirtualBox environment. The first screenshot shows the execution of the command `sudo adduser admin_Gr110_Binome_01 --force-badname`. The output shows the user being created, a password being set, and the user being added to the `sudo` group. The second screenshot shows the output of the command `cat /etc/group`, which lists the groups and their members. The `sudo` group is listed with `admin_Gr110_Binome_01` as a member.

```
sae@sae103:~$ sudo adduser admin_Gr110_Binome_01 --force-badname
Adding user 'admin_Gr110_Binome_01' ...
Adding new group 'admin_Gr110_Binome_01' (1001) ...
Adding new user 'admin_Gr110_Binome_01' (1001) with group 'admin_Gr110_Binome_01' ...
Creating home directory '/home/admin_Gr110_Binome_01' ...
Copying files from '/etc/skel' ...
Retype new password:
passwd: password updated successfully
Changing the user information for admin_Gr110_Binome_01
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n]
sae@sae103:~$ sudo adduser admin_Gr110_Binome_01 sudo
Adding user 'admin_Gr110_Binome_01' to group 'sudo' ...
Adding user admin_Gr110_Binome_01 to group sudo
Done.
sae@sae103:~$
```

```
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:sae
floppy:x:25:
tape:x:26:
sudo:x:27:sae,admin_Gr110_Binome_01
audio:x:29:pulse
dip:x:30:sae
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:sae
staff:x:50:
games:x:60:
```

On regarde avec `cat /etc/group` que notre utilisateur a été ajouté au groupe sudo, et c'est le cas !

On peut alors passer à la suite pour la Manipulation 2.

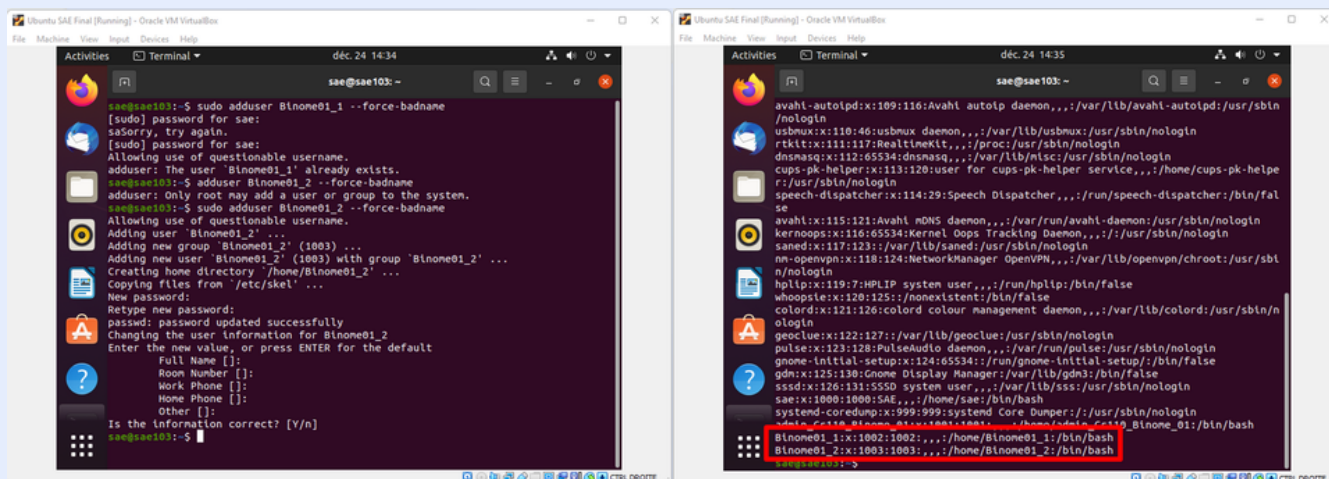
PRACTIQUE

Manipulation 2 (Création de deux utilisateurs) :

- 1) adduser Binome01_1 --force-badname
- 2) adduser Binome01_2 --force-badname

Explications :

- adduser : Créer un utilisateur sous Ubuntu
- Binome01_X : Le nom des utilisateurs que l'on va créer
- --force-badname : Utilisation de l'underscore, on doit forcer parce que l'expression régulière ne supporte pas ce caractère correctement.



The image shows two terminal windows from an Ubuntu VM. The left window shows the execution of the 'adduser' command for 'Binome01_1' and 'Binome01_2' with the '--force-badname' flag. The right window shows the output of the 'cat /etc/passwd' command, listing system users and regular users. The last two lines of the output are highlighted in red: 'Binome01_1:x:1002:1002:,,,:/home/Binome01_1:/bin/bash' and 'Binome01_2:x:1003:1003:,,,:/home/Binome01_2:/bin/bash'.

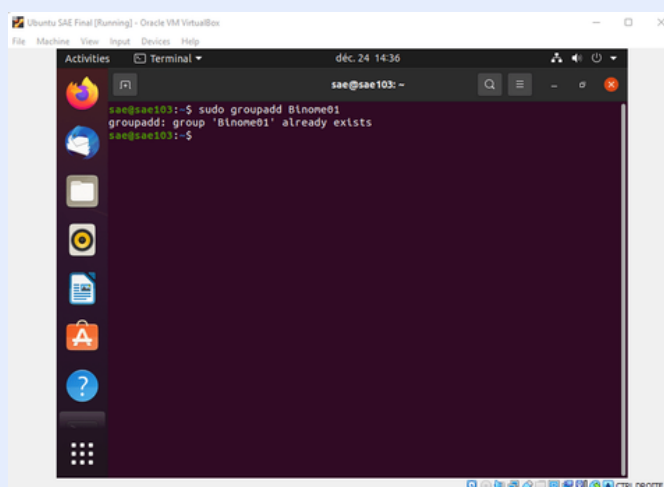
On regarde maintenant à l'aide de cat /etc/passwd si nos deux utilisateurs ont été créés. C'est bien le cas comme on peut le voir sur la capture d'écran de droite. On peut alors continuer et passer à la Manipulation 3 !

Manipulation 3 (Créer un groupe) :

- 1) groupadd Binome01

Explications :

- groupadd : Commande pour créer un groupe
- Binome01 : Le nom du groupe que l'on va créer



The image shows a terminal window where the 'groupadd' command is used to create the 'Binome01' group. The output shows that the group already exists.

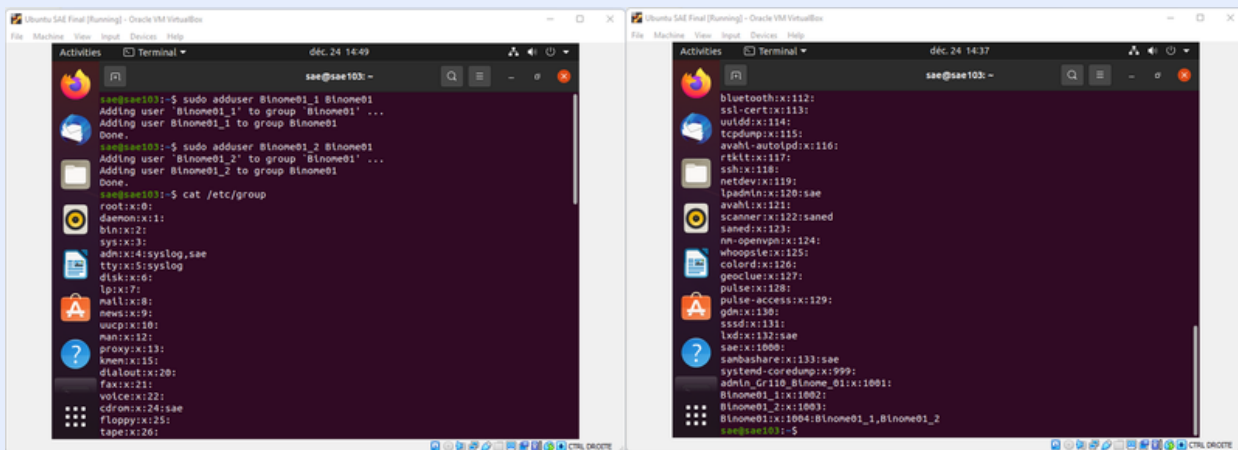
PRACTIQUE

Manipulation 4 (Ajouter des utilisateurs à un groupe) :

- 1) adduser Binome01_1 Binome01
- 2) adduser Binome01_2 Binome01

Explications :

- adduser : Créer un utilisateur sous Ubuntu (ici pour ajouter un utilisateur à un groupe)
- Binome01_X : Nom de l'utilisateur auquel on ajoute un groupe
- Binome01 : Nom du groupe auquel on va ajouter un utilisateur



The image shows two terminal windows from a virtual machine. The left window shows the execution of the `adduser` command twice, adding users `Binome01_1` and `Binome01_2` to the `Binome01` group. The right window shows the output of `cat /etc/group`, which lists the system groups and the two newly added users at the bottom.

```
sae@sae103:~$ sudo adduser Binome01_1 Binome01
Adding user 'Binome01_1' to group 'Binome01' ...
Adding user Binome01_1 to group Binome01
Done.
sae@sae103:~$ sudo adduser Binome01_2 Binome01
Adding user 'Binome01_2' to group 'Binome01' ...
Adding user Binome01_2 to group Binome01
Done.
sae@sae103:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,sae
tty:x:5:syslog
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:sae
floppy:x:25:
tape:x:26:
```

```
bluetooth:x:112:
ssl-cert:x:113:
uucdd:x:114:
tcpdump:x:115:
avahi-autoipd:x:116:
rtkit:x:117:
ssh:x:118:
netdev:x:119:
lpadmin:x:120:sae
avahi:x:121:
scanner:x:122:saned
saned:x:123:
nm-openvpn:x:124:
whooptex:x:125:
colord:x:126:
geoclue:x:127:
pulse:x:128:
pulse-access:x:129:
gdm:x:130:
sssd:x:131:
lxd:x:132:sae
sae:x:1000:
sambashare:x:133:sae
systemd-coredump:x:999:
admGr10_Binome01:x:1001:
Binome01_1:x:1002:
Binome01_2:x:1003:
Binome01:x:1004:Binome01_1,Binome01_2
sae@sae103:~$
```

On regarde encore une fois à l'aide de `cat /etc/group` si nos utilisateurs ont bien été ajoutés. C'est bien le cas, on peut alors passer à la suite !

INVITE DE COMMANDE

Manipulation 5 (Modifier l'invite de commande) :

On peut modifier l'invite de commande de plusieurs manières.

La première manière revient à changer une variable, `PS1` de `.bashrc` dans le répertoire de l'utilisateur.

La seconde manière revient à modifier depuis la console la couleur ou la police d'écriture.

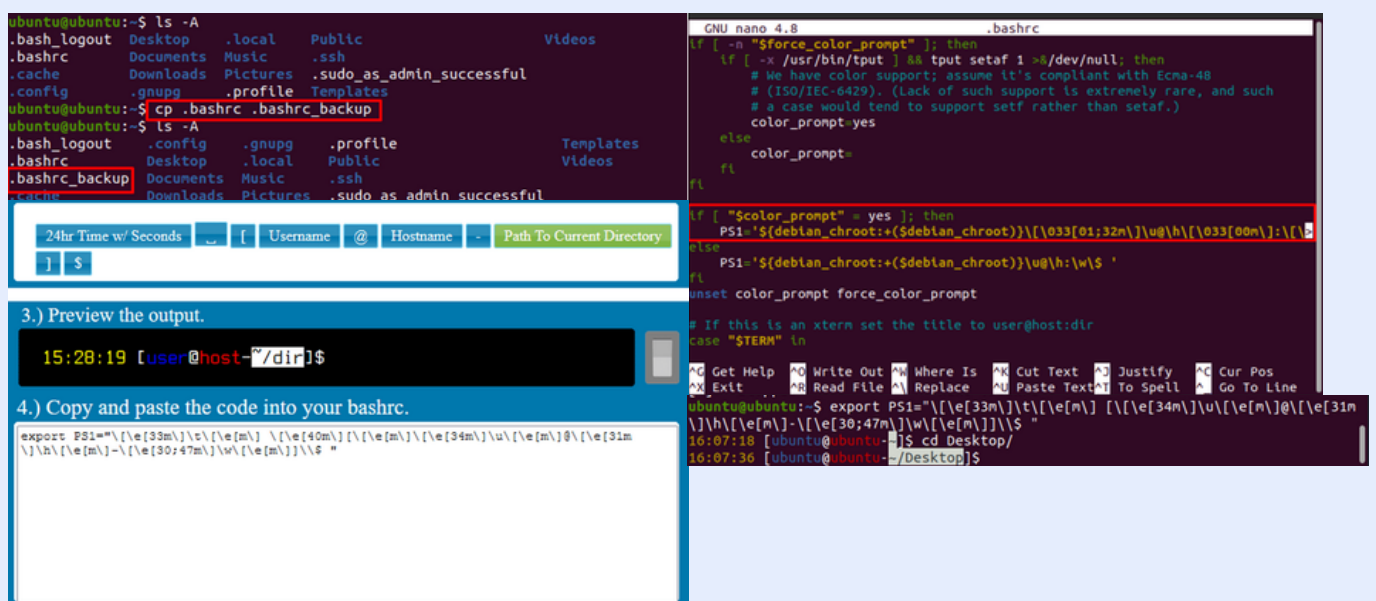
1ère méthode :

On va faire une backup de `.bashrc` au cas où l'on casserait quelque chose.

Ensuite, on regarde ce qu'il se trouve dedans. Si on descend, on verra PS1. Pour la modifier, on peut utiliser nano, ou bien export PS1 = "nouvelle valeur".

On se rend sur le site ezprompt.net pour générer notre PSI facilement.

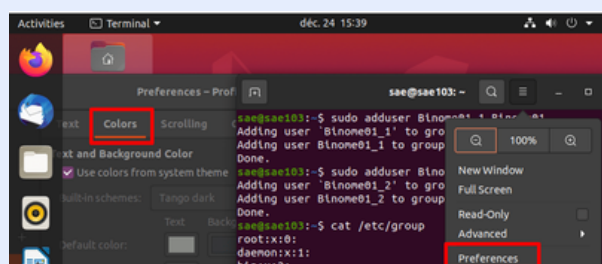
Il nous restera qu'à copier l'output du site et le copier dans la console. Au final, tout marchera !



2ème méthode :

On se rend simplement dans Preferences puis Colors par exemple. On pourra à partir de là modifier la couleur de l'invite de commande.

Manipulations de la semaine 3 finies !



GCC, SUBLIME TEXT

Manipulation 6 (Installer gcc) :

gcc c'est quoi? GNU Compiler Collection (gcc) est un ensemble de multiples compilateurs. Il nous servira à convertir du code en C/C++ par exemple en langage que la machine peut comprendre.

```
16:14:02 [ubuntu@ubuntu-~]$ gcc

Command 'gcc' not found, but can be installed with:

sudo apt install gcc

16:14:04 [ubuntu@ubuntu-~]$
```

On nous demande ici d'installer à l'aide de apt le package gcc.

Mais apt c'est quoi? Advanced Packaging Tool, est un système utilisé pour installer des "packages" facilement. C'est une commande très importante et donc à retenir !

Le programme nous demande de lancer la commande `sudo apt install gcc`.

Comme nous l'avons vu lors des premières manipulations, `sudo` nous donne les droits administrateurs (car oui on installe quelque chose, donc pour pas que tout le monde installe n'importe quoi, on doit avoir ces droits). Ensuite, `apt` est l'outil utilisé pour installer `gcc`. `install` ici précise à `apt` que l'on veut installer un package et `gcc` est le package à installer. On lance la commande :

```
16:14:04 [ubuntu@ubuntu-~]$ sudo apt install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc-9 libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1 linux-libc-dev
  manpages-dev
Suggested packages:
  binutils-doc gcc-multilib make autoconf automake libtool flex bison gcc-doc
  gcc-9-multilib gcc-9-doc gcc-9-locales glibc-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-9 libasan5
  libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0
  libctf0 libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1
  linux-libc-dev manpages-dev
0 upgraded, 21 newly installed, 0 to remove and 183 not upgraded.
Need to get 3,514 kB/23.3 MB of archives.
After this operation, 107 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.3) ...
Setting up binutils (2.34-6ubuntu1.3) ...
Setting up gcc-9 (9.3.0-17ubuntu1-20.04) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...
Progress: [ 99%] [#####]
```

Toutes ces lignes sont entrain de nous montrer que l'on installe `gcc`. Maintenant il nous suffit d'attendre un peu.

GCC, SUBLIME TEXT

Manipulation 6 (Installer gcc) :

gcc c'est quoi? GNU Compiler Collection (gcc) est un ensemble de multiples compilateurs. Il nous servira à convertir du code en C/C++ par exemple en langage que la machine peut comprendre.

```
16:14:02 [ubuntu@ubuntu-~]$ gcc

Command 'gcc' not found, but can be installed with:

sudo apt install gcc

16:14:04 [ubuntu@ubuntu-~]$
```

On nous demande ici d'installer à l'aide de apt le package gcc.

Mais apt c'est quoi? Advanced Packaging Tool, est un système utilisé pour installer des "packages" facilement. C'est une commande très importante et donc à retenir !

Le programme nous demande de lancer la commande `sudo apt install gcc`.

Comme nous l'avons vu lors des premières manipulations, sudo nous donne les droits administrateurs (car oui on installe quelque chose, donc pour pas que tout le monde installe n'importe quoi, on doit avoir ces droits). Ensuite, apt est l'outil utilisé pour installer gcc. install ici précise à apt que l'on veut installer un package et gcc est le package à installer. On lance la commande :

```
16:14:04 [ubuntu@ubuntu-~]$ sudo apt install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc-9 libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd libctf0
  libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1 linux-libc-dev manpages-dev
Suggested packages:
  binutils-doc gcc-multilib make autoconf automake libtool flex bison gcc-9-multilib gcc-9-doc gcc-9-locales glibc-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-9 libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd
  libctf0 libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1 linux-libc-dev manpages-dev
0 upgraded, 21 newly installed, 0 to remove and 183 not upgraded.
Need to get 3,514 kB/23.3 MB of archives.
After this operation, 107 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.3) ...
Setting up binutils (2.34-6ubuntu1.3) ...
Setting up gcc-9 (9.3.0-17ubuntu1~20.04) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...

Progress: [ 99%] [#####]
```

Toutes ces lignes sont entrain de nous montrer que l'on installe gcc. Maintenant il nous suffit d'attendre un peu. On teste la commande gcc maintenant : C'est installé !

```
16:25:12 [ubuntu@ubuntu-~]$ gcc --h
Usage: gcc [options] file...
Options:
  -pass-exit-codes      Exit with highest error code from a phase.
  --help                Display this information.
  --target-help          Display target specific command line options.
  --help={common|optimizers|params|target|warnings|[*]}[{joined|separate|undoc
ted)][,...].           Display specific types of command line options.
                        (Use '-v --help' to display command line options of sub-processes).
  --version              Display compiler version information.
  -dumpspecs             Display all of the built in spec strings.
  -dumpversion           Display the version of the compiler.
  -dumpmachine           Display the compiler's target processor.
  -print-search-dirs     Display the directories in the compiler's search
                        path.
  -print-libgcc-file-name Display the name of the compiler's companion lib
```

GCC, SUBLIME TEXT

Manipulation 7 (Installer Sublime Text, un IDE) :

Un IDE c'est quoi? Integrated Development Environment, c'est un logiciel qui nous permet de développer un ensemble de langages informatique avec une interface graphique ce qui facilite le développement d'applications complexes. Nous avons choisi d'installer Sublime Text car c'est un IDE excellent pour développer dans de multiples langages.

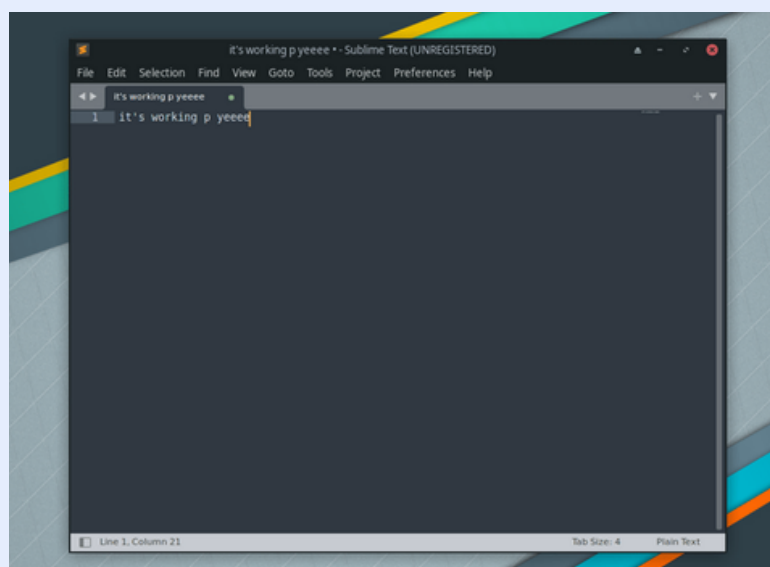
Nous nous rendons donc sur le site internet de Sublime Text et regardons où et comment l'installer :

```
1 - curl -O https://download.sublimetext.com/sublimehq-pub.gpg && sudo pacman-key --add  
sublimehq-pub.gpg && sudo pacman-key --lsign-key 8A8F901A && rm sublimehq-pub.gpg
```

```
2 - echo -e "\n[sublime-text]\nServer = https://download.sublimetext.com/arch/stable/x86_64" |  
sudo tee -a /etc/pacman.conf
```

```
3 - sudo pacman -Syu sublime-text
```

Nous avons alors exécuté ces 3 commandes sur notre seconde machine sous Manjaro, où nous allons faire les dernières manipulations.



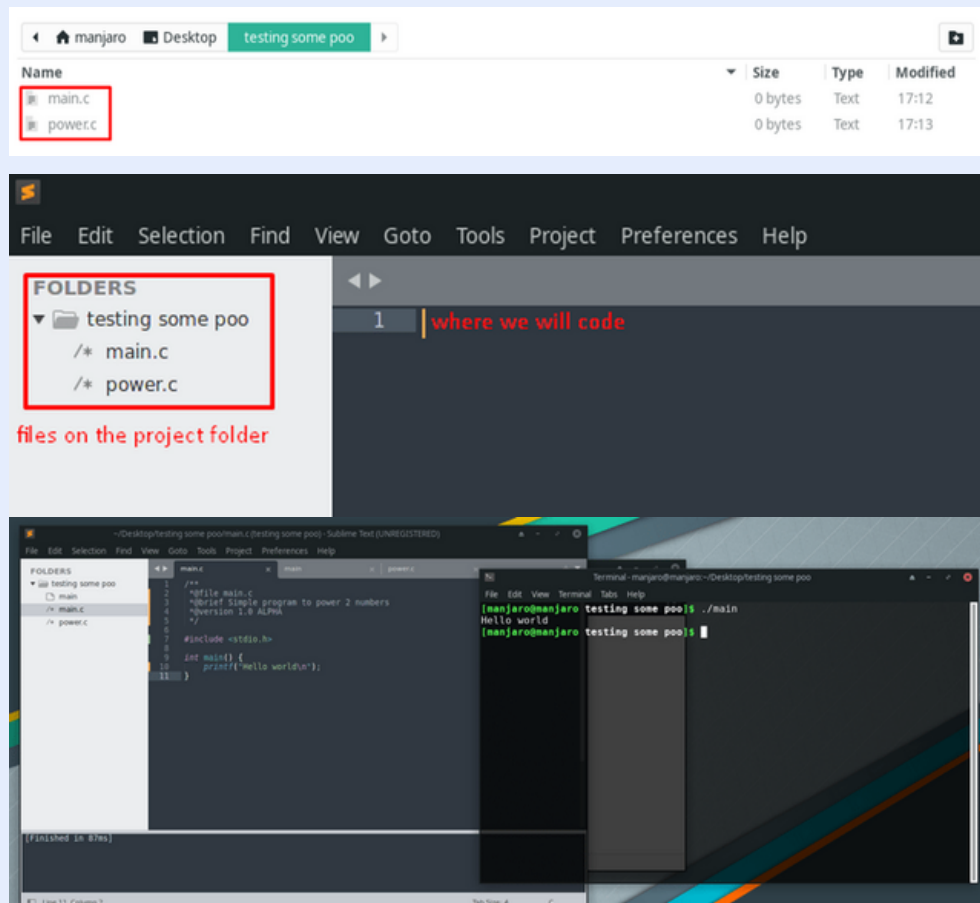
C'est bon c'est installé, nous pouvons continuer !

On a alors gcc et Sublime d'installé. On peut commencer à développer en compilation séparée.

Avant de continuer, c'est quoi la compilation séparée? La compilation séparée c'est le fait de faire plusieurs fichier .c, .cpp... qui seront ensuite transformés un par un en .obj pour object. Si l'on modifie un seul fichier, nous n'aurons qu'à recompiler et linker tous les .obj en eux. On a alors un gain de temps, et une meilleure lisibilité du code.

COMPILATION

Manipulation 8 (Compilation séparée) :

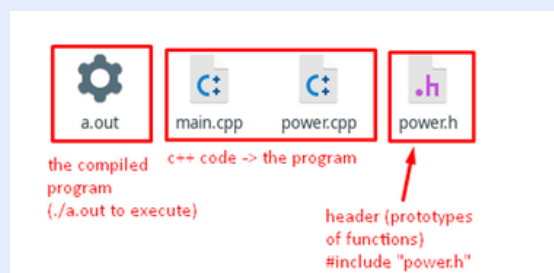


On crée alors deux fichiers .cpp et un .h pour commencer, où l'on va faire un petit programme qui va afficher Hello World, puis calculer la puissance d'un nombre et l'afficher aussi.

Pour compiler, nous avons utilisé la commande suivante : (car C++ et pas C)

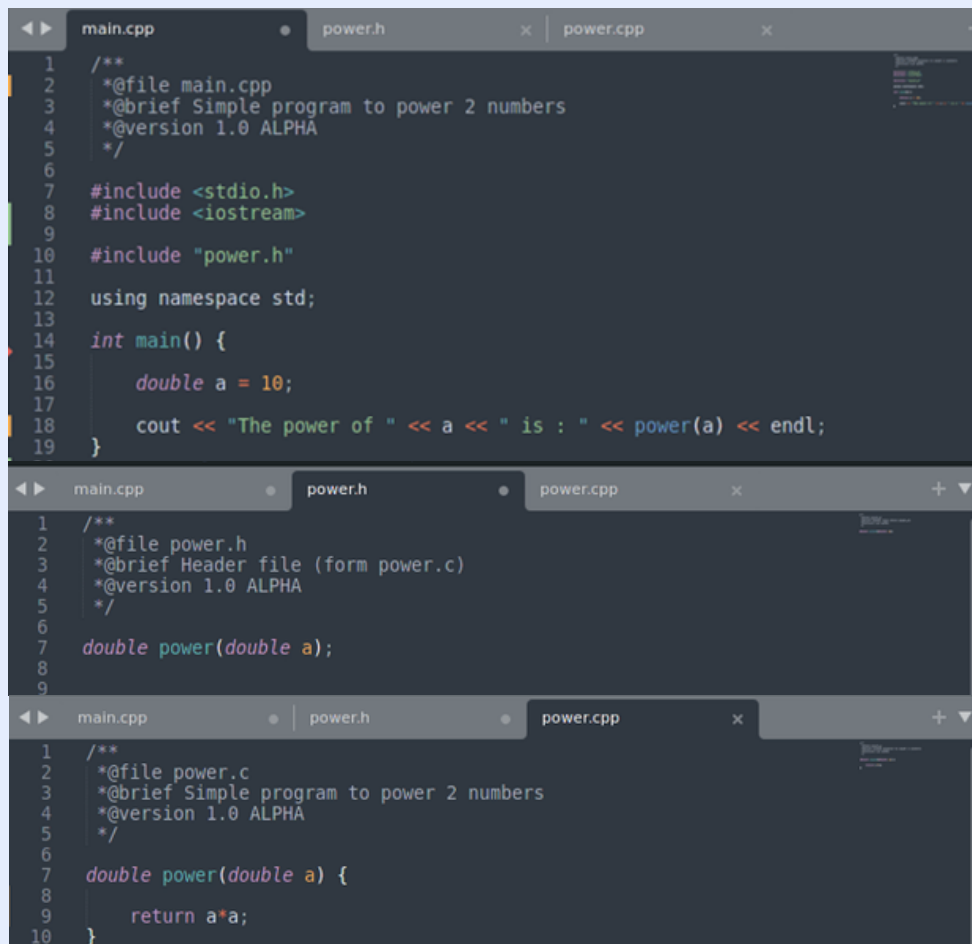
```
[manjaro@manjaro testing some poo]$ gcc -Wall main.cpp power.cpp power.h -lstdc++
```

On se retrouve alors avec un a.out qui est le programme. Pour le lancer, nous aurons à exécuter la commande suivante : ./a.out.



COMPILATION

Manipulation 8 (Compilation séparée suite) :



```
1  /**
2  * @file main.cpp
3  * @brief Simple program to power 2 numbers
4  * @version 1.0 ALPHA
5  */
6
7  #include <stdio.h>
8  #include <iostream>
9
10 #include "power.h"
11
12 using namespace std;
13
14 int main() {
15
16     double a = 10;
17
18     cout << "The power of " << a << " is : " << power(a) << endl;
19 }
```

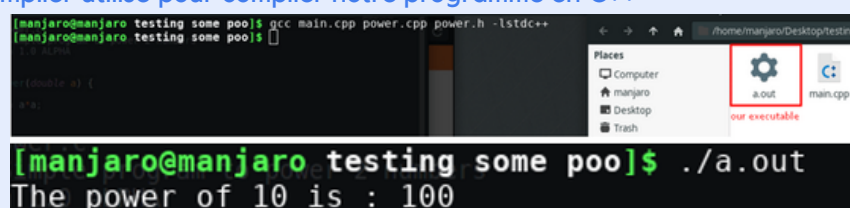
```
1  /**
2  * @file power.h
3  * @brief Header file (form power.c)
4  * @version 1.0 ALPHA
5  */
6
7  double power(double a);
8
9
```

```
1  /**
2  * @file power.c
3  * @brief Simple program to power 2 numbers
4  * @version 1.0 ALPHA
5  */
6
7  double power(double a) {
8
9      return a*a;
10 }
```

Pour compiler ce programme, nous avons utilisé la commande suivante (plus simple que celle au dessus) : `gcc main.cpp power.cpp power.h -lstdc++`

Explications :

- gcc : Programme utilisé pour compiler en C/C++
- main.cpp / power.cpp : Code source du programme
- power.h : Fichier header (les prototypes des fonctions de power.cpp)
- -lstdc++ : Le compiler utilisé pour compiler notre programme en C++



```
[manjaro@manjaro testing some poo]$ gcc main.cpp power.cpp power.h -lstdc++
[manjaro@manjaro testing some poo]$
./a.out
The power of 10 is : 100
```

Le programme marche, nous avons donc fini cette SAE avec succès !

CONCLUSION

Merci à vous d'avoir lu jusque là. Nous espérons que ce rapport vous ait plu.
N'hésitez pas à nous envoyer un message en cas d'oubli de notre part et à nous donner un feedback pour ce travail.
C'était très intéressant d'apprendre à installer une VM, des logiciels et utiliser les différentes commandes des manipulations.

Merci !

Un travail du Trinôme 01 (Groupe 110) constitué de :
BARRY ADONDE Ibrahim
HERILUS Allan
CHEVALIER Gaspard

Etudiants en BUT Informatique à l'IUT de Paris.