

Loué Emilien
VELIA Raphaël
S1A

SAE 1.03 Intro Système

Installation de poste



SOMMAIRE

Dossier d'étude et de choix des solutions	3
Un schéma de l'architecture logicielle	4
Installation de la machine	5-7
Notice d'utilisation	8-9
Bibliographie.....	10

Dossier d'étude et de choix des solutions :

Installation de la machine virtuelle :

Pour créer notre machine virtuelle, nous avons utilisé le logiciel VMware. Le système choisi est Linux car nous devons travailler sur une distribution de Linux. En parlant de la distribution, nous avons dû en choisir une parmi de nombreuses disponibles. En revanche nous avons des contraintes à respecter :

- La distribution doit être gratuite et toujours en service.
- Elle doit aussi avoir une bonne réputation et si elle est utilisée par beaucoup de monde et souvent, c'est toujours mieux.

On a donc choisi la distribution nommée : Ubuntu.

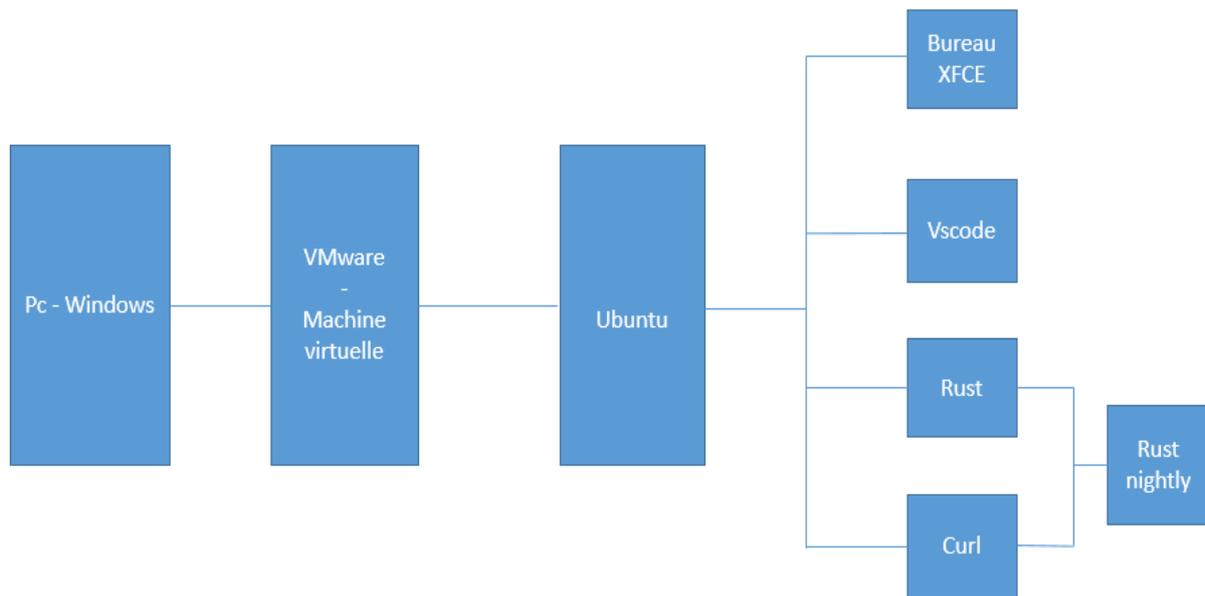
On a voulu prendre Ubuntu car c'est une distribution qui est souvent mise à jour (une tous les six mois), mais elle est également facile d'utilisation pour des débutants.

Lors de la création de notre machine on a dû lui donner un nom, on a donc choisi : Virtual machine. En sachant que nous n'allons pas avoir beaucoup de choses dans la machine virtuelle, nous avons laissé la taille maximum du disque par défaut, à savoir 20GB.

Au niveau de la configuration de la machine :

- La mémoire RAM de la machine a été définie à 2GB, au commencement on a voulu la mettre à 1GB mais au lancement de la machine, il était impossible de se déplacer sans avoir de la latence. Donc on a passé la RAM à 2GB.
- Ensuite nous avons mis dans l'onglet « New CD/DVD (SATA) » notre fichier iso pour remplacer un lecteur physique.
- Pour tout le reste de la configuration on a laissé les paramètres par défaut.

Un schéma de l'architecture logicielle :



Installation de la machine :

- Création de l'utilisateur Alice :

Le clavier de notre machine virtuelle était à l'initial en qwerty donc nous n'avons pas pu rentrer « alice@but » au début. Après le lancement de la machine virtuelle, notre clavier est repassé en azerty et nous avons donc pu modifier le mot de passe de Alice avec la commande suivante :

```
$sudo passwd alice
```

- Création de l'utilisateur Bob :

Une fois la machine virtuelle lancée avec l'utilisateur Alice, il nous a fallu ouvrir un terminal (donc en tant qu'administrateur) pour créer notre nouvel utilisateur : Bob. Pour ce faire nous avons utilisé cette commande :

```
$sudo useradd bob
```

Ensuite on a dû lui attribuer un mot de passe personnel et pour cela on a utilisé cette commande :

```
$sudo passwd bob
```

Nous lui avons donc attribuer le mot de passe : bob@but

Étant donné que nous sommes administrateur avec Alice, nous avons dû mettre avant la ligne « useradd bob » et « passwd bob » le mot sudo. Celui-ci indique que nous sommes en train de faire des modifications en tant qu'administrateur.

De plus comme Alice possède un répertoire, nous avons voulu en créer un pour Bob. Pour créer son répertoire nous avons fait ceci dans un terminal :

```
$sudo mkhomedir_helper bob
```

Le shell initial de Bob étant en sh, nous avons voulu le faire passer en bash pour pouvoir faire les étapes d'après concernant l'utilisateur Bob. Pour ce faire nous avons dû taper la commande suivante dans le terminal de Bob :

```
$chsh -s /bin/bash
```

Après avoir rentré la commande suivante, déconnectez-vous de la session de Bob, puis relancez-la et votre shell sera passé de sh à bash !

- Installation du bureau XFCE :

Ensuite nous avons installé l'environnement de bureau nommé : XFCE.

Pour ce faire nous avons utilisé cette commande :

```
$sudo apt install xubuntu-desktop
```

- Installation de Visual Code Studio :

Pour installer le logiciel Visual Code Studio nous avons utilisé la commande suivante :
\$sudo snap install –classic code

- Installation de git :

Pour faire le git de Bob, nous avons dû installer le logiciel git avec la commande suivante depuis le terminal de Alice :

\$sudo apt install git

Puis après l'installation de git, nous avons créé un répertoire où l'on clonera l'adresse du git adresser à Bob avec la commande suivante :

\$git clone <https://gitlab-ce.iut.u-bordeaux.fr/rgiot/hello.sae4.but1.iut.git>

- Adaptation du prompt avec les informations sur l'état du dépôt git du répertoire courant :

Pour avoir les informations du dépôt git, il faut tout d'abord chercher le fichier caché « bashrc » situé dans notre répertoire personnel. Ensuite on l'ouvre avec la commande suivante :

\$nano bashrc

Après l'avoir ouvert, nous allons rajouter les lignes de commandes suivantes à la fin de ce script :

```
# Configuration Git Branch - Modification prompt Linux
parse_git_branch() {
    git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/ (\1)/'
}
export PS1="\u@\h \[\033[32m\]\w\[\033[33m\]\$(parse_git_branch)\[\033[00m\] $ "
```

Dans la variable PS1, on a les propriétés suivantes :

- *\u@\h\[\033[32m\]* : Permet d'avoir le nom de l'utilisateur et le hostname de la machine.
- *\w\[\033[33m\]* : Permet d'avoir le nom du répertoire de travail actuel. Lorsque l'on changera de répertoire avec la commande cd, le nom du répertoire changera.
- *\\$(parse_git_branch)\[\033[00m\]* : Permet d'afficher la branche git dans laquelle on se trouve actuellement, seulement si ce répertoire est un repository git.

- Installation de Rust:

Comme demandé dans le sujet, nous avons dû installer Rust nightly. Pour ce faire nous avons dû dans un premier temps installer Rust et Curl. Curl nous est indispensable afin d'installer Rust nightly. Pour les installer nous avons dû rentrer les commandes suivantes dans le terminal de Alice :

```
$sudo apt install curl
```

```
$curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Maintenant que ces 2 logiciels sont installés, nous pouvons mettre Rust nightly. Pour changer Rust en Rust nightly il faut taper la commande suivante :

```
$rustup default nightly
```

- Installation de Cargo :

Pour ouvrir le script contenu dans le git de Bob, nous avons besoin du logiciel Cargo pour que son exécution fonctionne. Pour ce faire nous allons utiliser la commande suivante :

```
$sudo apt install cargo
```

Notice d'utilisation :

Création du projet Rust :

- Déjà assurez-vous que le logiciel Rust. Si celui-ci n'est pas encore installé, voir cf « Installation de la machine » et taper la commande associée à l'installation de Rust dans votre terminal. Puis nous allons voir si Rust est à jour avec la commande suivante :
\$sudo apt update (pour rechercher les mises à jour potentielles).
\$sudo apt upgrade (pour appliquer les mises à jour trouvées).
- Ensuite nous allons créer le projet Rust. Commencez par créer un répertoire où stocker notre projet Rust avec la commande suivante :
\$mkdir ~/projects (« projects » est le nom du répertoire).
- Puis déplacez-vous dans votre répertoire avec la commande « cd » et créer un nouveau répertoire nommé « hello_world » puis rendez-vous à l'intérieur de ce second répertoire. Une fois dedans, on va créer un script Rust que l'on va nommer « hello_world.rs » puis on va le modifier avec la commande suivante :

```
$sudo nano hello_world.rs
```

Lorsque nous serons dans la modification du fichier, tapez les instructions suivantes :

```
fn main() {  
    println!("Hello, world!");  
}
```

Enfin pour sortir du script et le sauvegarder il faut faire CTRL+X puis Y pour confirmer la sauvegarde et ENTRÉE pour sortir. Maintenant nous devons compiler et lancer le script avec ces commandes :

```
$rustc hello_world.rs  
$/hello_world
```

Une fois cela fait nous verrons apparaître ce message : *Hello, world!*

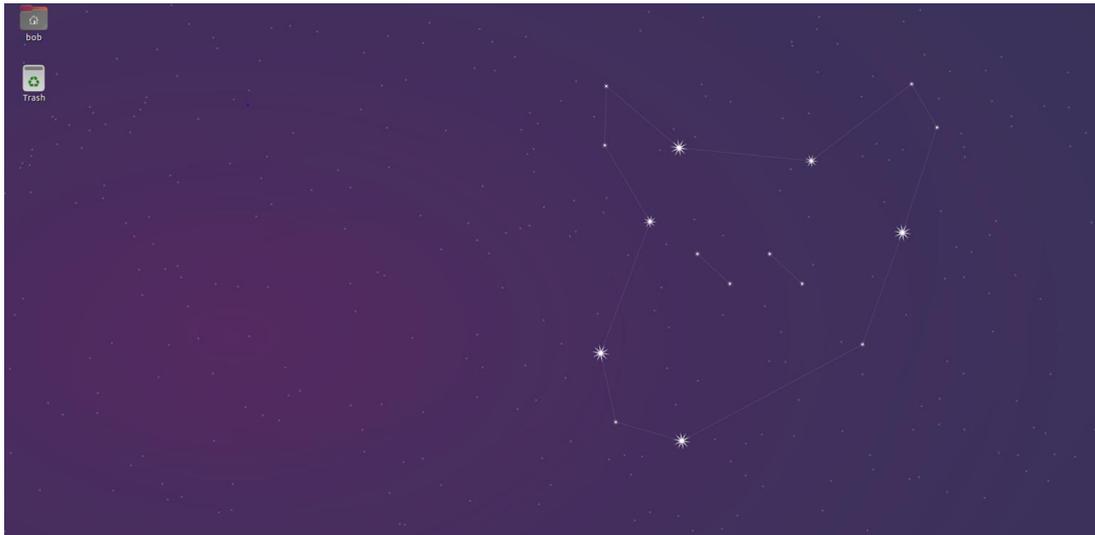
Vérification du test d'installation :

Malheureusement, par un manque de stockage sur notre répertoire personnel, nous n'avons pas pu tout installer d'un seul coup. La capacité de stockage de la machine virtuelle avec l'installation de Rust et de XFCE ne tient pas sur les Go disponibles de notre répertoire à l'IUT. A chaque tentative, la machine virtuelle s'éteint et refuse de se rallumer, nous faisant perdre toute notre avancée. Pour tester le script, nous avons besoin du logiciel « Cargo » et de « Rust nightly ». On peut faire l'installation d'un des deux logiciels, mais les 2 simultanément nous fait tout simplement crash notre machine virtuelle.

Portfolio :

Durant la réalisation de la SAE, nous avons découvert les rudiments pour créer une machine virtuelle et s'en servir à notre guise. Malgré certaines étapes assez compliquées où l'on devait faire énormément de recherches pour trouver une solution adéquate à notre problème. Par exemple, nous avons eu du mal à comprendre comment installer Rust nightly. Mais finalement, nous avons réussi à accomplir toutes les étapes du projet.

Environnement de bureau XFCE



Etat du dépôt git de notre répertoire courant

```
bob@alice-virtual-machine ~ $ cd projects/  
bob@alice-virtual-machine ~/projects $ cd helplo/  
bob@alice-virtual-machine ~/projects/helplo $ cd hello.sae4.but1.iut/  
bob@alice-virtual-machine ~/projects/helplo/hello.sae4.but1.iut (master) $ ls  
Cargo.toml  src  test.sh  
bob@alice-virtual-machine ~/projects/helplo/hello.sae4.but1.iut (master) $
```

Bibliographie :

- <https://korben.info/comment-changer-de-shell-sous-linux-ou-macos.html> : Changer de shell.
- <https://ubunlog.com/fr/visual-studio-code-editor-codigo-abierto-ubuntu-20-04/> : Installer Visual Code Studio.
- <https://www.atlassian.com/fr/git/tutorials/install-git> : Installer git.
- <https://www.it-connect.fr/linux-afficher-la-branche-git-dans-le-prompt/> : Afficher la branche du git dans le prompt de notre répertoire courant.
- <https://rustup.rs/> : Installer Rust et Curl.
- <https://www.oreilly.com/library/view/rust-programming-by/9781788390637/e07dc768-de29-482e-804b-0274b4bef418.xhtml> : Passer de Rust à Rust nightly.