

# Rapport Outils Libres

THIAM Amadou Moctar et KNORST Nicolas

Janvier 2022

## 1 Efficacité environnement de travail

### 1.1 Exercice 1 :

- Désactivation de notre souris :

# *xinput -list* → permet de lister les périphériques connectés ;  
# *xinput -set-prop "id" "Device Enabled" 0* → pour désactiver la souris en remplaçant *id* par la valeur affichée.

- Tableau de problèmes et de ses correctifs :

Problème	Correctif
Ouvrir le terminal sous Ubuntu	Ctrl + Alt + T
Démarrer Firefox ou Brave depuis le terminal	\$ firefox ou brave-browser
Saisir directement sur la barre d'URL	Ctrl + E
Page précédente / suivante	Alt + ← — Alt + →
Switcher entre les onglets du terminal (raccourcis créer)	Alt + ← — Alt + →

### 1.2 Exercice 2 :

- Nous avons choisi TypingClub comme site web pour améliorer notre saisie automatique ;
- Nous l'avons choisi parce que :
  - Une interface utilisateur belle et propre ;
  - Possibilité de s'entraîner pas à pas pour progresser à son rythme et avoir un aperçu de ses progrès ;
  - Possibilité de créer un compte et de sauvegarder ses progrès ;
  - Possibilité d'apprendre en fonction de la disposition de son clavier ;
  - Possibilité d'apprendre en mode jeu.
  - Capture de l'interface

jjkkkj kjkkj jkkj kkjk jjkkj kjkk kjkj kjkj jkkj



**RATATYPE**COURSTESTAPPRENDRELES COURS

Flash  
16:50

A stylized illustration of a hand holding a stopwatch. The hand is pink, and the stopwatch is blue and white. Radiating lines emanate from the stopwatch, suggesting speed or time.

### Passez un test de dactylographie gratuit

Tapez un court texte pour savoir à quelle vitesse vous pouvez taper, testez votre vitesse de frappe en mpm en français, anglais ou espagnol. Impressionnez amis ou employeurs avec votre propre certificat.

PASSEZ UN TEST DE DACTYLOGRAPHIE

### 1.3 Exercice 3 :

- Effectuer les tutoriels avec VIM :

```
=====
=  B i e n v e n u e  d a n s  l e  T u t o r i e l  d e  V I M  -  V e r s i o n  1.7  =
=====

Vim est un éditeur très puissant qui a trop de commandes pour pouvoir
toutes les expliquer dans un cours comme celui-ci, qui est conçu pour en
décrire suffisamment afin de vous permettre d'utiliser simplement Vim.

Le temps requis pour suivre ce cours est d'environ 25 à 30 minutes, selon
le temps que vous passerez à expérimenter.

ATTENTION :
Les commandes utilisées dans les leçons modifieront le texte. Faites une
copie de ce fichier afin de vous entraîner dessus (si vous avez lancé
"vintutor" ceci est déjà une copie).

Il est important de garder en tête que ce cours est conçu pour apprendre
par la pratique. Cela signifie que vous devez exécuter les commandes
pour les apprendre correctement. Si vous vous contentez de lire le texte,
vous oublierez les commandes !

Maintenant, vérifiez que votre clavier n'est PAS verrouillé en
majuscules, et appuyez la touche j le nombre de fois suffisant pour
que la Leçon 1.1 remplisse complètement l'écran.

-----
Leçon 1.1 : DÉPLACEMENT DU CURSEUR

** Pour déplacer le curseur, appuyez les touches h,j,k,l comme indiqué. **
      ^
      k
< h   l >   Astuce : La touche h est à gauche et déplace à gauche.
      j       La touche l est à droite et déplace à droite.
      v       La touche j ressemble à une flèche vers le bas.

1. Déplacez le curseur sur l'écran jusqu'à vous sentir à l'aise.
"/tmp/tutorCjH8Ng" 1038 lignes, 39314 caractères
```

- Paramétrage de GNU Readline :
  - `$ set -o vi`
- L'éditeur choisi est Vim
- Vim comme éditeur par défaut : dans le fichier `.bashrc` :
  - `export VISUAL=vim`
  - `export EDITOR="$VISUAL"`

#### 1.4 Exercice 4 :

- Regarder notre historique :
  - `$ history` ou bien `$ cat /home/user/.bash_history`
- Oui il y en a certaines commandes sensibles et la façon de gérer cela est d'empêcher ces commandes de l'historique bash lui-même
- Pour éviter que certaines commandes de l'historique n'apparaissent, dans le fichier `.bashrc`, on ajoute la ligne suivante :
  - `HISTIGNORE = ls cd pwd`

#### 1.5 Exercice 5 :

- Écriture d'une bash fonction `mkcd` :
 

Dans le fichier `.bashrc`, on ajoute les lignes suivantes :

  - `mkcd() {`  
`mkdir "$1" ;`  
`cd "$1" ;`  
`}`

- Écriture de la fonction `gitemergency` :

```

- gitemergency() {
  git add .;
  git commit -m "@";
  git push origin Head;
}

```

## 1.6 Exercice 6 :

- `$ touch backup.sh` → création du fichier backup
- `$ chmod +x backup.sh`
- Code du fichier `bash_completion` :

```

if ! shopt -oq posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi

```

- `$ source bash_autocompletion`

## 1.7 Exercice 7 :

- Installation de ZSH :
  - `(sudo) apt install zsh -y`
- Modification du prompt pour inclure les informations de vagrant :  
Les configurations sont à faire dans le fichier `.zshrc`, en spécifiant :
  - `ZSH_THEME=agnoster` → pour spécifier le nom du thème choisi ;
  - `plugin=vagrant-prompt.plugin.zsh` → choisir le nom du
  - Dans le fichier `.oh-my-zsh/themes/agnoster.zsh-theme`, on ajoute les lignes suivantes :

PROMPT='%{\$fg[NCOLOR]%}%B%n%b%{\$reset\_color%}:%{\$fg[blue]%}%B%c/%b%{\$reset\_color%}  
\$(vagrant\_prompt\_info)\$(svn\_prompt\_info)\$(git\_prompt\_info)%(!.#.\$)'

→ modification de variable de prompt ;

- Ajout des variables de couleurs :

```

ZSH_THEME_VAGRANT_PROMPT_PREFIX="%{$fg_bold[blue]%}["
ZSH_THEME_VAGRANT_PROMPT_SUFFIX="%{$fg_bold[blue]%}%{$reset_color%} "
ZSH_THEME_VAGRANT_PROMPT_RUNNING="%{$fg_no_bold[green]%}●"
ZSH_THEME_VAGRANT_PROMPT_POWEROFF="%{$fg_no_bold[red]%}●"
ZSH_THEME_VAGRANT_PROMPT_SUSPENDED="%{$fg_no_bold[yellow]%}●"
ZSH_THEME_VAGRANT_PROMPT_NOT_CREATED="%{$fg_no_bold[white]%}○"

```

### Exercice 9 :

- Installation de trois émulateurs de terminaux à savoir guake, roxterm et xterm:
  - *\$ sudo apt-get install -y guake roxterm xterm*
- Nous avons choisi Guake comme émulateur parce qu'il est : léger, rapide, multi-tab et hautement personnalisable.
- Suppression des autres :
  - *\$ sudo apt-get --purge remove -y roxterm xterm*

## 2 SSH :

### 2.1 Exerice 1 :

- `$ vagrant up` → Démarrer l'environnement Vagrant :
- `$ ssh bob@srv.local` → connexion avec l'utilisateur bob
- `$ ssh alice@srv.local` → connexion avec l'utilisateur alice
- `$ ssh carole@srv.local` → connexion avec l'utilisateur carole
- Vérification qu'on est sur la Vagrant :

<pre>bob@srv:~\$ hostname &amp;&amp; users srv bob bob@srv:~\$</pre>	<pre>alice@srv:~\$ hostname &amp;&amp; users srv alice alice@srv:~\$</pre>	<pre>carol@srv:~\$ hostname &amp;&amp; users srv carol carol@srv:~\$</pre>
--	--	--