

TP Outil Libre

Paradeis Arnaud - Boussetta Nael

Université de Lorraine — 13 janvier 2022



UNIVERSITÉ
DE LORRAINE



Charlemagne
Informatique

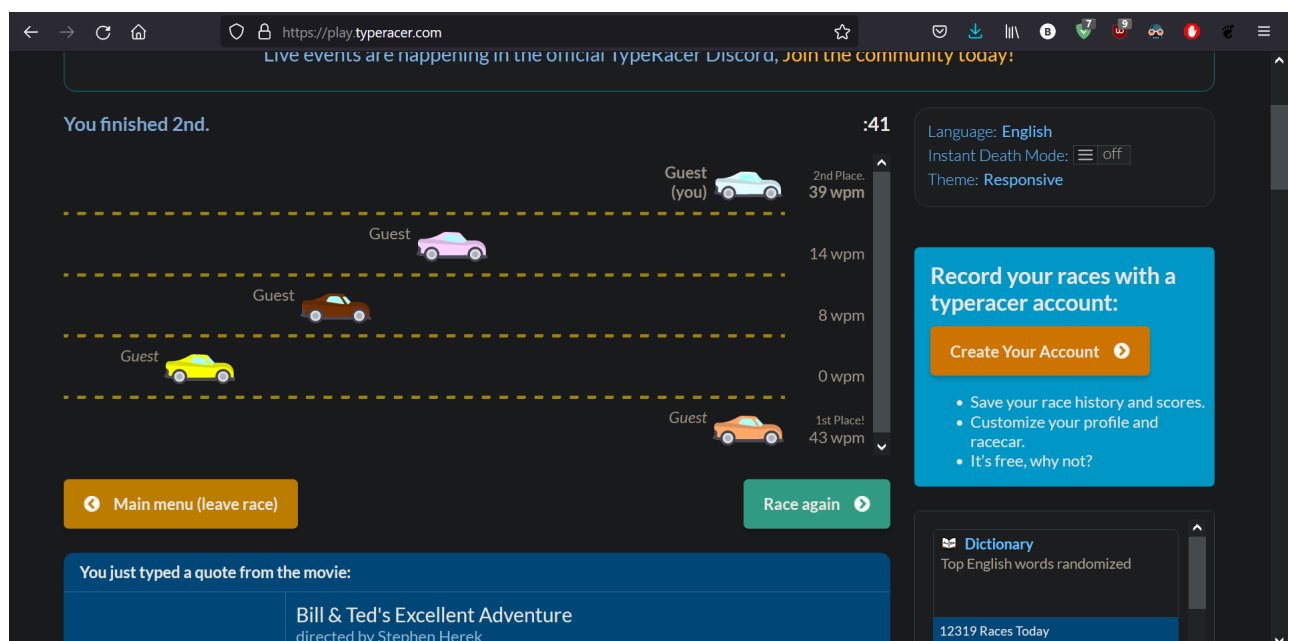
1 (Environment efficiency)

1.1 Tableau

Afficher la liste des périphériques	xinput list
Désactiver ou réactiver le périphérique	sudo xinput set-prop 13 "Device Enabled" 1 ou 0
Naviguer dans les fenêtres ouvertes	alt+tab
Rechercher les appli	Touche windows
Naviguer sur une page internet	tab
Fermer la page web	alt+f4
Changer d'onglet sur firefox	ctrl + tab
Ouvrir et fermer les onglets sur firefox	F10 + flèches directionnelles
Aller directement dans la barre de recherche	F6

1.2 Amélioration au clavier

Il existe le site <https://play.typeracer.com/>



1.3 Tutoriels Vim et Emacs

1.3.1 Avec emacs

1. Quitter emacs, ctrl + x puis ctrl + c
2. Descendre d'une page, ctrl + v
3. Monter d'une page, alt + v
4. Déplacer le cursor, ctrl + <Première lettre du mot de la direction> (Donc f pour forward b pour back etc...)

Pour que emacs soit l'éditeur par défaut : update-alternatives --config editor

1.4 History

Pour supprimer des éléments de mon historique : export HISTIGNORE=ls ou cd ou pwd

1.5 Alias

Command Line

```
function mkdcd () { mkdir "$1" && cd "$1" }
function gitemergency() {
git add .
git commit -a -m "$1"
git push
git status
}
```

1.6 Back up

sudo update-alternatives --config editor

Command Line

```
_backup() {
    local cur prev opts
    cur="${COMP_WORDS[COMP_CWORD]}"
    prev="${COMP_WORDS[COMP_CWORD-1]}"
    local files=("${cur}.*")
    case $COMP_CWORD in
        1) opts='getent passwd | cut -d: -f1';;
        2) opts="now tonight tomorrow";;
        3) opts="${files[@]}";;
        *);;
    esac
    COMPREPLY=()
    COMPREPLY=( $(compgen -W "$opts" -- ${cur}) )
    return 0
}
complete -o nospace -F _backup backup
```

1.7 ZSH Vagrant

Command Line

```
cd /home/arnaud/.oh-my-zsh/plugins/vagrant-prompt
sudo nano vagrant-prompt.plugin.zsh
sudo nano ~/.zshrc
~/oh-my-zsh/themes/robbyrussell.zsh-theme
```

On ajoute dans `/.oh-my-zsh/themes/robbyrussell.zsh-theme` :

```
PROMPT="%{?:%{$fg_bold[green]%}→ :%{$fg_bold[red]%}→ )"
PROMPT+= ' %{$fg[cyan]%}%c%{$reset_color%} $(git_prompt_info) $(vagrant_prompt_info) '

ZSH_THEME_GIT_PROMPT_PREFIX="%{$fg_bold[blue]%}git:(%{$fg[red]%})"
ZSH_THEME_GIT_PROMPT_SUFFIX="%{$reset_color%} "
ZSH_THEME_GIT_PROMPT_DIRTY="%{$fg[blue]%}) %{$fg[yellow]%}X"
ZSH_THEME_GIT_PROMPT_CLEAN="%{$fg[blue]%})"

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]%}○"
```

1.8 Raccourci ZSH

Dans le `/.zshrc` :

Command Line

```
function apache_start() {
if /etc/init.d/apache2 status > /dev/null; then
    echo "apache2 se stop"
else
    echo "le service apache2 est déjà stoppé"
fi
sudo service apache2 stop
}

zle -N apache_start
bindkey "^b" apache_start
```

1.9 Emulateurs de terminaux

1.9.1 Cool-Retro-Term

Ce terminal simule un terminal old school avec un style Fallout. Avec l'éditeur de texte « nano » il n'y a aucun code couleur avec ce terminal a part une typologie plus foncé lors des éléments importants. Avec l'éditeur « Vim » la typologie des éléments importants est un peu plus rouge. Ce terminal ne change pas grand-chose, c'est juste le style.

1.9.2 eDEX-UI

Fonctionnalités : Permet d'avoir plusieurs onglets en même temps, déplacement dans les dossiers depuis l'UI, informations sur l'utilisation de l'hardware, clavier virtuel pour écrans tactiles, différents thèmes pour différents goûts.

Rapidité : Assez rapide à lancer, mais réduit les performances

Documentation : Courte mais complète

Ecosystème : Puisqu'il s'agit d'un UI, le fonctionnement du shell reste en principe le même.

2 (SSH)

2.1 Connection

Ssh bob@10.0.0.2

On remarque que l'on n'a pas accès aux commandes lancées sur l'environnement Vagrant

2.2 Clés privées et publiques

ssh-keygen -b 4096 va ajouter une clé

Pour ajouter la clé à l'utilisateur de la machine distante : ssh-copy-id alice@10.0.0.2

Il nous faudra rentrer une passphrase pour la première connexion

Pour ajouter une clé publique manuellement sur notre machine hôte, afficher notre clé publique créée précédemment. Ensuite il faut se connecter à l'utilisateur de la machine souhaitée. Il faut avoir un dossier .ssh ainsi qu'un fichier authorized_keys dans le home de l'utilisateur. Ensuite, il faut copier la clé publique dans ce fichier.

Bonus :

Une passphrase est un "mot de passe" permettant de protéger une clé de cryptage La clé de cryptage est dérivée de la passphrase pour chiffrer la ressource à protéger.

En exécutant ssh-add, cela va nous permettre de mettre notre passphrase en mémoire pour éviter de la taper à chaque connexion.

2.3 Purge SSH

Pour éviter de rentrer une empreinte lors de la première connexion : il faut afficher l'empreinte de notre serveur sur le quel nous voulons nous connecter.

ssh-keyscan -H 10.0.0.3

On l'ajoute dans notre ~/.ssh/known_hosts. Cela nous permettra de nous connecter sans empreinte . Il faut créer notre fichier config dans .ssh

Command Line

```
Host bc # alias
Hostname 10.0.0.3 # ip machine
User bob # l'utilisateur au quel nous voulons nous connecter.
```

Après avoir exécuter ssh bc, nous serons directement connecté à bc avec l'utilisateur Bob.

2.4 SFTP

On se connecte en sftp à alice : sftp alice@10.0.0.2

get test # pour prendre un fichier de la machine virtuelle vers notre machine.

put test # pour prendre un fichier de notre machine vers la machine virtuelle.

Pour synchroniser des répertoires de notre machine virtuelle à notre machine hôte avec sshfs.

sshfs alice@10.0.0.2 :/home /tmp/alicecli
Le home de alice sera synchronisé et mis à jour dans notre répertoire /tmp/alicecli



Warning: Bien créer les répertoires avant, cette commande ne va pas les créer à votre place

Si nous modifions notre fichier paratagée que ce soit sur notre machine hôte ou notre machine virtuelle les changements seront effectués dans les 2 sens.

2.5 Tunnel SSH 1

Depuis notre local host pour se connecter a notre serveur en passant par la vm CLI
ssh -L 8000 :10.0.0.3 :80 bob@10.0.0.2



Warning: 10.0.0.3 est le serveur que nous voulons contacter et 10.0.0.2 est la machine par laquelle nous voulons passer pour se connecter au serveur

une fois connecté a notre CLI il faut ouvrir un autre terminal et via notre machine host
curl http ://localhost :8000/cgi-bin/test1.cgi et nous aurons comme résultat :

```
+ vagrant [~] cd
+ - curl http://localhost:8000/cgi-bin/test1.cgi
Connexion depuis: 10.0.0.2
Adresse de cli.local: 10.0.0.2
OK: la connexion est bien établie depuis l'adresse de la VM cli
+ - curl http://localhost:8000/cgi-bin/test1.cgi
curl: (7) Failed to connect to localhost port 8000: Connexion refusée
+ - curl http://localhost:8000/cgi-bin/test1.cgi
Connexion depuis: 10.0.0.2
Adresse de cli.local: 10.0.0.2
OK: la connexion est bien établie depuis l'adresse de la VM cli
+ -
```

Si nous coupons le tunnel sur bob@cli sur notre deuxième terminal le curl http ://localhost :8000/cgi-bin/test1.cgi ne fonctionnera plus.

2.6 Tunnel SSH 2

ssh -L 9000 :10.0.0.3 :80 bob@10.0.0.2
Pour créer le tunnel, ensuite dans le terminal de notre machine hôte
ssh -D 9000 bob@cli.local
Pour créer le port 9000,
Installer tsocks.
tsocks on
tsocks firefox
et dans l'url taper http ://srv,local/test1.....etc
et nous aurons accès à notre serveur via CLI .

2.7 X11 Forwarding

Sur le serveur on installe les paquets x server avec la commande : sudo apt install x11-apps
En local sur notre machine on se connecte en ssh : ssh -X bob@10.0.0.2
Quand nous sommes connectés sur le serveur nous pouvons exécuter une application graphique qui se lancera en local :
xeyes

2.8 Rebonds

2.8.1 ProxyJump

Il faut ajouter dans le fichier ~/.ssh/config

Command Line
Host bastion
Hostname 10.0.0.2
User bob
Host srv
Hostname 10.0.0.3
ProxyJump bastion
User bob

On se connecte avec ssh srv, en exécutant la commande who on voit l'adresse de CLI et non SRV

2.8.2 ProxyCommand

Command Line
Host bastion
Hostname 10.0.0.2
User bob
Host srvcmd
Hostname 10.0.0.3
User bob
ProxyCommand ssh bastion -W %h:%p

Cela revient au même avec ProxyCommand mais on laisse des traces sur le serveur passerelle.

2.9 BONUS

3 Git

3.1

On crée un repository dans le répertoire : git init
 On copie le vagrant file et dossier ssh dans le nouveau dossier
 On les envoie sur le repository : git commit
 On vérifie : git status
 Les 2 fichiers apparaîtront en rouge
 On fait git add des 2 fichiers
 vagrant up
 vagrant halt
 On vérifie le repository : git status
 Nouveau ,vagrant/ apparaît en rouge On crée le .gitignore dans le dossier
 Ajouter le .vagrant
 lors d'un git status le vagrant ne s'affiche pas en rouge.

3.2

On crée une nouvelle branche : git checkout -b branche
 On ajoute l'utilisateur Patrick et on installe php et apache
 On effectue les commit : git add -p Vagrantfile | git commit -m new "ajout du branche"
 On revient sur la branche main : git checkout master
 On vérifie que l'on n'a pas de modifications : git log
 Le working directory est comme on l'a laissé avant de créer la nouvelle branche
 On fusionne les branches : git merge branche

On inspecte le commit : `git log`
On a bien les modifications
On vérifie si la branche existe toujours : `git branch`
Elle existe toujours mais on en a plus besoin
On supprime la branche : `git branch -d branche`

3.3

On crée une nouvelle branche : `git checkout -b forward-new-port`
On modifie le VagrantFile et on commit : `git add Vagrant | git commit -m "ajout du port"`
On retourne sur main, on modifie le VagrantFile et on commit : `git add Vagrant | git commit -m "ajout du port"`
On merge : `git merge`
On aura donc un conflit, il nous suffit de modifier le fichier et de l'ajouter : `git add VagrantFile`
On vérifie que le conflit est résolu : `git status`
On termine le merge : `git merge --continue`

3.4