

Tech'Dej Oxxeo

Booster son environnement linux avec une suite d'utilitaires.



Sommaire

1. Introduction
2. Oh My Zsh
3. tmux
4. fd
5. eza
6. fzf
7. Outils divers
8. starship
9. Références

Introduction

Personnaliser son terminal avec ces outils, et booster votre productivité !

- **Oh My Zsh** - Le framework indispensable pour le shell **zsh**
- **tmux** - Un multiplexeur de terminaux
- **fd** - Une alternative plus simple et rapide que **find**
- **eza** - Un remplaçant de **ls**, plus "user friendly"
- **fzf** - Filtrer le flux de son entrée standard
- **bat** - Un clone de **cat** avec de la coloration syntaxique
- **zoxide** - Pour ajouter de l'intelligence à la commande **cd**
- **starship** - Un outil de personnalisation du prompt ultra-rapide

Oh My Zsh

Un framework pour zsh.

Définition [wiki Ubuntu.fr](#) :

Oh My Zsh est un framework open source pour la gestion de la configuration Zsh et est piloté par la communauté. Il est livré avec des tonnes de fonctions utiles au travers des plugins, des assistants, des thèmes.

Zsh est un interpréteur de commandes (shell), meilleur que bash sur la complétion et sa personnalisation.

Installation de zsh et son framework

```
1 # Installation de zsh
2 sudo apt install zsh
```

Oh My Zsh

Un framework pour zsh.

Définition wiki Ubuntu.fr :

Oh My Zsh est un framework open source pour la gestion de la configuration Zsh et est piloté par la communauté. Il est livré avec des tonnes de fonctions utiles au travers des plugins, des assistants, des thèmes.

Zsh est un interpréteur de commandes (shell), meilleur que bash sur la complétion et sa personnalisation.

Installation de zsh et son framework

```
1 # Installation de oh-my-zsh
2 sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Oh My Zsh

Un framework pour zsh.

Définition wiki Ubuntu.fr :

Oh My Zsh est un framework open source pour la gestion de la configuration Zsh et est piloté par la communauté. Il est livré avec des tonnes de fonctions utiles au travers des plugins, des assistants, des thèmes.

Zsh est un interpréteur de commandes (shell), meilleur que bash sur la complétion et sa personnalisation.

Installation de zsh et son framework

```
1 # Vérifier que zsh est bien installé
2 echo $SHELL
3 /usr/bin/zsh
4
5 # Fichier de configuration oh-my-zsh existe bien
6 ls ~/.zshrc
7 /home/chris/.zshrc
```

Oh My Zsh

Une tonne de plugins pour booster sa productivité.

Plugins disponibles gratuitement sur github : <https://github.com/ohmyzsh/ohmyzsh/tree/master/plugins>

En fonction de son métier et spécialités, on peut aller piocher le plugin correspondant, pour ajouter de la complétion par exemple.

Exemples de plugins :

- **zsh-autosuggestions** : pour suggérer les commandes à partir de l'historique de zsh
- **zsh-syntax-highlighting** : pour ajouter de la coloration syntaxique.
- **git** : ajouter des alias pour les commandes git et gagner en productivité
- **docker** : ajouter des alias pour docker
- **web-search** : alias pour effectuer des recherches rapides dans un navigateur
- **fzf** : pour filtrer le flux de son entrée standard
- **eza** : pour utiliser eza avec la commandes **ls**

tmux

Un multiplexeur de terminaux.

Avantages d'un tel outil :

- Scinder plusieurs terminaux au sein d'un seul affichage
- Sauvegarder sa session pour y revenir plus tard
- Afficher des informations complémentaires sur la machine où on est située

Exemples de raccourcis, après la commande Ctrl-B :

- % : scinder la fenêtre en 2 panneaux verticaux
- " : scinder la fenêtre en 2 panneaux horizontaux
- + : zoomer : le panneau
- c : créer une nouvelle fenêtre, dans un nouvel onglet
- d : détacher la session, pour y revenir plus tard
- w : afficher les fenêtres actuelles

fd

Une alternative plus simple et rapide que find.

Permet de rechercher des entrées plus rapidement et facilement qu'avec la commande **find**.

Exemples de commandes :

- **fd** : afficher toutes les entrées du dossier courant
- **fd slide** : afficher les entrées contenant slide
- **fd -e md** : afficher seulement les fichiers avec l'extension md
- **fd --type d** : afficher seulement les dossiers
- **fd --type f --exec wc -l** : afficher le nombre de lignes de tous les fichiers trouvés
- **fd -e md --exec cat|bat -l md** : afficher le contenu tous les fichiers md avec la coloration syntaxique

eza

Un remplaçant de **ls**, plus "user friendly".

Cet outil :

- offre plus de fonctionnalités que **ls** dans ces options
- permet de distinguer le type d'entrée avec des couleurs (fichier, dossier, lien symbolique)
- permet de voir plus facilement les droits des entrées
- plus rapide que **ls**

De plus, cet outil s'intègre bien avec **oh-my-zsh**, pour remplacer la commande **ls** par **eza**.

fzf

Filtrer le flux de son entrée standard.

fzf permet d'effectuer des recherches plus efficaces en sortie d'une commande **find**, **historique**, **process**, etc...

L'option **preview** permet d'avoir un aperçu du contenu d'un fichier, ou d'un dossier, lors de la recherche d'une entrée.

En configurant des raccourcis clavier, on peut remplacer la recherche dans l'historique, les fichiers Exemple :

- **Ctrl-r** : effectuer une recherche dans l'historique
- **Ctrl-t** : effectuer la recherche d'une entrée dans le dossier courant
- **Alt-c** : aller dans un dossier à partir d'une recherche préalable

Utilisé avec **zsh** et **tmux**, **fzf** devient encore plus intéressant.

Outils divers

bat, un clone de **cat** avec de la coloration syntaxique.

En plus de la coloration syntaxique, **bat** permet :

- d'intégrer un **pager** pour parcourir un fichier
- d'afficher les modifications si on est dans un projet git
- de voir plus facilement les caractères de fin de ligne et tabulations avec l'option -A
- et s'intègre bien avec fzf, fd, git ou man !

zoxide, pour sauvegarder les dossiers les plus utilisés, et y accéder plus facilement.

Peut être utilisé avec **fzf** avec la commande **zi**.

starship

Un prompt personnalisable à souhait et très rapide.

Alternative à **powerlevel10k**, un thème **zsh** populaire.


Pour l'installer :

```
curl -sS https://starship.rs/install.sh | sh
# on ajoute cette ligne dans le fichier ~/.zshrc
eval "$(starship init zsh)"
```

Toute la configuration de **starship** est à ajouter dans le fichier `~/.config/starship.toml`.

On peut alors ajuster les variables de chaque module, pour une expérience plus personnalisable du prompt.

Exemple lorsqu'on est dans un projet nodejs :

```
[nodejs]
format = "[ $version](bold green) "
detect_files = ["package.json", ".node-version"]
detect_folders = ["node_modules"]
```

Références

Oh My Zsh

- <https://ohmyz.sh/>
- <https://github.com/ohmyzsh/ohmyzsh/tree/master/plugins>

Tmux

- <https://doc.ubuntu-fr.org/tmux>
- <https://github.com/tmux/tmux/wiki>

fd

- <https://github.com/sharkdp/fd>
- <https://www.baeldung.com/linux/fd-find-alternative>

Références

eza

- <https://github.com/eza-community/eza>
- <https://eza.rocks/>

fzf

- <https://github.com/junegunn/fzf>
- <https://thevaluable.dev/fzf-shell-integration/>

bat

- <https://github.com/sharkdp/bat>
- <https://github.com/junegunn/fzf#preview-window>

Références

zoxide

- <https://github.com/ajeetsouza/zoxide>

starship

- <https://github.com/starship/starship>
- <https://starship.rs/>
- <https://starship.rs/config/>