

Arch Linux

Arch Linux est une distribution Linux réputée pour sa philosophie "Keep It Simple", offrant une approche minimaliste et hautement personnalisable pour les utilisateurs avancés. Lancée en 2002 par Judd Vinet, Arch Linux est conçue pour les utilisateurs qui souhaitent une maîtrise totale de leur système, avec une installation de base épurée que l'on peut configurer et optimiser selon ses besoins spécifiques.

La distribution se distingue par son système de gestion de paquets *pacman*, qui simplifie l'installation, la mise à jour et la gestion des logiciels, tout en proposant un modèle de publication en continu (*rolling release*). Cela signifie qu'Arch Linux reste constamment à jour, sans nécessiter de réinstallation lors de nouvelles versions majeures, ce qui en fait une option attractive pour ceux qui veulent toujours utiliser les dernières versions des logiciels.

Arch Linux ne fournit pas de préconfiguration ou d'environnement de bureau par défaut, ce qui donne aux utilisateurs la liberté d'assembler leur système à partir de zéro. Cette approche encourage une compréhension plus profonde des composants Linux, rendant Arch particulièrement prisée par les utilisateurs qui aiment expérimenter et apprendre.

En plus de la documentation exceptionnelle fournie par la communauté, le **Arch Wiki** est souvent considéré comme une ressource incontournable pour tous les utilisateurs de Linux, peu importe la distribution. Arch Linux est la quintessence de l'approche *DIY* ("Do It Yourself") dans l'univers Linux, idéale pour ceux qui veulent une expérience sur mesure et qui n'ont pas peur de mettre les mains dans le cambouis.

- Commandes et Config
 - Commandes Basique
- Services
 - FRP (Fast Reverse Proxy)

Commandes et Config

Les commandes Linux constituent l'essence même de l'interaction avec le système d'exploitation à travers un terminal ou une interface en ligne de commande (CLI). Elles permettent d'exécuter une multitude de tâches, allant de la navigation dans le système de fichiers à la gestion des processus, en passant par la configuration réseau et la gestion des utilisateurs.

Le système Linux repose fortement sur la philosophie UNIX, où chaque commande effectue une tâche spécifique. Ces commandes peuvent être combinées en utilisant des pipes et des redirections pour créer des scripts puissants et automatiser des processus complexes. Les utilisateurs de Linux apprécient la polyvalence et la puissance des commandes, qui offrent un contrôle total sur le système.

Qu'il s'agisse de simples commandes comme `ls` pour lister les fichiers ou de commandes plus complexes comme `grep` pour rechercher du texte à travers plusieurs fichiers, maîtriser les commandes Linux est une compétence précieuse pour les administrateurs système, les développeurs et les utilisateurs expérimentés.

Commandes Basique

Quelques commandes essentielles pour débuter :

- **ls** : Affiche la liste des fichiers et répertoires dans un répertoire donné. Exemples :
 - **ls** : Liste les fichiers dans le répertoire courant.
 - **ls -l** : Liste les fichiers avec des détails comme les permissions, la taille et la date de modification.
- **cd** : Change le répertoire courant. Exemples :
 - **cd /home/user/Documents** : Se déplace vers le répertoire `Documents`.
 - **cd ..** : Revient au répertoire parent.
- **cp** et **mv** : Copie ou déplace des fichiers/répertoires.
 - **cp file1.txt /home/user/** : Copie `file1.txt` vers le répertoire `/home/user/`.
 - **mv file1.txt /home/user/** : Déplace `file1.txt` vers le répertoire `/home/user/`.
- **rm** : Supprime des fichiers ou des répertoires. Attention à l'utilisation car cette commande ne place pas les fichiers dans une corbeille.
 - **rm file1.txt** : Supprime le fichier `file1.txt`.
 - **rm -r /home/user/dir** : Supprime récursivement le répertoire `dir` et son contenu.
- **chmod** et **chown** : Modifie les permissions et le propriétaire d'un fichier ou d'un répertoire.
 - **chmod 755 script.sh** : Donne les permissions de lecture, écriture et exécution au propriétaire, et les permissions de lecture et exécution aux autres.
 - **chown user:user file1.txt** : Change le propriétaire et le groupe de `file1.txt`.
- **ps** et **top** : Affiche les processus en cours.
 - **ps aux** : Affiche tous les processus en cours avec des informations détaillées.
 - **top** : Affiche les processus actifs avec une mise à jour en temps réel.
- **grep** : Recherche des chaînes de caractères dans des fichiers.
 - **grep 'mot' fichier.txt** : Recherche toutes les occurrences de `mot` dans `fichier.txt`.
 - **grep -r 'mot' /etc/** : Recherche récursivement `mot` dans tous les fichiers du répertoire `/etc/`.
- **find** : Recherche des fichiers dans un répertoire donné.
 - **find /home/user/ -name "*.txt"** : Recherche tous les fichiers `.txt` dans le répertoire `/home/user/`.
- **tar** : Archive et compresse des fichiers.
 - **tar -cvf archive.tar /home/user/dir** : Crée une archive `tar` de `dir`.
 - **tar -xvf archive.tar** : Décompresse l'archive `tar`.
- **ssh** : Se connecte à un serveur distant en utilisant le protocole SSH.

- `ssh user@ip_serveur` : Se connecte en SSH au serveur avec l'adresse `ip_serveur`.
- **sudo** : Exécute des commandes avec des privilèges administratifs.
 - `sudo apt update` : Exécute la commande de mise à jour des paquets avec les droits d'administration.

Configuration de base pour la gestion des commandes :

1. Bashrc et Aliases :

- `~/.bashrc` : Le fichier de configuration des commandes du terminal Bash où tu peux définir des alias pour simplifier les commandes répétitives. Par exemple :

```
bash
alias ll='ls -alF'
alias grep='grep --color=auto'
```

Ces alias te permettent de taper `ll` au lieu de `ls -alF` pour voir les fichiers détaillés, ou de colorer automatiquement les résultats de `grep`.

2. Gestionnaire de paquets :

- Selon ta distribution, tu utiliseras des commandes comme `apt` (Debian/Ubuntu) ou `pacman` (Arch) pour gérer l'installation, la mise à jour et la suppression de logiciels. Par exemple :
 - Sur Ubuntu : `sudo apt install package_name`
 - Sur Arch : `sudo pacman -S package_name`

3. Scripting Bash :

- L'écriture de scripts Bash permet d'automatiser des tâches courantes. Par exemple, un script simple pour sauvegarder un répertoire pourrait ressembler à ceci :

```
bash
#!/bin/bash
tar -czvf backup.tar.gz /home/user/dossier
echo "Sauvegarde terminée."
```

Services

FRP (Fast Reverse Proxy)

📌 Mettre en place un tunnel FRP

📌 Objectif

Configurer un tunnel sécurisé entre :

- 📌 **Serveur distant (ex : école)** inaccessible de l'extérieur
- 📌 **Serveur personnel (chez vous)** accessible via Internet

Le but est d'exposer un service local (ex : un site en localhost sur le port 3000) du serveur école via un domaine (ex : `mondomaine.fr`) en passant par le serveur personnel.

📌 Étape 1 — Télécharger FRP

Rendez-vous sur : <https://github.com/fatedier/frp/releases>

Téléchargez la dernière version pour Linux (ex : `frp_0.62.0_linux_amd64.tar.gz`).

Décompressez sur chaque machine :

```
tar -xvzf frp_0.62.0_linux_amd64.tar.gz
cd frp_0.62.0_linux_amd64
```

📌 Étape 2 — Configurer le serveur FRP (`frps`) sur le **serveur personnel**

Créez le fichier `frps.toml` :

```
[common]
bind_port = 7000
vhost_http_port = 8080
dashboard_port = 7500
dashboard_user = "admin"
dashboard_pwd = "admin"
authentication_method = "token"
token = "votreTokenSecret"
```

Lancer le serveur FRP :

```
./frps -c frps.toml
```

(Optionnel) Lancer via PM2 :

```
pm2 start ./frps --name frps -- -c ./frps.toml pm2 save pm2 startup
```

📌 Étape 3 — Configurer le client FRP (frpc) sur le **serveur distant**

Créez le fichier `frpc.toml` :

```
[common]
server_addr = "IP_DU_SERVEUR_PERSONNEL"
server_port = 7000
authentication_method = "token"
token = "votreTokenSecret"

[appweb]
type = "http"
local_port = 3000
custom_domains = "mondomaine.fr"
```

Lancer le client :

```
./frpc -c frpc.toml
```

(Optionnel) Lancer via PM2 :

```
pm2 start ./frpc --name frpc -- -c ./frpc.toml pm2 save pm2 startup
```

📦 Étape 4 — Configurer le DNS

Ajoutez un enregistrement DNS :

Type	Nom	Valeur	Proxy
A	mondomaine.fr	IP_DU_SERVEUR_PERSONNEL	<input type="checkbox"/> activé

📦 Étape 5 — (Optionnel) Reverse proxy avec Caddy

Si vous utilisez **Caddy** sur le serveur personnel :

```
mondomaine.fr {  
  reverse_proxy localhost:8080 {  
  }  
}
```

Recharger Caddy :

```
sudo systemctl reload caddy
```

📦 Vérification

1. Accédez à `https://mondomaine.fr`
2. Visitez le dashboard FRP (si activé) : `http://IP:7500`
3. Consultez les logs avec `pm2 logs frps` ou `frpc`

📦 Résumé

- 📦 Connexion sortante seulement depuis le serveur distant
- 📦 Tunnel HTTP sécurisé entre deux machines
- 📦 Service redirigé automatiquement depuis un domaine public

📦 Liens utiles

- Documentation officielle : <https://gofrp.org/en/>
- GitHub : <https://github.com/fatedier/frp>