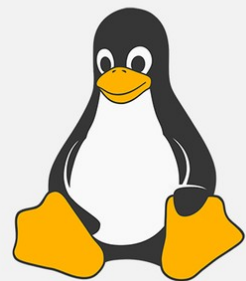


Introduction à Linux

Un guide pour débutants, intermédiaires et experts

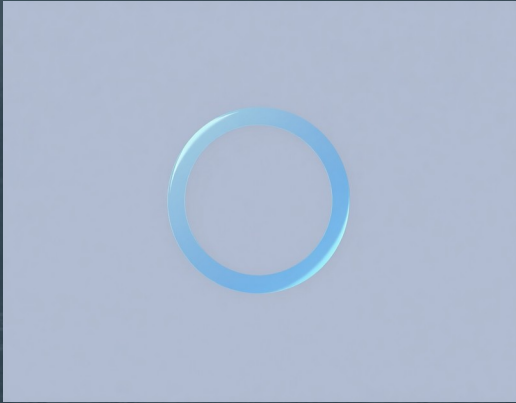


Linux™

Pourquoi choisir Linux ?

Gratuit et Open Source

Linux est entièrement gratuit. Son code source est ouvert, ce qui signifie que n'importe qui peut le voir, le modifier et le distribuer librement.



Sécurité et Stabilité

Réputé pour sa robustesse, Linux est moins sujet aux virus et malwares. Sa stabilité en fait le choix privilégié pour les serveurs et les systèmes critiques.



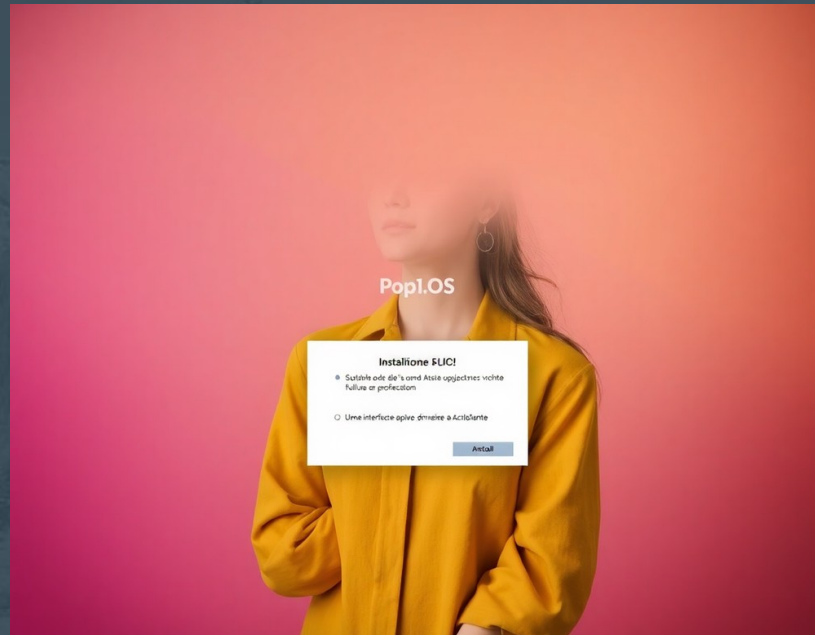
Personnalisation

Des environnements de bureau aux fonctionnalités du noyau, presque tout est personnalisable pour s'adapter parfaitement à vos besoins et à votre flux de travail.



Installer Pop!_OS : Un guide rapide

- Téléchargez l'image ISO depuis le site officiel de System76.
- Créez une clé USB amorçable avec un outil comme BalenaEtcher ou Rufus.
- Démarrez votre ordinateur à partir de la clé USB (changez l'ordre de démarrage dans le BIOS/UEFI).
- Suivez les instructions de l'installateur graphique simple et intuitif.
- Choisissez vos options de langue, de clavier, et de partitionnement (ou laissez l'installateur le faire).
- Redémarrez et profitez de votre nouvelle installation !



Premières commandes : Naviguer dans le terminal

Les bases pour les débutants

Commande	Description
ls	Lister les fichiers et répertoires dans le dossier actuel.
cd [répertoire]	Changer de répertoire (se déplacer vers un autre dossier).
pwd	Afficher le chemin du répertoire de travail actuel.
whoami	Afficher votre nom d'utilisateur actuel.
clear	Effacer l'écran du terminal pour plus de clarté.

Le terminal est votre outil le plus puissant sous Linux. Maîtriser ces commandes est la première étape pour devenir un utilisateur avancé.

Manipuler les fichiers et répertoires

Créer, copier, déplacer et supprimer (Débutant)

Gestion des Fichiers

- ``touch [nom_fichier]`` : Crée un fichier vide.
- ``cp [source] [destination]`` : Copie un fichier.
- ``mv [source] [destination]`` : Déplace ou renomme un fichier.
- ``rm [nom_fichier]`` : Supprime un fichier (attention, c'est permanent !).

Gestion des Répertoires

- ``mkdir [nom_répertoire]`` : Crée un nouveau répertoire.
- ``cp -r [source] [destination]`` : Copie un répertoire et son contenu.
- ``mv [source] [destination]`` : Déplace ou renomme un répertoire.
- ``rm -r [nom_répertoire]`` : Supprime un répertoire et son contenu.

Rechercher des fichiers : La commande ``find``

Niveau Intermédiaire

La commande ``find`` est un outil extrêmement puissant pour localiser des fichiers et des répertoires en se basant sur une multitude de critères.



Recherche par nom

```
`find . -name "*.log"`
```

Trouve tous les fichiers se terminant par `.log` dans le répertoire courant et ses sous-dossiers.



Recherche par type

```
`find /home -type d -name "Images"`
```

Trouve tous les répertoires (type `d`) nommés 'Images' dans `/home`.



Recherche par taille

```
`find . -size +50M`
```

Trouve tous les fichiers dont la taille est supérieure à 50 Mégaoctets.

Permissions et propriété des fichiers

Sécurisez vos données (Intermédiaire)

Comprendre les permissions

Chaque fichier a des permissions pour le propriétaire, le groupe et les autres. Elles se déclinent en lecture (r), écriture (w) et exécution (x). La commande `ls -l` permet de les visualiser.

```
rule 30.1--1399.45, .8100-.830, .1.10
type : . .41.601, 3..305.1 impl.03,
tirwe : . .11.5.1, 05.852.5.0025
ture }
```

Commandes Clés

- `chmod [permissions] [fichier]` : Change les permissions d'un fichier. Exemple : `chmod 755 script.sh` pour le rendre exécutable.
- `chown [utilisateur]:[groupe] [fichier]` : Change le propriétaire et le groupe d'un fichier.
- Les permissions sont fondamentales pour la sécurité et le bon fonctionnement d'un système multi-utilisateurs.

Écrire vos premiers scripts Shell

Automatisez vos tâches (Expert)

Structure d'un script

Un script shell est un fichier texte contenant une série de commandes. Il commence par `#!/bin/bash` (le 'shebang') pour indiquer quel interpréteur utiliser.

- Utilisez des variables pour stocker des données.
- Utilisez des conditions (if/else) pour la logique.
- Utilisez des boucles (for/while) pour répéter des actions.

Exemple de script

```
#!/bin/bash
# Script qui crée une sauvegarde datée

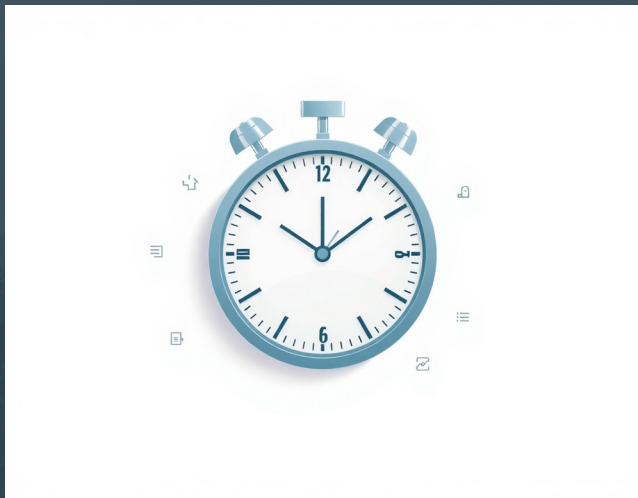
SRCDIR="/home/user/documents"
TGTDIR="/mnt/backups"
FILENAME="backup_$(date +%Y-%m-%d).tar.gz"

tar -czf "$TGTDIR/$FILENAME" "$SRCDIR"
echo "Sauvegarde créée : $TGTDIR/$FILENAME"
````
```

Rendez-le exécutable avec `chmod +x mon_script.sh`  
et lancez-le avec `./mon_script.sh`.



# Automatisation avec Cron



## Planifiez vos tâches (Expert)

Cron est un service qui exécute des tâches planifiées (scripts ou commandes) à des intervalles de temps spécifiés. C'est l'outil d'automatisation par excellence sur Linux.

- ``crontab -e`` : Pour éditer votre fichier de tâches cron (crontab).
- ``crontab -l`` : Pour lister les tâches cron actuelles.
- Syntaxe : ``minute heure jour_du_mois mois jour_de_la_semaine /chemin/vers/commande``
- Exemple : ``0 2 * * * /usr/bin/backup.sh`` exécute un script de sauvegarde tous les jours à 2h00 du matin.

# Conclusion et prochaines étapes

## Ce que nous avons appris

- Les avantages de l'écosystème Linux.
- Comment installer et naviguer dans le système.
- La gestion essentielle des fichiers et permissions.
- L'introduction au scripting et à l'automatisation.

## Pour aller plus loin

- Explorez différentes distributions (Arch, Fedora, Debian).
- Apprenez des outils de ligne de commande avancés comme ``sed``, ``awk`` et ``grep``.
- Contribuez à des projets open source sur des plateformes comme GitHub.
- Découvrez la conteneurisation avec Docker et Kubernetes.

# Questions & Réponses

