

5 - Logiciels, processus, services et journaux

Installer des paquets, surveiller les processus, piloter systemd et consulter les journaux.

- [5.1 - Installer des logiciels avec apt, pacman et dnf](#)
- [5.2 - Surveiller et arrêter les processus](#)
- [5.3 - Gérer les services avec systemctl](#)
- [5.4 - Consulter les journaux avec journalctl et dmesg](#)

5.1 - Installer des logiciels avec apt, pacman et dnf

Les gestionnaires de paquets installent les logiciels, leurs dépendances et les mises à jour depuis des dépôts configurés. Utiliser celui de sa distribution évite les installations dispersées.

Identifier la distribution

```
cat /etc/os-release
```

Debian et Ubuntu : apt

```
sudo apt update           # actualiser l'index
apt list --upgradable     # voir les mises à jour
sudo apt upgrade          # appliquer les mises à jour
apt search nom
apt show paquet
sudo apt install paquet
sudo apt remove paquet
sudo apt purge paquet     # retirer aussi sa configuration système
sudo apt autoremove       # dépendances devenues inutiles
```

Retrouver le paquet d'un fichier installé :

```
dpkg -S /usr/bin/curl
dpkg -L curl
```

Arch Linux : pacman

```
sudo pacman -Syu                # synchroniser et mettre tout à jour
pacman -Ss nom                  # rechercher dans les dépôts
pacman -Si paquet               # informations dépôt
pacman -Qi paquet               # informations paquet installé
sudo pacman -S paquet           # installer
sudo pacman -Rns paquet         # retirer avec dépendances inutiles et configuration
pacman -Ql paquet               # lister les fichiers
pacman -Qo /usr/bin/curl        # propriétaire d'un fichier
```

Sur Arch, éviter les mises à jour partielles comme `pacman -Sy paquet` sans mise à niveau générale : elles peuvent rendre les bibliothèques incohérentes.

Pour les paquets AUR avec `yay`, examiner le `PKGBUILD` avant la compilation :

```
yay -S nom-du-paquet
```

Fedora : dnf

```
sudo dnf check-upgrade
sudo dnf upgrade
dnf search nom
dnf info paquet
sudo dnf install paquet
sudo dnf remove paquet
dnf provides /usr/bin/curl
```

Vérifier une installation

```
command -v curl
curl --version
```

Bonnes pratiques

- Lire la liste des paquets supprimés ou remplacés avant de confirmer.
- Ne pas mélanger sans nécessité dépôts stables, testing et dépôts tiers.

- Préférer les dépôts officiels ou une source vérifiée.
- Sauvegarder les configurations importantes avant une mise à niveau majeure.
- Ne pas interrompre une opération de paquet en cours.
- Sur un serveur, prévoir une fenêtre de maintenance et vérifier les services après la mise à jour.

Flatpak, Snap, AppImage et les conteneurs complètent le gestionnaire natif, mais possèdent leurs propres règles de mise à jour, d'isolation et de stockage.

5.2 - Surveiller et arrêter les processus

Un **processus** est une instance d'un programme en cours d'exécution. Il possède un identifiant unique, le PID, ainsi qu'un propriétaire et un état.

Lister les processus

```
ps
ps aux
ps -ef
ps -u "$USER"
ps -p 1234 -o pid,ppid,user,%cpu,%mem,etime,cmd
```

Rechercher par nom :

```
pgrep -a sshd
pidof sshd
```

Éviter `ps aux | grep nom` lorsque `pgrep -a nom` suffit.

Surveiller en direct

```
top
htop
```

Dans `top`, `P` trie par processeur, `M` par mémoire et `q` quitte. `htop` est plus convivial mais n'est pas toujours installé.

Résumé de la mémoire :

```
free -h
```

Sous Linux, une partie de la mémoire « utilisée » sert de cache récupérable. La colonne `available` est généralement plus pertinente que `free` seule.

Envoyer un signal

```
kill 1234          # SIGTERM : demande un arrêt propre
kill -15 1234      # identique
kill -9 1234       # SIGKILL : arrêt forcé par le noyau
pkill -TERM nom
```

Méthode recommandée :

1. vérifier le PID et le propriétaire ;
2. envoyer `SIGTERM` ;
3. attendre et contrôler ;
4. utiliser `SIGKILL` seulement si le processus ne répond pas.

`SIGKILL` empêche le programme de nettoyer ses fichiers temporaires ou de terminer une écriture.

Tâches au premier et à l'arrière-plan

```
commande-longue &
jobs
fg %1
bg %1
```

Suspendre le programme au premier plan avec `Ctrl + Z`, puis utiliser `bg` ou `fg`. `Ctrl + C` demande son interruption.

Garder une commande après la fermeture du terminal :

```
nohup commande-longue > commande.log 2>&1 &
```

Pour les tâches importantes, préférer un service `systemd`, `tmux` ou `screen`, qui offrent une gestion plus claire.

Priorité

```
nice -n 10 commande  
renice 10 -p 1234
```

Une valeur `nice` plus élevée réduit la priorité CPU. Des valeurs négatives demandent généralement des privilèges.

Pour un service, agir avec `systemctl` plutôt que tuer directement son processus : le gestionnaire peut sinon le relancer immédiatement.

5.3 - Gérer les services avec systemctl

Sur de nombreuses distributions, **systemd** démarre le système et gère les services. Une unité de service porte généralement un nom comme `ssh.service` ou `nginx.service`.

Examiner un service

```
systemctl status ssh
systemctl is-active ssh
systemctl is-enabled ssh
systemctl show ssh
```

Le nom peut être `sshd` au lieu de `ssh` selon la distribution :

```
systemctl list-unit-files --type=service | grep -i ssh
```

Démarrer, arrêter et recharger

```
sudo systemctl start nginx
sudo systemctl stop nginx
sudo systemctl restart nginx
sudo systemctl reload nginx
sudo systemctl reload-or-restart nginx
```

`reload` demande au service de relire sa configuration sans arrêt complet, s'il prend cette opération en charge. Après une modification, valider la syntaxe avant le rechargement :

```
sudo nginx -t && sudo systemctl reload nginx
```

Démarrage automatique

```
sudo systemctl enable nginx
sudo systemctl disable nginx
sudo systemctl enable --now nginx
sudo systemctl disable --now nginx
```

`enable` prépare le prochain démarrage mais ne lance pas forcément le service immédiatement. `--now` combine les deux actions.

Masquer un service

```
sudo systemctl mask nom.service
sudo systemctl unmask nom.service
```

`mask` empêche tout démarrage, même manuel ou demandé par une dépendance. À utiliser seulement lorsque cet effet est voulu.

État général

```
systemctl --failed
systemctl list-units --type=service --state=running
systemctl list-timers --all
```

Après modification d'une unité

```
sudo systemctl daemon-reload
sudo systemctl restart mon-service
```

`daemon-reload` est requis après modification d'un fichier `.service`, `.timer` ou d'une surcharge `systemd` ; il n'est pas nécessaire pour la configuration interne de l'application.

Créer une surcharge sans modifier le fichier fourni par le paquet :

```
sudo systemctl edit mon-service
```

Puis vérifier la configuration assemblée :

```
systemctl cat mon-service  
systemd-analyze verify /etc/systemd/system/mon-service.service
```

Redémarrer ou éteindre

```
systemctl reboot  
systemctl poweroff
```

Sur un serveur distant, confirmer qu'aucune opération de stockage ou mise à jour critique n'est en cours avant l'arrêt.

5.4 - Consulter les journaux avec journalctl et dmesg

Les journaux permettent de comprendre ce qui s'est produit avant de modifier la configuration. Avec systemd, `journalctl` interroge le journal central.

Journaux d'un service

```
journalctl -u ssh
journalctl -u nginx --since today
journalctl -u nginx -n 100
journalctl -u nginx -f
```

`-f` suit les nouvelles entrées ; quitter avec `Ctrl` + `C`.

Journaux du démarrage

```
journalctl -b          # démarrage actuel
journalctl -b -1       # démarrage précédent
journalctl --list-boots
```

Priorités importantes :

```
journalctl -p warning..alert -b
journalctl -p err -b
```

Filtrer dans le temps :

```
journalctl --since "2026-08-07 08:00:00" --until "2026-08-07 09:00:00"
journalctl --since "30 minutes ago"
```

Afficher sans pagination :

```
journalctl -u nginx --no-pager -n 50
```

Messages du noyau

```
dmesg -T  
sudo dmesg -T | tail -n 100  
sudo dmesg -T | grep -Ei 'error|fail|warn'  
journalctl -k -b
```

Les dates produites par `dmesg -T` peuvent être approximatives après une veille ou une modification de l'horloge ; `journalctl -k` fournit souvent un contexte temporel plus cohérent.

Journaux classiques

Selon la distribution et le service :

```
ls -lah /var/log  
sudo less /var/log/auth.log  
sudo less /var/log/syslog  
sudo less /var/log/messages
```

Espace occupé

```
journalctl --disk-usage  
sudo journalctl --vacuum-time=14d
```

La dernière commande supprime les journaux archivés plus anciens que 14 jours. Elle doit respecter la politique de conservation et les besoins d'audit.

Collecte de diagnostic

```
journalctl -u mon-service --since "1 hour ago" --no-pager > journal-service.txt  
systemctl status mon-service --no-pager >> journal-service.txt
```

Avant de partager un journal, rechercher et masquer mots de passe, jetons, adresses privées sensibles, noms d'utilisateurs et autres secrets.