6. Gestion des modules du noyau

Le noyau Linux est modulaire, ce qui signifie qu’il peut charger et décharger des composants (modules) dynamiquement en fonction des besoins du système. Cela permet une flexibilité accrue et une gestion optimisée des ressources matérielles et logicielles.

Affichage des modules chargés : lsmod

La commande lsmod permet de lister tous les modules actuellement chargés dans le noyau. Elle s’appuie sur les informations contenues dans le fichier /proc/modules.

Exemple d’utilisation :

lsmod | more # Affiche la liste paginée des modules

Chaque ligne affiche :

• Module : Nom du module chargé

• Size : Taille du module en mémoire (en octets)

• Used by : Nombre de processus utilisant ce module

Chargement d’un module : modprobe

La commande modprobe est utilisée pour ajouter dynamiquement un module au noyau. Elle gère automatiquement les dépendances et charge les modules requis.

Exemple d’utilisation :

sudo modprobe nouveau # Charge le module "nouveau" (pilote libre pour les cartes NVIDIA)

Si un module n’est pas trouvé, il se peut qu’il ne soit pas installé ou que sa configuration soit incorrecte.

Déchargement d’un module : rmmod

rmmod permet de supprimer un module du noyau. Il est utile pour désactiver un module sans redémarrer le système.

Exemple d’utilisation :

sudo rmmod nouveau # Décharge le module "nouveau"

⚠️ Un module en cours d’utilisation ne peut pas être supprimé directement. Dans ce cas, utilisez :

sudo modprobe -r nom\_du\_module # Désactive proprement le module en tenant compte des dépendances

Vérification des fichiers de modules

Les modules sont stockés dans le répertoire /lib/modules/$(uname -r)/kernel/ et sont chargés en fonction des configurations définies dans /etc/modprobe.d/.

Pour rechercher un module spécifique, utilisez :

find /lib/modules/$(uname -r) -type f -name "\*.ko" | grep "usb"

7. Configuration du noyau

Personnalisation du noyau

Il est parfois nécessaire de configurer ou recompiler le noyau pour optimiser ses performances ou ajouter des fonctionnalités spécifiques. Sur Debian, cette configuration est facilitée par plusieurs outils.

Étapes pour configurer le noyau :

1. Installez les paquets nécessaires :

sudo apt update && sudo apt install build-essential libncurses-dev bison flex libssl-dev

2. Récupérez le code source du noyau Debian :

sudo apt install linux-source

Le code source sera stocké dans /usr/src/linux-source-\*.

3. Décompressez et entrez dans le répertoire source :

tar -xvf /usr/src/linux-source-\*.tar.xz -C /usr/src/

cd /usr/src/linux-source-\*/

4. Utilisez l’un des outils suivants pour configurer le noyau :

• make menuconfig (interface ncurses en terminal)

• make xconfig (interface graphique nécessitant Qt)

• make gconfig (interface GTK)

make menuconfig

Cela permet d’activer ou désactiver des fonctionnalités avant compilation.

5. Compilez et installez le nouveau noyau :

make -j$(nproc)

sudo make modules\_install

sudo make install

6. Mettez à jour GRUB et redémarrez :

sudo update-grub

sudo reboot

Note : Debian propose des noyaux précompilés optimisés pour divers matériels, donc la recompilation n’est nécessaire que pour des besoins spécifiques.

8. Démarrage du noyau et GRUB

Rôle de GRUB

GRUB (Grand Unified Bootloader) est le gestionnaire de démarrage par défaut sur Debian. Il permet :

• De sélectionner un noyau spécifique au démarrage

• D’accéder à des options avancées (mode récupération, modifications des paramètres du noyau, etc.)

• De gérer plusieurs systèmes d’exploitation

Structure de GRUB sur Debian

• Fichier principal de configuration : /etc/default/grub

• Scripts additionnels : /etc/grub.d/

• Fichiers générés : /boot/grub/grub.cfg

Modification des options de démarrage

Pour modifier GRUB, éditez /etc/default/grub :

sudo nano /etc/default/grub

Par exemple, pour afficher le menu de sélection au démarrage :

GRUB\_TIMEOUT=5 # Temps avant démarrage automatique

GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet splash" # Options passées au noyau

Appliquez les changements :

sudo update-grub

Mode de récupération

Au démarrage, maintenez Shift (BIOS) ou Esc (UEFI) pour afficher GRUB et choisir le mode de récupération.

9. Système de fichiers et gestion de la mémoire

Le noyau Linux gère à la fois :

• La mémoire physique (RAM)

• La mémoire virtuelle (swap)

• Le système de fichiers (ext4, XFS, Btrfs, etc.)

Gestion de la mémoire

• Afficher l’utilisation de la mémoire :

free -h

• Vérifier l’état du swap :

swapon --summary

Systèmes de fichiers pris en charge

Debian utilise principalement ext4, mais supporte d’autres systèmes :

• XFS : Performant pour les grandes bases de données

• Btrfs : Supporte le snapshot et la compression

• ZFS : Avancé, avec gestion automatique des erreurs

Pour afficher les partitions et systèmes de fichiers :

lsblk -f

df -hT

10. Sécurité et contrôle d’accès

Le noyau intègre plusieurs mécanismes de sécurité :

• Capabilités Linux : Autorisations restreintes pour root

• SELinux : Sécurité renforcée (désactivé par défaut sur Debian)

• AppArmor : Contrôle l’accès des applications aux ressources

Utilisation d’AppArmor

Debian active AppArmor par défaut. Vérifiez son état :

aa-status

Activer ou désactiver un profil :

sudo aa-enforce /etc/apparmor.d/usr.sbin.sshd

11. Optimisation du noyau pour Debian

Debian optimise son noyau pour supporter divers matériels. Cependant, vous pouvez l’optimiser davantage :

1. Charger uniquement les modules nécessaires

Éditez /etc/modules pour charger des modules spécifiques.

2. Améliorer les performances du disque

echo noop > /sys/block/sda/queue/scheduler

3. Ajuster les paramètres réseau

Ajoutez dans /etc/sysctl.conf :

net.core.default\_qdisc=fq

net.ipv4.tcp\_congestion\_control=bbr

Appliquez avec :

sudo sysctl -p