Institut Universitaire de Technologie Nice Côte d'Azur

Département Réseaux et Télécommunications

**M2102**

*TP n°2*

03/03/22

# Noyau, Système de fichiers, droits utilisateurs

Utilisez createvm pour créer une machine Debian et connectez vous à la machine avec le nom d'utilisateur rt (mot de passe rt). Vous pouvez passer root dans un terminal en tapant la commande su (=superuser) et en mettant le mot de passe rt.

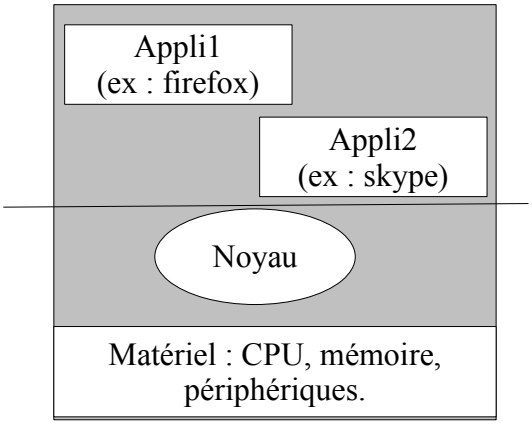
**Vous devez rendre un rapport par binôme à la fin de chaque séance.**

## I. Le noyau

### 1. *C'est quoi un noyau ?*

On distingue :

- Le noyau (ou kernel) qui est un programme chargé au démarrage de la machine et qui gère le matériel. C'est le cœur du système d'exploitation.
- Les applications des utilisateurs qui sont lancées par les utilisateurs et passent par le noyau pour accéder aux périphériques (écran, disque dur, carte réseau).




### 2. *Mais où est ce fameux noyau ?*

Dans le répertoire `/boot`. En effet on trouve ici plusieurs fichiers dont 2 ont une importance capitale pour le démarrage : « `vmlinuz` » et « `System.map` ».

Ces deux fichiers comportent en particulier des numéros de version. Cela permet de conserver plusieurs exemplaires différents du noyau et de choisir au démarrage la version que l'on désire. Ces fichiers peuvent aussi être les liens. Il faudra suivre ces liens si c'est le cas pour répondre à certaines questions ci-après.

Quels sont les noyaux présents dans le répertoire de démarrage (fichiers commençant par vmlinuz)?

Quelle est leur taille (utilisez la commande ls -lh)? Est-ce gros ? Pour répondre à cette dernière question, regardez la taille de différents programmes installés sur votre machine dans le répertoire /usr/bin. Pour cela, placez vous dans ce répertoire (commande cd) et listez par taille les exécutables (ls -lhSr | grep '\-x') et pensez que chaque programme rempli une seule fonction.

Détail de la commande ls -lhSr | grep '\-x' :

- ls liste les fichiers d'un répertoire. L'option -l affiche plus d'informations. L'option -h affiche les tailles en version « humaine », c'est-à-dire en Ko, Mo ou Go. L'option -S fait un tri (S=sort en anglais qui veut dire trier). L'option est r veut dire « reverse » et veut dire que le tri va aller du plus petit au plus grand alors que S le fait normalement dans le sens inverse.
- Le signe | signifie que l'on prend le résultat de la première commande et qu'on la donne à la seconde, ici grep.
- Grep est une commande qui cherche une chaine de caratère ici -x (qui signale qu'un fichier est un exécutable), dans un ensemble d'entrée, ici, ce qui a été retourné par ls -lshSr.

## *3. « Le noyau gère les applications »*

La gestion des applications et du matériel sont les deux principaux travaux du noyau. Une application est matérialisée par un ou plusieurs **processus**. Le noyau lorsqu'il travaille génère aussi des processus. On va donc discerner les processus en mode système (=du noyau) et ceux en modes utilisateurs (= des applications que vous lancez).

L'application top vous permet de voir l'ensemble des processus actifs et leur état : leur consommation en mémoire, en cpu.

Lancez la commande top depuis un terminal. Il y a 5 lignes d'informations générales suivies d'un tableau avec une ligne par processus.

Sur la première ligne, il y a l'heure puis le temps depuis lequel le système est actif (après 'up'). Reportez cette valeur dans votre rapport. Interprétez ce que veut dire cette valeur ?

La seconde ligne correspond aux processus (appélés aussi tâche=task en anglais) et leur état :

* Running (R dans la liste, colonne S pour State=Etat) veut dire que le processus s'exécute

* Sleeping (S) veut dire qu'il attend, par exemple que vous le réveilliez ou que le noyau lui apporte des données d'un disque.

*Stopped (T) veut dire qu'il est bloqué, par exemple par vous.

* Zombie (Z) on oublie pour l'instant....

Reportez le nombre de processus total et ceux dans chaque état.

Comptez ensuite vos processus. Pour cela, tapez u dans la fenêtre top puis tapez votre nom d'utilisateur rt ; mais si il y a trop de processus pour compter à la main , utilisez la commande :

```
ps -Ao user | grep rt | wc -l
```

La troisième ligne correspond à la consommation CPU. On trouve notamment les consommations en mode utilisateur (us=user), noyau (sy=système) et aussi le temps passé à ne rien faire (id=idle). Reportez ces valeurs ? Votre CPU se tourne-t-elle les pouces ou est-elle en train de bosser ?

La quatrième ligne correspond à la consommation mémoire. On trouve la mémoire totale, la mémoire utilisée par les processus et la mémoire non utilisée (free).

Reportez les valeurs et commentez (comme on l'a fait pour la CPU avant).

On va jouer un peu avec les processus. Ne regardez que les vôtres.

Quel est l'état du processus top (car top est un processus!) et pourquoi ?

Créons maintenant un processus faignant qui démarre et ne fait rien que dormir pendant 100s. Ouvrez un autre terminal et tapez sleep 100.

Quel est l'état du processus sleep dans le tableau? Cela semble raisonnable ?

On va maintenant interrompre momentanément ce processus. Placez-vous dans le terminal où sleep s'exécute. Si il a fini (plus de 100s), relancez le. Interrompez le en tapant les touches CrtL et Z simultanément (on appuie sur Crtl, on ne relâche pas la touche, puis on appuie sur Z).

Quel est le nouvel état du processus sleep ?

On va finir avec un processus gourmand en ressources : le processus yes. Dans le terminal où vous avez exécuté sleep, faites un man yes pour savoir ce que fait la commande yes, puis tapez yes et observez le processus dans la fenêtre top.

Quel est l'état du processus yes et quelles ressources (mémoire, processeur) consomme-t-il ? Montrez que c'est logique par rapport à ce que vous disait le man.

Enfin, une dernière chose importante est que le noyau organise les processus en arbres. Un processus a un père, un ou plusieurs fils et le noyau leur donne un numéro. Chaque processus a un numéro unique, appelé PID (process id) dans top.

Relevez le pid du processus top dans top lui-même.

Visualisez votre ou vos arbres de processus (avec leur pid) avec la commande pstree -up rt (si pstree est absent, l'installer avec apt install psmisc)

Quels sont les parents du processus top ?

Combien d'arbres avez-vous ? (Les arbres sont séparés par des lignes blanches)

Montrez que tous ces arbres sont des parties de l'arbre global des processus en tapant la commande pstree -p?

## 4. Le répertoire /dev ou Comment les périphériques sont exposés aux applications utilisateur?

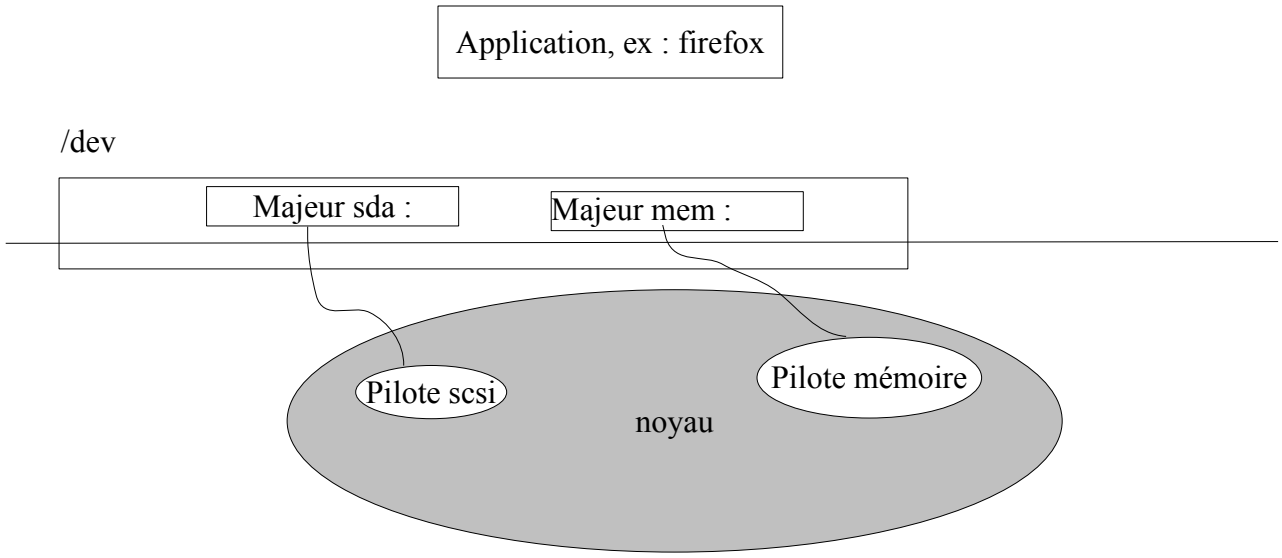
Les applications demandent l'accès aux disques, l'écran, le réseau et tout autre périphérique de l'ordinateur avec une interface particulière que le noyau met à disposition du mode « utilisateur ». On trouve ces interfaces sous forme de fichiers dans le répertoire `/dev`. C'est par ce biais là que le mode utilisateur fait ses requêtes au noyau.

Ce qu'il faut retenir c'est qu'en linux **un périphérique est vu comme un fichier**.

> Listez les fichiers dans le répertoire /dev en demandant les détails (il faut faire un ls -l et non un simple ls).

Pour chaque fichier, juste avant la date, on voit 2 numéros (parfois 1 seul) séparés par une virgule. Ce sont les « nœuds » majeurs et mineurs. Ils servent d'adresses au noyau pour communiquer avec l'espace « utilisateur ». Le majeur correspond au pilote (driver) que va utiliser le noyau. Par exemple le pilote pour l'écran, la carte son, les disques.

> Refaites le schéma ci-dessous dans votre rapport et complétez les numéros de majeurs correspondants sachant que les disques scsi sont nommés sda (pour scsi disk a - on peut avoir avoir b, c, d si il y a plus d'un disque), la mémoire est nommée mem.




On va maintenant montrer comment des programmes utilisateurs interagissent avec les périphériques.

Le terminal dans lequel vous écrivez est une partie de l'écran et est identifié dans /dev par un périphérique. Pour le trouver, il suffit de taper tty. Pour montrer que le terminal est vu comme un fichier, il faut montrer que lorsqu'on écrit dans ce fichier, cela s'affiche dans le terminal.

> Pour cette manipulation, il vous faut 2 terminaux. Récupérer le tty sur l'un des 2 et écrire dans le fichier correspondant avec la commande echo en faisant une redirection vers un fichier. Cela donne
>
> echo "coucou" > nom_du_fichier_tty_autre_terminal
>
> Que se passe-t-il ?

Certains périphériques sont virtuels car ils ne correspondent pas à des périphériques physiques, par exemple /dev/null. /dev/null est un trou noir qui absorbe tout ce qu'on y envoie sans le redonner.

Tapez :

echo "coucou" > /dev/null

Que se passe-t-il ?

Un autre périphérique virtuel est /dev/zero. Voyons voir ce qu'il fait :

Placez-vous dans votre répertoire racine et utilisez la commande dd qui créée un fichier de sortie (of = output file) à partir d'un fichier d'entrée (if = input file) :

dd if=/dev/zero of=fichierzero count=200 bs=512

Que trouve-t-on dans le fichier fichierzero créé. ? Attention, il y a un piège. Utilisez, pour répondre, les commandes :

cat fichierzero

et

hexdump fichierzero

qui vous donne directement les caractères ascii (sous forme hexadécimale) puis cherchez une table ascii sur Internet.

## *5. Le démarrage (boot)*

### a) L'historique du démarrage

Il est possible de voir ce que le noyau, lorsqu'il se charge, trouve comme périphérique et indique au moment où il met en service le pilote correspondant.

Examiner les messages du démarrage avec la commande dmesg en tapant :

dmesg | less

La commande précédente veut dire que les messages sont envoyés à less qui affiche page par page ce qu'on lui donne. A chaque fois que vous tapez sur la barre d'espace, less avance d'une page. Vous pouvez aussi utilisez les flèches pour remonter ou descendre.

Cherchez les messages liés aux disques scsci grâce à la commande :

```
dmesg | grep 's[dr][a-f][0-9]'
```

Cela correspond-t-il à ce qu'on a trouvé dans /dev en terme de disques scsi?

Regardons maintenant les informations sur la carte réseau

Quelles sont les cartes réseaux ?

`dmesg | grep eth`

Cela correspond-t-il à ce qu'indique ip address show qui liste les interfaces?

Cherchons des informations sur le bus PCI de la machine

Faites un :

dmesg | grep PCI

C'est un peu difficile à lire. Nous allons utiliser l'utilitaire lshw (qui fait un listing (ls) hardware). Si lshw n'est pas installé, apt install lshw

On peut trouver à quelle vitesse le bus traite les informations de la carte réseau avec la commande :

lshw -class network

Sachant la vitesse en MHz du bus et la largeur du bus (le nombre de bits envoyés à chaque), quel est la vitesse du bus en Mb/s ? Comparez la vitesse du bus avec la vitesse des cartes réseaux.

### b) Gestion du démarrage

Le démarrage des systèmes Linux est géré par GRUB (**GRand Unified Bootloader**). GRUB permet de donner le choix entre plusieurs systèmes d'exploitation ou versions d'un système.

Les fichiers de GRUB sont majoritairement dans le répertoire `/boot/grub/` : un petit coup d'œil dans le dossier permet de voir les nombreuses options possibles. Il fonctionne de manière très automatisée, avec le fichier `grub.cfg` qui est aussi dans ce répertoire. Ce fichier ne doit pas être édité à la main mais généré par la commande `update-grub` à partir des *modèles* présents dans `/boot/grub` et du fichier `/etc/default/grub`. Les options les plus souvent modifiées sont : le système à démarrer par défaut (GRUB_DEFAULT=0 indique le premier sytème du disque), le temps d'attente avant démarrage (GRUB_TIMEOUT), la taille du graphique de démarrage (GRUB_GFXMODE), la génération automatique des entrées de menu pour le mode « *single user* » (GRUB_DISABLE_LINUX_RECOVERY) et le fait de distinguer les disques par leur UUID plutôt que par leur nom système.

Ouvrir le répertoire `/boot/grub/` et examiner son contenu, puis examiner le fichier `/etc/default/grub`.

Modifiez le temps de démarrage par défaut (GRUB_TIMEOUT) en le mettant à 100s en editant le fichier `/etc/default/grub`, par exemple avec l'éditeur nano (tapez nano /etc/default/grub dans un terminal où vous êtes root).

Lancer la re-configuration avec la commande :

```
sudo      update-grub
```

et redémarrer la machine.

Comment voyez-vous que cela fonctionne ?

Quels sont les systèmes que `GRUB` peut démarrer sur cette machine ?

## 6. Pilotes, extensions : les modules ?

Si le noyau peut être considéré comme un gros exécutable, il est possible de lui ajouter des greffons (*plug-in*) (que l'on appelle ici modules) de manière dynamique, ce qui permet de modifier la liste des pilotes actifs en fonction des périphériques, sans redémarrer l'ordinateur.

Les modules actifs sont listés avec la commande lsmod.

Que vous donne la commande (faîtes un man wc si besoin) ?

```
lsmod | wc -l
```

Quel est le nom du module correspondant au driver de la carte réseau ? Pour le trouver, regarder les noms des modules et comparez avec les résultats obtenus avec lshw -class network

La liste des modules disponibles s'obtient avec la commande

```
less /lib/modules/3.2.0-3-amd64/modules.dep
```

```
(il faut peut-être ajuster le numéro 3.2.0-3 en fonction de
      la version du noyau)
```

Les modules eux-mêmes sont dans les fichiers avec une extension ko dans le répertoire /lib/modules et dans les sous-répertoires.

```
ls -ld /lib/modules/*
```

On peut les lister, depuis le répertoire /lib/modules/ avec la commande

```
find . -name *.ko
```

Comptez les en renvoyant la sortie du find vers une autre commande ? Laquelle ?

Quel est le répertoire dans lequel se trouve le pilote de la carte réseau ? Pour se faire, adaptez la commande find ci-dessus avec le nom du module de la carte à la place de *.