**Anneaux de protection**

Le processeur crée les anneaux de protections, pas le kernel.

* 0 : mode superviseur (kernel)
* 1
* 2
* 3 : user

En mode virtualisation, le 0 est l’hyperviseur et le 1 est le kernel.

**Périphériques**

* Mode caractère : flux d’octets
* Mode bloc : ex sda
* Network

1 numéro majeur (pilote)

1 numéro mineur (le périph)

Chaque périph possède un UUID.

**Kernel**

* Vmlinuz : image kernel
* System.map : symboles du noyau
* Initrd : chargement des drivers

Dépendance entre ces fichiers et régénérés à chaque compil.

**Modules**

**Disks**

* LVM : logical volume manager
* Partition classique

Partition = partie du DD destiné à accueillir un système de fichier

Table des partitions :

Partitionnement – création swap/FS (+ blanking = formatage)

**Authentification**

PAM : Pluggable authentication module

Entre l’utilisateur et le service demandé.

Conf des services utilisant PAM dans /etc/pam.d/

**Hardware**

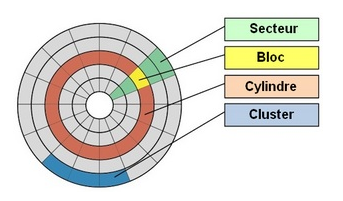
Un CPU peut avoir plusieurs core : un core = un ALU.

Hyperthreaded architecture : un seul ALU mais plusieurs étages fetch/décode afin de traiter virtuellement plusieurs instructions à la fois (2 instructions sont ramenés) mais un seul ALU : en fait on veut utiliser à 100% tous les étages tout le temps ;

SMP : symetric multi processor : plusieurs processus pour une seule même mémoire

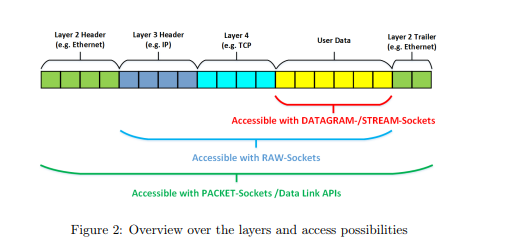
NUMA : non uniform memory access : une plage mémoire accordé à chacun des processeur : avec possibilité de mémoire partagée.

* HDD



**Network**

Enp0s1 : ethernet bus 0 slot 1



Kernel doit être compilé pour accepter l’utilisation des sockets RAW.

UNIX\_SOCKET : mode utilisateur

RAW\_SOCKET : couche 3

PACKET\_SOCKET : couche 2

Librairies : Libpcap, libnet.

IRQ ok quand peu de traffic, polling sinon avec buffer conséquent. (NAPI active polling au dela d’un certain seuil)

NIC envoie un signal à la réception d’un paquet.

* 1 ksoftIRQd s’occupe de tourner en boucle sur chaque CPU : permet de soulager l’interrupt handler qui va bloquer les autres IRQ.KsoftIRQ s execute à un haut niveau de PRIO mais pas autant que le handler d’IRQ, et il tourne avec IRQ activés.
* 1 structure softnet\_data crée sur chaque CPU (qui contient entre autre un poll\_list).

Linux utilise NAPI :permet d’éviter les tempêtes IRQ ( linux utilise les IRQ et au dela d’un certain seuil utilise NAPI : sondage du device ethernet toutes les X secondes : permet en gros de traiter plusieurs paquets d’un coup plutôt que un par un : technique du polling : mais attention si le buffer est trop plein : pertes de paquets !)