

# UE NSY103

## La Gestion de la Mémoire Linux



# Sommaire

I- Introduction

II- La Mémoire Physique

III- La Pagination

IV- La Segmentation Intel

V- La Segmentation Paginée



# I- Introduction

## La mémoire centrale des ordinateurs est destinée à héberger le code machine et les données des programmes en cours d'exécution

- La mémoire centrale est donc au cœur du fonctionnement d'un ordinateur, cependant la mémoire centrale est un composant relativement cher et on va donc en disposer de façon limitée et la stratégie d'allocation de la mémoire aux programmes en cours d'exécution, va constituer un élément clef de la performance des ordinateurs.
- En dehors du fait que nous ne disposons que d'une quantité de mémoire centrale limitée, cette quantité se voit généralement réduite pour satisfaire à un certain nombre de mécanismes annexes qui eux aussi ont besoin de la mémoire :
  - La table des vecteurs d'interruptions qui est installée en mémoire centrale au lancement du système.
  - La zone cache disque, qui permet d'accélérer l'accès aux disques durs en mémorisant les derniers secteurs utilisés.
  - Les registres d'entrée-sorties qui sur certains micro-processeurs font partie de (sont mappés sur) la mémoire centrale ; ce n'est pas le cas pour les processeurs de la famille Intel 80x86.
  - La mémoire allouée à la carte graphique si cette dernière ne possède pas de mémoire propre.
- ...



## II- La Mémoire Physique

La mémoire centrale des ordinateurs comporte un certain nombre d'octets et elle est découpée en mots mémoire.

- Les mots sont de taille fixe : 1 octet pour les premières générations de micro-processeurs et 8 octets (64 bits) pour les générations actuelles.
- Ci-dessous la représentation d'une mémoire avec mots de 64 bits :

octets									mots
0									0
8									1
16									2
24									3
32									4
48									5
56									6

### Adresse Physique

Les "n" octets de la mémoire sont ordonnés de 0 à "n-1" et pour accéder à un octet (le lire ou l'écrire) il faut utiliser son numéro d'ordre aussi appelé adresse physique.

Pour accéder à un mot entier (le lire ou l'écrire) il faut utiliser l'adresse physique de son premier octet.



# III- La Pagination

## L'adressage linéaire

L'espace mémoire d'un processus Linux comporte un certain nombre "m" d'octets ordonnés et chaque octet possède une adresse dite linéaire du premier octet "0" au dernier au dernier "m-1".

## La Pagination

- Pour partager la mémoire entre les processus en cours d'exécution, Les systèmes Linux découpent virtuellement la mémoire centrale en cadres de pages, chaque cadre étant constituée d'un même nombre d'octets (généralement 4KO).
- L'espace mémoire des processus est de son côté découpé en pages de mêmes dimensions que les cadres.
- Ainsi lorsque l'on va lancer un processus, seule sa première page de code sera chargée en mémoire (ou la première page + quelques pages), les autres pages seront mises en attente sur le disque dur dans une zone appelée zone de swap (cf. plus loin) et elles ne seront chargées lorsqu'elles seront appelées par le processus.
- Lorsqu'une page appelée par le processus n'est pas en mémoire une interruption "Page Fault" est levée par le composant MMU et le système d'exploitation va automatiquement charger cette page en mémoire à partir de la zone de swap et s'il n'y a plus de cadres de libre le système tentera de libérer des cadres selon des stratégies présentées sommairement ci-dessous dans le paragraphe "La Mémoire Dynamique".



# III- La Pagination

## Protection

La pagination Linux prévoit 2 types de protection pour les pages :

- 1) Le type niveau de privilèges : page accessible uniquement en mode noyau
- 2) Le type d'accès : page accessible en lecture seulement ou en lecture/écriture

## Avantages

- 1) Elle facilite le fonctionnement multitâche en rendant souple la gestion de la mémoire : Une page remplace très exactement une autre, il n'y a plus de gestion des "trous".
- 2) Elle permet de mettre en place des protections pour les pages

# III- La Pagination

## La mémoire Dynamique

- Le système va réserver un certain nombre de cadres pour les processus en mode noyau, les autres cadres que tous les processus vont pouvoir utiliser constituent "La mémoire Dynamique (Dynamic Memory) :
- Si le nombre de pages des processus en cours d'exécution ne dépasse pas celui des cadres de la mémoire dynamique, Linux fonctionne au maximum de ses performances.
- Si le nombre de pages des processus en cours d'exécution dépasse celui des cadres de la mémoire dynamique, Linux bascule certaines pages dans la zone de swap.
- La stratégie consiste à basculer les pages des processus en attente de ressources : Les pages seront basculées/rebasculées au fur et à mesure du déblocage/blocage des processus. Il pourra arriver cependant que des processus éligibles soient mis en attente en raison de l'indisponibilité des cadres mémoire.
- Si le nombre de pages des processus en cours d'exécution dépasse celui de "cadres mémoire dynamique + zone de swap", Linux va stopper l'exécution de certains processus en affichant le message d'erreur "Mémoire insuffisante".



# III- La Pagination

## L'Adresse Virtuelle / Réelle

L'utilisation de la pagination va nous amener à définir :

- Une adresse virtuelle processus qui est le couple "n° de page / décalage dans la page".
  - Une adresse réelle mémoire qui est le couple "n° de cadre / décalage dans la page".
- Lors de l'exécution il faudra convertir l'adresse linéaire processus en adresse virtuelle processus puis l'adresse virtuelle en adresse réelle mémoire.

## La Zone de Swap

- La zone de swap correspond à une partition indépendante "swap", il est conseillé de mettre en place une zone de swap de une à 2 fois la taille de la mémoire centrale.



# IV- La Segmentation Intel

**Les microprocesseurs Intel 80x86 utilisent un adressage segmenté c'est à dire que l'on gère la mémoire physique en tronçons appelés segments.**

- un processus utilise essentiellement 3 segments : Un pour le code, un pour les données et un pour la pile.
- Pour gérer le découpage de la mémoire physique en segments et la protection de ces segments les microprocesseurs Intel 80x86 utilisent les 2 composants MMU et MPU (Memory Management Unit et Memory Protection Unit). Ces composants ne sont accessibles qu'en mode noyau.
- En cas d'utilisation abusive des segments c'est à dire débordement du segment ou dépassement de droits d'accès (lecture / écriture / exécution), l'interruption "Segmentation Fault" sera levée par le composant MPU.
- Les microprocesseurs Intel 80x86 peuvent toujours fonctionner en mémoire non segmentée (mode real) mais ce mode est obsolète car il ne permet d'accéder qu'aux 1024 premiers octets de la mémoire.

## L'Adressage Logique

- Dans le cas de la segmentation (mode protected) l'adressage utilisé par l'assembleur Intel 80x86 est un adressage dit logique, une adresse logique correspond au couple "n° de segment / décalage dans ce segment".
- A l'exécution c'est le composant MMU qui va effectuer les conversions d'adresses logiques en adresse physiques.



# V- La Segmentation Paginée

**Linux a fait le choix de la pagination, parce que d'une part la gestion de la mémoire est plus performante (on n'a pas à gérer les trous) et d'autre part Linux est un système d'exploitation universel qui doit pouvoir s'implémenter sur des plateformes ne supportant pas la segmentation.**

- Cependant avec les microprocesseurs Intel 80x86 on est obligé d'utiliser le mode "protected" lequel implique la segmentation. C'est pourquoi les implémentations Linux sur plateformes PC vont mettre en place une segmentation paginée :
  - 1) On va utiliser un segment différent pour chacune des zones mémoire des processus Linux :
    - La zone Text qui contient le code et qui est protégée en écriture.
    - La zone Data et Bss qui contiennent les données et qui sont protégées en exécution et qui sont privées.
    - Les zones Pile et Tas qui sont privées.
  - 2) On va paginer chaque segment ainsi constitué

## Nouvelle Adresse Virtuelle

- L'adresse virtuelle qui était un couple "n° de page / décalage dans la page" va évoluer en un triplet " n° de segment / n° de page / décalage dans la page".
- A l'exécution c'est le composant MMU qui va effectuer les conversions d'adresses logiques en adresses virtuelles puis les conversions d'adresses virtuelles en adresses réelles.



# V- La Segmentation Paginée



Chaque processus possède une table des segments et chaque segment possède une table des pages, par exemple le processus "test" qui possède les 3 segments S1, S2 et S3

- La colonne "flag" indique si la page est actuellement chargée en mémoire.

Table des segments	
Taille en pages	Segment
5	S1
3	S2
4	S3

Table des pages		
Page	Cadre	flag
0	2	x
1		
2	10	x
3	13	x
4		

0	1	x
1	8	x
2	15	x

0	18	x
1	9	x
2		
3	20	x



# V- La Segmentation Paginée

Le système d'exploitation quant à lui maintient une table d'occupation de la mémoire centrale

Cadre	Page	flag disponible
0		x
1	test S2-01	
2	test S1-0	
3	autre processus	
4		x
5		x
6	autre processus	
7	autre processus	
8	test S2-1	
9	test S3-1	
10	test S1-2	
11		x
12	autre processus	
13	test S1-3	
14	autre processus	
15	test S2-2	
16		x
17	autre processus	
18	test S3-0	
19		x
20	test S3-3	



# EXERCICES

Vous pensez avoir bien assimilé les concepts présentés dans ce cours.  
Vous devez alors passer aux exercices, ce sont eux qui  
vous permettront de valider vos connaissances .