

# Systeme d'Exploitation

## TD2 : *Le poisson c'est bon, mangez-en!*

Jean-Christophe Deneuille  
<jean-christophe.deneuille@unilim.fr>

9 mars 2016

### Exercice 1 Allocation de la mémoire

Sur une architecture où le partitionnement de la mémoire est faite dynamiquement, nous avons l'état de la RAM est décrit par le tableau suivant :

10	10	20	30	10	5	30	20	10	15	20	20
----	----	----	----	----	---	----	----	----	----	----	----

*Les tailles sont en ko, et les blocs grisés sont ceux utilisés.*

Différentes requêtes d'allocation de mémoire sont faites dans cet ordre là : 20 ko, 10 ko, 5 ko, 25 ko.

1. À quelles adresses sont alloués les blocs si on utilise la politique *First Fit* ?
2. À quelles adresses sont alloués les blocs si on utilise la politique *Best Fit* ?
3. À quelles adresses sont alloués les blocs si on utilise la politique *Worst Fit* ?
4. À quelles adresses sont alloués les blocs si on utilise la politique *Next Fit* ?

### Exercice 2 La Pagination

1. Imaginez un système de pagination avec une table de pages stockée en mémoire.
2. Différents programmes utilisent la mémoire principale. Leur ordre d'allocation/désallocation est la suivante :
  - Allocation d'un espace A de 16ko
  - Allocation d'un espace B de 20ko
  - Allocation d'un espace C de 30ko
  - Désallocation de l'espace B
  - Allocation d'un espace D de 10koIndiquez, à chaque instant, l'état de la mémoire (logique et physique) de votre système.

### Exercice 3 Dur Dur ! (*Pagination à la demande*)

La méthode de pagination à la demande est la plus répandue des implémentations de mémoire virtuelle et demande de nombreuses ressources matérielles.

Ici, nous partons d'un système de swap où la mémoire est découpée en pages. Comme pour le swap, quand un programme doit être exécuté nous le chargeons en mémoire (phase de *swap in*) mais, au lieu de faire un swap complet, nous utilisons un « swappeur paresseux » (*lazy swapper*). Ce swappeur paresseux charge une page **uniquement si** elle est nécessaire.

1. Que ce passe-t-il alors quand le programme essaie d'accéder à une page qui est hors mémoire ?

En général, une erreur d'adresse est due à une tentative d'accès à une adresse extérieure (invalide). Dans ce cas, le programme doit être interrompu, c'est le comportement normal d'un système de swap.

Il est toutefois possible, avec un swappeur paresseux, que la page existe mais ne soit pas en mémoire centrale, d'où les étapes suivantes dans ce cas :

- On peut faire démarrer un processus sans aucune page en mémoire. La première *page fault* aurait lieu à la lecture de la première instruction (l'instruction n'étant pas en mémoire).
- Il faut réaliser une forme spéciale de sauvegarde de contexte, il faut garder une image de l'état du processus qui vient d'effectuer une *page fault* mais de plus il faudra réexécuter l'instruction qui a placé le processus dans cet état, en effet il est possible que l'instruction ne se soit pas terminée par manque de données.

Le système d'exploitation a ici un rôle important, c'est lui qui va réaliser le chargement de la page manquante puis relancer le processus et l'instruction.

Les circuits nécessaires à la méthode de pagination à la demande sont les mêmes que ceux que l'on utilise pour un système de swap paginé, c'est-à-dire une mémoire secondaire et un gestionnaire de pages (table des pages).

Par contre, la partie logicielle est beaucoup plus importante.

Enfin il faut que les instructions soient **interruptionnelles**, ce qui n'est pas toujours le cas sur tous les processeurs et ce qui est fondamental. Par exemple, pour cette instruction nous avons :

`add A,B in C`

1. chercher et décoder l'instruction `add`
2. charger le contenu de l'adresse A
3. charger le contenu de l'adresse B
4. additionner et sauvegarder dans C

Si l'erreur de page a lieu dans le 4<sup>e</sup> accès à la mémoire (C), il faudra de nouveau recommencer les 3 accès mémoire de l'instruction, c'est-à-dire lire l'instruction, etc. Un autre type de problème vient d'instructions comme la suivante que l'on trouve sur PDP-11<sup>1</sup>.

`MOV (R2)++, --(R3)`

Cette instruction déplace l'objet pointé par le registre R2 dans l'adresse pointé par R3, R2 est incrémenté après le transfert et R3 avant.

2. Que se passe-t-il si l'on a une erreur de page en cherchant à accéder à la page pointé par R3 ?

*Source* : <http://www-igm.univ-mlv.fr/~dr/>

---

1. Un PDP-11 est un modèle d'ordinateur de la gamme des PDP (Programmable Data Processor) construit par Digital Equipment Corporation (DEC) entre 1970 et 1993.