

<p>info524 : Systèmes d'exploitation</p> <p>Examen</p>
--

Pierre Hyvernât et Rodolphe Lepigre  
Pierre.Hyvernât@univ-smb.fr  
Rodolphe.Lepigre@univ-smb.fr

---

Durée : 1h30.

Documents et calculatrices interdits.

Un barème provisoire est donné dans la marge.

1 point négatif sera réservé à la présentation de vos réponses...

---

**Partie 1 : général**

- (3) Question 1. Donnez une liste, aussi précise et complète que possible, des appels systèmes provoqués par la commande

```
$ cp -rv mon_repertoire/ ma_sauvegarde/
```

qui permet de recopier tout le contenu d'un répertoire (`mon_repertoire`) dans un autre répertoire (`ma_sauvegarde`), en affichant les noms des fichiers au fur à et à mesure qu'ils sont copiés.

Note : il n'est pas nécessaire de donner le nom des appels systèmes si vous ne les connaissez pas. Vous pouvez simplement dire ce qu'ils font...

- (2) Question 2. La commande

```
$ sort mon_fichier
```

permet d'afficher, sur la sortie standard, les lignes du fichier `mon_fichier` dans l'ordre alphabétique. On peut rediriger la sortie standard dans un autre fichier avec une redirection `>`, mais ceci ne marche pas si on spécifie le même fichier : la commande

```
$ sort mon_fichier > mon_fichier
```

ouvre 2 fois le fichier : une fois en lecture (`sort`) et une fois en écriture (redirection). L'ouverture en écriture supprime les données et cette commande a donc pour effet de vider le fichier.

La commande

```
$ sponge mon_fichier
```

absorbe toute l'entrée standard (jusqu'à ce qu'elle soit fermée) et seulement à ce moment, écrit ce qu'elle a absorbé dans un fichier.

Utilisez les commandes `sort` et `sponge` (et uniquement celles ci) avec des redirection pour trier le fichier `mon_fichier` et sauvegarder le résultat dans le même fichier.

**Partie 2 : processus**

- (3) Question 1. On considère les processus suivants :

processus	temps d'arrivée	priorité	durée
$P_1$	0	1	8
$P_2$	1	2	4
$P_3$	5	2	2
$P_4$	5	1	4
$P_5$	9	1	2

Décrivez l'ordonnancement de ces processus si on utilise un tourniquet FIFO avec priorités et préemption au bout d'un quantum de 2 unités de temps.

*Note* : un processus de priorité 1 est plus prioritaire qu'un processus de priorité 2.

(3) *Question 2.* En utilisant les appels système `fork` et `wait`, écrivez un programme qui fait la chose suivante :

- le processus appelle la fonction `init()`,
- le processus crée un fils qui appelle la fonction `action_fils()`,
- le processus initial appelle la fonction `action_pere()`,
- le processus initial attend que son fils termine son action et appelle la fonction `final()`.

*Note* : vous pouvez ignorer l'argument de `wait` et considérer que l'appel `wait()` attend la fin d'un des fils du processus courant et renvoie son PID.

### Partie 3 : mémoire

(2) *Question 1.* La taille des pages pour un système Linux est de 4Ko ( $2^{12}$  octets). Le système considéré est un système 32 bits avec 1Go ( $2^{30}$  octets) de RAM.

Combien de *page* est-ce que la mémoire virtuelle contient ?

Combien de *cadres* est-ce que la mémoire physique contient ?

(3) *Question 2.* En hexadécimal, une adresse de ce système est constitué de 8 chiffres, dont 3 constituent l'adresse relative dans une page / cadre et les 5 autres le numéro de page / cadre.

Voici un extrait de la table des pages :

Page	In/Out	Cadre
⋮	⋮	⋮
0xabcde	In	0x12345
⋮	⋮	⋮
0x54321	Out	0xedcba
⋮	⋮	⋮

les numéros de pages et de cadres sont donnés en hexadécimal...

Pour les adresses virtuelles suivantes, donnez l'adresse physique correspondante, ou expliquez pourquoi vous ne pouvez pas répondre :

- 0xabcde543
- 0x54321cba
- 0x123abcde
- 0x543edcba

(2) *Question 3.* Un programme fait référence aux pages suivantes :

1, 2, 1, 3, 1, 4, 3, 2, 2, 1, 5, 2, 3, 2, 5, 6, 5, 3, 1, 2.

Combien de défauts de page sont générés dans un système avec 3 cadres si on utilise le remplacement LRU ("least recently used").

Quel est le nombre minimal de défaut de page qu'il est possible d'obtenir pour ces références ?

### Partie 4 : Fichiers

(2) *Question 1.* Sur un système de fichier avec FAT, il est possible de créer un fichier (corrompu) dont la liste des blocs est cyclique.

Ceci est-il possible sur un système de fichiers avec inodes ?