

<p>info524 : Systèmes d'exploitation Examen, session de rattrapage</p>
--

Pierre Hyvernats et Rodolphe Lepigre
Pierre.Hyvernats@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

```
$ ls | wc -l > nb_fichiers
```

qui permet de lister les fichiers du répertoire courant (commande `ls`), de lancer la commande `wc -l` (qui compte le nombre de lignes, et donc de fichiers, du résultat de la commande précédente), et de mettre le résultat dans le fichier `nb_fichiers`.

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

```
$ tee <fichier>
```

permet de dupliquer la sortie standard : elle est affichée

- sur la sortie standard,
- et dans le fichier `<fichier>`.

Si le fichier n'existe pas, il est créé, sinon, il est remplacé.

En utilisant des redirections, donnez une commande unique qui permet d'afficher la liste des fichiers du répertoire courant en même temps :

- sur la sortie standard,
- dans le fichier `L1`,
- dans le fichier `L2`.

Expliquez brièvement comment fonctionne votre commande.

Partie 2 : processus

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

processus	temps d'arrivée	priorité	durée
P_1	0	1	4
P_2	1	1	2
P_3	5	3	4
P_4	5	2	4
P_5	9	1	2

Décrivez l'ordonnement 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 initial affiche "START",
- le processus initial doit créer 20 fils :
 - lors de la création d'un fils, le processus initial doit afficher une ligne ">>>> nb" (où *nb* est le numéro du fils (0, 1, 2, ..., 19),
 - chaque fils doit exécuter la fonction `calcul(nb)` où *nb* est le numéro du fils (0, 1, 2, ..., 19),
 - il y a au plus deux fils en exécution (autrement dit, le processus père doit créer 2 fils et attendre que l'un des deux soit terminé avant de recréer un nouveau fils),
- lorsque tous les fils sont terminés, le processus initial affiche "END".

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 64 bits avec 8Go (2^{33} octets) de RAM. Sur ce système, les adresses sont stockées sur 48 bits.

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

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

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

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

(3) *Question 1.* Expliquez le fonctionnement des *liens physiques* sur un système de fichier avec inodes.

(2) *Question 2.* Expliquez pourquoi les systèmes de fichiers dédiés au CDROM (style ISO9660) sont en général plus simples que les systèmes de fichiers dédiés aux disques durs.