

Système d'Exploitation
Polytech Lille IS 3

15 Février 2023 - Durée : 2 heures

Tous documents papier autorisés. Calculatrices, téléphones et autres dispositifs portables interdits.

1 Processus et ordonnancement

Question 1:

- (a) On considère le code ci-dessous, à partir duquel on produit l'exécutable nommé `calc`. On lance cet exécutable deux fois, chacun depuis un terminal différent. La machine considérée dispose de 4 cœurs. On suppose que la charge générée par les autres processus est négligeable. Quelle sera l'occupation de chaque cœur, en pourcentage (à 5% près) ?

```
int main() {
    int cpt=0;
    while(1) { int even=1; if(!even) cpt++; even=(even+1)%2;}
    printf("Complete\n");
    return 0;
}
```

- (b) Même question pour le code ci-dessous.

```
int main() {
    int cpt=0;
    while(1) { int even=1; if(!even) cpt++; else sleep(1); even=(even+1)%2;}
    return 0;
}
```

Question 2:

Sur un système où le temps de commutation est négligeable, la priorité la plus faible est 1 et le quantum de temps est 10, on considère les processus suivants :

Processus	Priorité	Durée	Date d'arrivée
P1	2	30	0
P2	2	40	0
P3	3	30	20
P4	1	30	0

- (a) Donnez l'occupation en mono-cœur avec la politique par priorité ;
(b) Même question en bi-cœur.

2 Interpréteur de commandes

Question 1:

Pour cette question, vous ne **devez pas** utiliser la commande `cd`. On considère l'exécution suivante :

```
forget@capitole:~/tmp/Codes$ pwd
/home/forget/tmp/Codes
forget@capitole:~/tmp/Codes$ ls
C  Java  OCaml
```

- (a) Donnez une commande affichant le nombre de lignes de tous les fichiers d'extension `.ml` du répertoire `OCaml`.
(b) Donnez une commande permettant de copier le fichier `README.md` depuis le répertoire `Java` vers le répertoire `OCaml`, en utilisant uniquement des chemins relatifs.

(c) Donnez une commande pour effectuer la même copie, mais en utilisant uniquement des chemins absolus.

Question 2:

Donnez une commande permettant d'écrire dans le fichier `list_all_mods.txt` la liste des fichiers du répertoire courant pour lesquels au moins une personne a les droits de lecture, écrire, et exécution.

Question 3:

On considère une base de données très simple stockée dans un fichier `music.txt`, et contenant des titres musicaux. Les champs sont dans l'ordre suivant (pour simplifier, on suppose que chaque champs est constitué d'un seul mot) :

auteur album morceau duree annee

Un exemple est donné ci-dessous :

```
Bjork;Homogenic;Hunter;4:15;1997
Moondog;Moondog;Lament;1:47;1969
M83;Hurry;Wait;5:43;2011
M83;Hurry;Fountains;1:21;2011
```

(a) Donnez une commande affichant tous les morceaux de M83 dans l'album `Hurry`, triés par années croissantes;

(b) Donnez une commande qui transforme chaque ligne, comme illustré sur l'exemple ci-dessous :

```
Bjork;Homogenic;Hunter;4:15;1997
devient
Bjork Hunter 4m15
```

3 Gestion mémoire

Question 1:

On considère le programme ci-dessous :

```
int* pg; char t[5];
int main() {
    char s[]="toto"; float f;
    pg=malloc(10*sizeof(int));
    return 0;
}
```

(a) Donnez une ligne de code C permettant d'afficher : l'adresse de `f`, les valeurs de `pg`, `t`, et `s` (qui se trouvent toutes être des adresses);

(b) Pour chacune de ces 4 adresses, précisez dans quelle zone de la mémoire elle est située.

Question 2:

On considère un système d'exploitation avec des adresses sur 16 bits. La mémoire est gérée par pagination, avec 16 pages. La table des pages est gérée à l'aide de la politique FIFO. La mémoire physique utilisée ne dispose elle que de 4 cadres de pages. On suppose que la table des pages a l'état suivant :

N° page	N° cadre	Date de chargement
10	0	30
5	1	20
7	2	40
3	3	60

(a) Donnez la composition d'une adresse virtuelle;

(b) A la date 80, une instruction demande accès à l'adresse `0x4570`. Dans quel page se situe cette adresse ?

(c) Donnez l'état de la table des pages après cet accès.

4 Gestion disques

Question 1:

On reprend le fichier vu précédemment contenant la base de données de musique. Réalisez, à l'aide de l'API de fichiers sur les flots (`scanf`, `printf`, etc.), le travail suivant :

- (a) Écrivez une fonction qui permet de sélectionner les morceaux d'un artiste donné, pour une décennie donnée (par exemple 1990), et les écrit dans un autre fichier. Vous devez respecter les contraintes suivantes :
- La signature de la fonction est :
`affiche_morceaux(char src[], char dst[], char artiste[], int decennie)`
 - `src` et le nom du fichier contenant la base, `dst` le nom du fichier résultat.
- (b) Dans la fonction `main`, faites appel à cette fonction. Les paramètres utilisés pour l'appel de fonction sont lus au clavier via `scanf`.

Question 2:

On considère le programme suivant :

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#define MAX 1024

int main(int argc, char *argv[]) {
    int f, n;
    char buf[MAX];
    if((f=open("mon_fichier.txt", O_RDONLY)) == -1){ /* mode lecture */
        perror("ouverture fichier source impossible"); exit(2);
    }

    while ((n = read(f, buf, MAX)) > 0){
        write(1, buf, n);
    }
    exit(0);
}
```

- (a) Que désigne le `1`, dans l'appel à `write` ?
- (b) Pourquoi le dernier paramètre de l'appel à `write` est-il `n` et non pas `MAX` ?