

Système d'Exploitation
Polytech'Lille IS2A 3

18 Juin 2021 - Durée : 2 heures

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

1 Processus et ordonnancement

Question 1:

Quel est le risque impliqué par un quantum de temps trop court dans le cas où l'on choisit d'utiliser un ordonnancement de type tourniquet ?

Question 2:

Sur une 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	3	30	10
P2	2	30	0
P3	2	20	20
P4	1	40	10
P5	1	30	30

- (a) Donnez l'occupation en mono-cœur avec la politique FIFO (Premier arrivé, Premier servi)
- (b) Donnez l'occupation en mono-cœur avec la politique Priorité+temps partagé+âge.

2 Langage de commandes

Question 1:

Étant donné le comportement ci-dessous, donnez une (seule) commande permettant de déplacer tous les fichiers C du répertoire `~/Cours/Syst/TP1` vers le répertoire `~/Code/ExemplesC`

```
forget@capitole:~/Cours/Syst/TP1$ ls
clean.sh  config  file.c  file.h  list.c  list.h  log.txt
```

Question 2:

On considère une base de donnée d'état civil très simple, stockée dans un fichier `bd.txt`, dont un extrait est fourni ci-dessous :

```
Allard Flore 09-08-1999 Lyon
Depaul Tina 21-06-1993 Paris
Dupont Paul 01-10-1967 Angers
Durand Martine 23-05-1956 Tours
Marcieux Jean 04-09-2001 Lens
Marneux Adeline 10-06-1989 Lyon
...
```

- (a) Donnez une commande affichant les personnes triées par année de naissance ;
- (b) Donnez une commande affichant les noms et prénoms de toutes les personnes habitant à Lyon ;
- (c) Donnez une commande permettant de créer un fichier, nommé `bd-capitale.txt`, qui est identique à `bd.txt`, sauf que « Paris » y est remplacé par « la capitale ».

3 Gestion mémoire

Question 1:

On suppose un tableau déclaré comme suit : `int tab[SIZE]` ; . Expliquez le comportement probable lorsque l'on essaie d'afficher la valeur de `tab[i]`, dans les différents cas suivants :

- (a) $0 \leq i < SIZE$
- (b) $i = SIZE$
- (c) $i = SIZE + 1000$

Question 2:

On considère le programme suivant :

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
char g='a';
int main() {
    char* pc; int* i; char t[MAX];
    pc=&t[2];
    i=malloc(5*sizeof(int));
    return 0;
}
```

- (a) Écrivez une ligne de code C permettant d'afficher (si possible) : l'adresse de la case 3 de `t`, la valeur de `pc` (qui se trouve être une adresse), l'adresse de `g`, l'adresse de `MAX`;
- (b) Pour chacune de ces adresses, ainsi que pour l'adresse `i`, précisez dans quelle zone de la mémoire elle est située (que ce soit à la compilation ou à l'exécution).

Question 3:

On considère un système d'exploitation 16 bits utilisant une mémoire virtuelle à pagination. On suppose que la taille d'une page est 2^{10} o. La machine utilisée dispose de 8ko de RAM. La table des pages est gérée à l'aide de la politique FIFO, et son contenu est le suivant :

N° page	N° cadre	Date de chargement
7	0	200
9	1	150
5	2	160
3	3	40
4	4	310
12	5	250
11	6	410
12	7	330

- (a) Donnez la décomposition d'une adresse logique (combien de bits pour le numéro de page et pour la position dans la page);
- (b) Un processus souhaite accéder à l'adresse 1300 à la date 500. Donnez l'état de la table des pages après cet accès.

4 Gestion disques

Question 1:

On reprend le fichier vu précédemment contenant un état civil simplifié (Question 2.2). Réalisez, à l'aide de l'API ANSI C (donc à l'aide de `scanf`, `printf`, `fscanf`, ...), le travail suivant :

- (a) Écrivez une fonction qui permet d'afficher les habitants d'une ville donnée. Les contraintes sont les suivantes :
 - La signature de la fonction est :
`affiche_habitants_ville(FILE* f, char* nom_ville)`
 - Le format d'affichage d'un habitant est illustré ci-dessous, pour la première ligne du fichier :
`Flore Allard est nee le 09-08-1999 a Lyon`
- (b) Dans la fonction `main`, faites appel à cette fonction. Le nom du fichier contenant la base de données, et le nom de la ville sont lus au clavier via `scanf`.

Question 2:

Pour cette question vous utiliserez l'API C POSIX (`open`, `read`, `write`, etc). Vous ferez attention à bien vérifier les erreurs système possibles.

Réalisez un programme permettant de concaténer deux fichiers dans un troisième. Par exemple, si `concat` est ce programme, suite à la commande `concat in1.txt in2.txt out.txt`, le fichier `out.txt` contient le contenu de `in1.txt` suivi de celui de `in2.txt`.