

Système d'Exploitation
Polytech'Lille GIS2A 3

5 Avril 2019 - Durée : 2 heures

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

1 Processus et ordonnancement

Question 1:

Combien de processus sont exécutés par la commande suivante ?

```
ls | sort | more > res.txt
```

Question 2:

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

Question 3:

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	40	20
P2	2	30	30
P3	1	30	0
P4	1	20	0
P5	3	20	20

- (a) Donnez l'occupation en mono-cœur avec la politique par files d'attente multi-niveaux (priorité+tourniquet) ;
- (b) Même question dans le cas d'une exécution sur un système bi-cœur.

2 Langage de commandes

Question 1:

Etant donné le comportement ci-dessous, donnez une ligne de commande permettant de copier le répertoire DS2018 dans le répertoire TD. Attention, vous devez utiliser une seule commande.

```
forget@capitole:~/tmp/Cours$ find . -type d
.
./Syst
./Syst/Exams
./Syst/Exams/DS2017
./Syst/Exams/DS2018
./Syst/TP
./Syst/TD
```

Question 2:

Etant donné le comportement suivant :

```
forget@capitole:~/Cours/Syst/Exams/2019-GIS2A3$ ls /sbin/mkfs*
/sbin/mkfs          /sbin/mkfs.ext2    /sbin/mkfs.minix   /sbin/mkfs.vfat
/sbin/mkfs.bfs      /sbin/mkfs.ext3    /sbin/mkfs.msdos   /sbin/mkfs.xfs
/sbin/mkfs.cramfs   /sbin/mkfs.ext4    /sbin/mkfs.ntfs
/sbin/mkfs.exfat    /sbin/mkfs.fat     /sbin/mkfs.reiserfs
```

Complétez la commande précédente afin d'obtenir le résultat suivant :

```
exfat
ext2
ext3
ext4
```

Question 3:

Donnez une commande permettant d'écrire dans le fichier `process.txt` la liste de vos processus triée par durée d'exécution. Pour rappel, un exemple d'utilisation de la commande `ps` :

```

forget@capitole:~/Cours/Syst/Exams/2019-GIS2A3$ ps
PID TTY          TIME CMD
2424 pts/0      00:00:01 bash
3756 pts/0      00:00:28 emacs
5278 pts/0      00:00:04 evince
5768 pts/0      00:00:00 ps

```

Question 4:

Le fichier `villes.txt` contient la liste des plus grandes villes d'Europe. En voici un extrait :

```

1 Moscou Russie 14926513hab
2 Londres Royaume-Uni 14611324hab
3 Paris France 12568755hab
4 Ruhr Allemagne 11857353hab
5 Milan Italie 8247125hab

```

Donnez une commande permettant d'en extraire les plus grandes villes française. Chaque ligne respectera le format suivant, par exemple pour Paris :

```
3:Paris:France:12568755 habitants
```

3 Gestion mémoire

Question 1:

Expliquez brièvement à quoi correspond la fameuse erreur « segmentation fault ».

Question 2:

On considère le programme suivant :

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 10
int gv;

int main() {
    float f, tabf[MAX];
    char *s=(char*) malloc(MAX*sizeof(char));
    return 0;
}

```

- Donnez la liste des variables dont l'adresse est calculée à la compilation ;
- Pour chaque variable du programme, précisez dans quelle zone de la mémoire elle est allouée (que ce soit à la compilation ou à l'exécution).

Question 3:

On considère un petit système d'exploitation 8 bits, utilisant une mémoire virtuelle paginée découpée en 8 pages. La machine utilisée dispose de 4 cadres de pages. La table des pages est gérée à l'aide de la politique LRU, et son contenu est le suivant :

N° page	N° cadre	Date de chargement
2	0	100
3	1	80
5	2	220
6	3	160

- Donnez la décomposition d'une adresse logique ;
- Un programme demande à accéder à l'adresse 50 à la date 250. Donnez l'état de la table des pages après cet accès.

4 Gestion disques

Question 1:

Quel est le gain potentiel lorsque l'on utilise l'API ANSI plutôt que l'API POSIX pour manipuler des fichiers ?

Question 2:

On reprend le fichier vu précédemment contenant la liste des grandes villes européennes. Réalisez un programme à l'aide de l'API ANSI C (donc à l'aide de `scanf`, `printf`, `fscanf`, ...) qui lit le contenu de ce fichier et affiche les plus grande villes françaises au format suivant :

```
1 Paris : 14926513 d'habitants
2 Lyon : 2310850 d'habitants
...
```

Question 3:

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

Ecrivez un programme C qui :

1. Prend deux arguments sur la ligne de commande. Dans la suite, nous appellons `src` le premier argument et `dst` le deuxième;
2. Copie le contenu de `src` dans `dst` en insérant le texte « -nxt- » tous les 10 caractères copiés.
Par exemple, étant donné le contenu suivant du fichier `src` avant exécution du programme :

Ceci est le texte contenu dans le fichier dit `src`

On obtient alors comme contenu du fichier `dst` après exécution du programme :

Ceci est l-nxt-e texte co-nxt-ntenu dans-nxt- le fichie-nxt-r dit `src`