

Systeme d'Exploitation Polytech'Lille GIS 3

19 Mars 2019 - Durée : 2 heures

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

1 Processus et ordonnancement

Question 1:

En comparaison avec les autres algorithmes d'ordonnancement vus en cours, quelle est la principale qualité de l'algorithme du tourniquet ?

Question 2:

Vous venez de lancer depuis votre terminal la commande `emacs`, qui a provoqué la création du processus 2984. L'interface graphique de `emacs` ne répond plus. Donnez deux manières de terminer l'exécution de cet `emacs`.

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 | 3 | 20 | 10 |
| P2 | 2 | 40 | 0 |
| P3 | 1 | 30 | 0 |
| P4 | 2 | 30 | 0 |

- (a) Donnez l'occupation en mono-cœur avec la politique par files d'attente multi-niveaux (priorité+tourniquet) ;
- (b) Même question avec la politique priorité+tourniquet+âge.

2 Langage de commandes

Question 1:

Expliquez pourquoi, après avoir compilé un programme C, vous ajoutez généralement les caractères `./` devant le nom de l'exécutable produit pour demander son exécution à l'interpréteur de commandes.

Question 2:

On considère l'exécution suivante :

```
forget@capitole:~$ ls /lib32
libnss_compat-2.27.so  libnss_nisplus-2.27.so
libnss_compat.so.2    libnss_nisplus.so.2
libnss_dns-2.27.so    libnss_nis.so.2
libnss_dns.so.2       libpcprofile.so
libnss_files-2.27.so  libpthread-2.27.so
libnss_files.so.2     libpthread.so.0
libnss_hesiod-2.27.so libresolv-2.27.so
libnss_hesiod.so.2    libresolv.so.2
```

- (a) Donnez le résultat de la commande suivante : `ls /lib32/libnss*.so | wc -l`
- (b) Et pour la commande suivante :
`ls /lib32/ | grep nss | cut -d'_' -f2 | cut -d'.' -f1 > libs.txt`

Question 3:

Donnez une commande permettant de lister le contenu de votre répertoire personnel, de sorte que les éléments de la liste soient groupés par droits d'accès identiques. Exemple de résultat attendu :

```
drwxr-xr-x 7 forget forget 20480 janv. 16 11:05 Documents
drwxr-xr-x 8 forget forget 4096 avril 10 2017 Applis
-rw-r--r-- 1 forget forget 125 fevr. 8 14:54 notes.txt
-rw-r--r-- 1 forget forget 168414 nov. 3 2016 document.pdf
-rw-r--r-- 1 forget forget 8980 nov. 3 2016 examples.desktop
```

```

-rw-rw-r-- 1 forget forget      0 sept. 21 2017 test.m
-rw-rw-r-- 1 forget forget 10445 mars 12 2018 if_includes.txt
-rw-rw-r-- 1 forget forget   171 sept. 3 14:55 XiVO_Client.log
-rw-rw-r-- 1 forget forget 21626 fevr. 7 2018 rtsched.sty

```

Question 4:

Le contenu du fichier `poeme.txt` est donné ci-dessous. Donnez une commande permettant d’afficher le contenu de ce fichier tout en remplaçant “réseau” par “système” et en remplaçant les virgules par des points virgules :

```

j’aime le cours de reseau, je reve de reseau,
reseau, reseau, reseau, toujours reseau

```

3 Gestion mémoire

Question 1:

Soit un PC disposant de 8Go de mémoire RAM au total. A un instant donné, 7,5Go de cette mémoire sont occupés par des programmes en cours d’exécution. Un des programmes demande alors une allocation supplémentaire de 1Go à l’aide d’un `malloc`. Expliquez très brièvement le résultat de cette demande.

Question 2:

On considère le programme suivant :

```

#include <stdio.h>
#include <stdlib.h>
#define SIZE 10
char g='a';
int main() {
    char s[SIZE]; char* pc;
    pc=&g;
    return 0;
}

```

- Ecrivez une ligne de code C permettant d’afficher (si possible) : l’adresse de `g`, l’adresse de la case 2 de `s`, la valeur de `pc` (qui se trouve être une adresse), l’adresse de `SIZE`;
- Dans quelles zones usuelles de la mémoire sont situées chacune des adresses mentionnées dans la question précédente ?

Question 3:

On considère un petit système embarqué, dont le compilateur vise une mémoire virtuelle constituée 32 pages de 2ko chacune. Le matériel physique dispose en réalité de 4 cadres des pages. La table des pages est gérée à l’aide de la politique FIFO et son contenu est donné ci-dessous.

| N° page | N° cadre | Date de chargement |
|---------|----------|--------------------|
| 0 | 0 | 50 |
| 5 | 1 | 20 |
| 3 | 2 | 10 |
| 2 | 3 | 30 |

- Donnez la taille d’une adresse virtuelle, et détaillez sa composition ;
- A la date 60, une instruction demande à accéder à l’adresse 2300. Donnez l’état de la table des pages après cet accès.

4 Gestion disques

Question 1:

Expliquez ce que fait la commande suivante, et pourquoi elle échoue :

```

forget@capitole:~/Cours$ cp -r ~ /
cp: impossible de creer le repertoire '/forget': Permission non accordee

```

Question 2:

On considère un fichier décrivant les résultats d'une poule de compétition de curling. Chaque ligne précise : le nom puis le score de la première équipe, ensuite le nom puis le score de la deuxième équipe (on suppose qu'un nom d'équipe fait au maximum 20 caractères). Exemple :

```
Allemagne 8 Hollande 7  
Russie 10 Norvege 8  
Coree 4 Suede 5
```

Réalisez un programme à l'aide de l'API ANSI C (donc à l'aide de `scanf`, `printf`, `fscanf`, ...) qui : 1) Lit un nom de fichier au clavier (le fichier respecte le format décrit ci-dessus); 2) Affiche le contenu de ce fichier à l'écran selon le format suivant :

```
Match 1 : 8-7 (Allemagne vs Hollande);  
Match 2 : 10-8 (Russie vs Norvege);  
Match 3 : 4-5 (Coree vs Suede);
```

Question 3:

On souhaite réaliser un programme prenant trois arguments, qui lit le contenu du fichier correspondant au premier argument, copie 4 caractères dans le fichier correspondant au deuxième argument, puis copie les 4 suivants dans le fichier correspondant au troisième argument, et ainsi de suite.

Par exemple, si `mycommand` est ce programme, et si `in.txt` contient le texte "un chasseur sachant", la commande `mycommand in.txt out1.txt out2.txt` écrit "un ceur ant" dans `out1.txt` et "hasssach" dans `out2.txt`.

Réalisez ce programme uniquement à l'aide de l'API POSIX (vous ne devez pas utiliser l'API ANSI C, donc vous ne pouvez pas utiliser `printf/scanf`).