

Examen - Systèmes et Réseaux 1 - Licence 3 Informatique

Durée : 2h. Documents personnels autorisés. Le barème est indicatif.

Exercice 1: Redirection (3 pts)

Donner la suite de commandes qui vous permettra de créer dans le répertoire courant le fichier `information` qui contiendra les informations suivantes (dans cet ordre).

- le texte *début du fichier*
- la date du jour
- le contenu du fichier `listing.txt` se trouvant dans le répertoire père du répertoire courant
- le nombre de mots contenus dans le fichier `listing.txt`
- les droits liés au fichier `listing.txt`
- le nombre de fichiers contenus dans le répertoire `documents` (se trouvant dans le répertoire courant)
- le nom du répertoire courant
- la valeur de variable d'environnement `LANG`

Exercice 2: Trouver les commandes (3 pts)

La copie d'écran donnée en Annexe représente une fenêtre shell après un certain nombre de commandes. Ces commandes ont été cachées. Donner une proposition de ce que pourrait être chacune des ces commandes.

Exercice 3: Programmation système (7 pts)

1. Donner une suite de commandes permettant d'afficher la i ème ligne d'un fichier texte `fic`.
2. Écrire un script shell/awk `part fic numpart nombrepart` qui va afficher la part numéro `numpart` du fichier texte `fic` dans un partage équitable en `nombrepart` parts (à une ligne près) sous la forme :
`part numero numpart : de nom1 à nom2 inclu,`
où `nom1` est le nom se trouvant en début de la première ligne de la part et `nom2` celui se trouvant en début de la dernière.
3. Écrire le script `assigne fic1 fic2`, qui va découper le fichier `fic2` en autant de parts qu'il y a de noms dans le fichier `fic1` et afficher le récapitulatif de la répartition (voir exemple ci-dessous).
4. Écrire le script permettant d'envoyer la répartition à chaque utilisateur par mél (on supposera que la commande `mail log message` permet d'envoyer la chaîne `message` à l'utilisateur ayant pour login `log` et que ces logins se trouvent juste après le nom sur chaque ligne du fichier d'utilisateurs).
5. On dispose d'une fonction C `envoiSMS(char* msg, int tel)` de la bibliothèque `gsm.h`. Expliquer comment interfacier un programme C utilisant cette fonction avec le script `assigne` pour pouvoir envoyer la répartition par SMS aux utilisateurs (numéro de téléphone stocké après le login dans le fichier utilisateurs).

Par exemple, si les fichiers `dossiers.txt` et `utils.txt` contiennent les lignes suivantes :

```
$ cat dossiers.txt
tata 1 12 37
tete 5 5 62
titi 2 17 32
toto 9 2 77
tutu 6 11 32
```

```
$ cat utils.txt
Charles ch516 0618192021
Louis lo998 0622232425
```

la commande `assigne utils.txt dossiers.txt` va afficher

Charles -> part numero 1 : de tata à titi inclus

Louis -> part numero 2 : de toto à tutu inclus

Exercice 4: Sémaphores (3 pts)

Soit une variable X et deux sémaphores $S1$ et $S2$ dont les valeurs initiales sont données par : `Int X = 0; Sem-Init(S1, 2); Sem-Init(S2, 1);`.

On considère trois tâches concurrentes $T1$, $T2$ et $T3$ définies comme suit :

<u>Tâche T1</u>	<u>Tâche T2</u>	<u>Tâche T3</u>
P(S2);		P(S1);
P(S1);	P(S1);	P(S2);
X = X * 2;	X = X * X;	X = X + 3;
V(S1);	V(S1);	V(S2);
V(S2);		V(S1);

Quelles sont les valeurs possibles de X ? Expliquer comment vous obtenez les différentes valeurs. Préciser également les cas de blocage de l'ensemble des processus.

Exercice 5: Diffusion simple de valeur par n processus (4 pts)

On considère 3 processus P_0, P_1, P_2 partageant un tampon ou buffer B . B peut contenir une seule valeur de type quelconque T . Le processus P_0 , appelé diffuseur, produit et place dans le buffer B une valeur V (de type T) qui doit être lue par les autres processus P_1, P_2 , appelés récepteurs. Chaque valeur produite et déposée dans le buffer B doit être lue par les récepteurs P_1, P_2 avant la production et le dépôt de la valeur suivante par le processus diffuseur P_0 . Une valeur déposée dans B ne peut être lue qu'une seule fois par un processus récepteur.

1. Expliquer, en utilisant des sémaphores, les règles de synchronisation des processus P_0, P_1, P_2 pour réaliser la diffusion d'une valeur de P_0 vers les autres processus.
2. Écrire à l'aide de sémaphores un algorithme (pseudo code) pour les processus P_0, P_1 et P_2 .
3. En vous basant sur l'exemple de trace d'exécution de processus étudié en TD, illustrer votre solution pour le processus diffuseur P_0 et les processus récepteurs P_1, P_2 . Vous devez montrer les évolutions des variables et des sémaphores (valeurs et files d'attente) pour les événements ou arrivées de processus suivants :
 1. Exécution du Processus récepteur P_2
 2. Exécution du Processus diffuseur P_0
 3. Exécution du Processus récepteur P_1
 4. Exécution du Processus récepteur P_2
 5. Exécution du Processus diffuseur P_0
 6. Exécution du Processus récepteur P_1
 7. Exécution du Processus récepteur P_2
 8. Exécution du Processus récepteur P_2
 9. Exécution du Processus récepteur P_1
 10. Exécution du Processus diffuseur P_0
 11. Exécution du Processus récepteur P_2
 12. Exécution du Processus diffuseur P_0
 13. Exécution du Processus récepteur P_1

A Annexe exercice 2 : copie d'écran d'une fenêtre shell

```
migniot: ~/examen/documents
migniot:~/examen$ commande 1
/home/migniot/examen
migniot:~/examen$ commande 2
documents fichier.pdf info
migniot:~/examen$ commande 3
./documents/fichier.pdf
./fichier.pdf
migniot:~/examen$ commande 4
migniot:~/examen/documents$ commande 5
total 3120
-rw-rw-r-- 1 migniot migniot 1555217 avril 11 15:57 article.pdf
-rw-rw-r-- 1 migniot migniot 75355 avril 17 16:49 correction.odt
-rw-rw-r-- 1 migniot migniot 1555217 avril 11 15:57 fichier.pdf
-rw-rw-r-- 1 migniot migniot 34 juin 8 15:36 texte
migniot:~/examen/documents$ commande 6
migniot:~/examen/documents$ commande 5
total 3120
-rw-rw-r-- 1 migniot migniot 1555217 avril 11 15:57 article.pdf
-rw-rw-r-- 1 migniot migniot 75355 avril 17 16:49 correction.odt
-rw-rw-r-- 1 migniot migniot 1555217 avril 11 15:57 fichier.pdf
-rw-rw-r-- 1 migniot migniot 34 juin 8 15:36 texte
migniot:~/examen/documents$ commande 7
Voici le contenu du fichier texte
migniot:~/examen/documents$ commande 8
1 6 34 texte
migniot:~/examen/documents$ commande 9
texte: ASCII text
migniot:~/examen/documents$ commande 10
migniot:~/examen/documents$ commande 5
total 3120
-rw-rw-r-- 1 migniot migniot 1555217 avril 11 15:57 article.pdf
-rw-rw-r-- 1 migniot migniot 75355 avril 17 16:49 correction.odt
-rw-rw-r-- 1 migniot migniot 1555217 avril 11 15:57 fichier.pdf
lrwxrwxrwx 1 migniot migniot 7 juin 8 16:02 info -> ../info
-rw-rw-r-- 1 migniot migniot 34 juin 8 15:36 texte
migniot:~/examen/documents$ commande 11
PID TTY TIME CMD
6377 pts/3 00:00:00 ps
11030 pts/3 00:00:00 bash
11042 pts/3 00:00:38 bash
migniot:~/examen/documents$ commande 12
migniot:~/examen/documents$ commande 11
PID TTY TIME CMD
11030 pts/3 00:00:00 bash
14302 pts/3 00:00:00 ps
migniot:~/examen/documents$
```