

Examen Systèmes et Réseaux,

Durée 2h, Documents issus du cours autorisés

Exercice 1 (3 pts)

Soit le fichier « mots.txt » ci-dessous, qui se trouve dans le répertoire nommé « ressources » qui est un sous répertoire du répertoire courant « repAwk ».

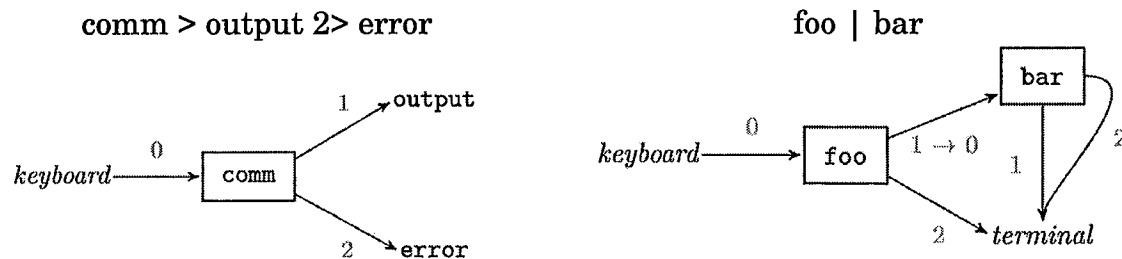
mots.txt

```
some words any data
some further words
somer morer things
more other thing
even more data
```

1. Que donne la commande :
`awk '{ print }' ressources/mots.txt`
2. Ecrire un script awk qui compte les lignes contenant le mot « some », affiche toutes ces lignes ainsi que leur nombre.
3. Montrer comment exécuter ce script à partir du répertoire parent de « repAwk ».

Exercice 2 (2 pts)

Les diagrammes de redirection sont un bon moyen pour visualiser les redirections des entres/sorties entres des commandes Unix. Deux exemples sont donnés ci-dessous :



Dans la commande `foo | bar` la sortie standard `stdout` de la commande `foo` est redirigée vers l'entrée standard `stdin` de la commande `bar`.

Dans la commande `foo |& bar` les deux sorties `stdout` et `stderr` de la commande `foo` sont redirigées vers l'entrée standard de la commande `bar`.

Expliquer brièvement la ligne suivante et dessiner son diagramme de redirection :

```
ls |& grep test | grep blub | awk '{ print $2 }' & > outfile
```

Exercice 3 (5 pts)

Écrire un script `seq.sh` qui affiche une séquence de nombres entiers et fonctionne de la façon suivante :

- `seq.sh -h`
Affiche un message d'aide, et se termine en retournant le code 0
- `seq.sh n1`
Où on suppose que $n1$ est un nombre entier positif
Affichera la séquence des nombre entiers entre 1 et $n1$, et se termine en retournant le code 1
- `seq.sh n1 n2`
Où on suppose que $n1$ et $n2$ sont deux nombres entiers positifs.
On ne suppose pas que $n1 > n2$, ou $n2 > n1$, le script doit identifier le plus petit des deux nombres et affiche la séquence dans l'ordre croissant. Le script se termine en retournant le code 1
- `seq.sh n1 n2 p`
Où on suppose que $n1$, $n2$ et p sont des nombres entiers positifs.
On ne suppose pas que $n1 > n2$, ou $n2 > n1$, le script doit identifier le plus petit des deux nombres et affiche la séquence dans l'ordre croissant avec un pas égal à p . Le script se termine en retournant le code 1
Exemple : L'appel `seq.sh 18 6 4` affichera 6 10 14 18
- Pour tout autre combinaison de paramètres, le script affiche un message d'erreur, et se termine en retournant le code 0

Exercice 4 (6pts) : Communication entre 3 processus

1. On considère 3 processus P_0 , P_1 , P_2 qui partagent un tampon ou buffer B . B peut contenir une seule valeur de type quelconque T . Le processus P_0 , produit et place dans le buffer B une valeur V (de type T). Chaque processus P_1 , P_2 lit (une et une seule fois) chaque valeur de B . P_0 peut déposer une nouvelle valeur dans B seulement si la valeur en cours a déjà été lue par P_1 et P_2 .
 - a) Expliquer, en utilisant des sémaphores, comment on peut synchroniser les processus P_0 , P_1 et P_2 et réaliser le traitement décrit ci-dessus.
 - b) Ecrire le pseudo code des processus P_0 , P_1 et P_2 .
2. Expliquer comment faire le même traitement avec des tubes nommés et des signaux au lieu d'un buffer partagé (P_0 devra toujours s'assurer que P_1 et P_2 ont lu la valeur V avant d'envoyer une nouvelle valeur). Donner le pseudo-code de P_0 .
3. Si maintenant les trois processus ne sont pas sur la même machine, expliquer comment faire le traitement à l'aide des sockets (choix du mode de socket adapté, primitive à appeler au sein de chaque process, problèmes spécifiques, etc).
4. Expliquer quelle solution entre buffer partagé et tubes/signaux (question 1 ou 2) vous semble la plus adaptée si le nombre de processus lecteurs (2 ici) est amené à être plus grand (plusieurs centaines par exemple).

Exercice 5 (4pts)

Répondez aux questions suivantes par VRAI ou FAUX. Vous devez écrire votre réponse dans le tableau ci-dessous et rendre l'énoncé avec votre copie. Toute réponse fautive vous pénalise car elle annule les points d'une réponse juste.

	Question	Votre réponse (Vrai ou Faux)
1	Un processus est un programme qui est stocké dans un fichier.	
2	L'instruction "kill -usr1 9090" permet de terminer l'exécution ou tuer le processus numéro 9090.	
3	Une section critique représente un segment de code où le programme peut accéder à une variable ou une ressource partagée.	
4	Un thread est appelé processus léger parce qu'il peut contenir un nombre limité d'instructions.	
5	Dans un modèle Producteurs-Consommateurs, les processus Producteurs ou Consommateurs peuvent effectuer des accès concurrents sur le buffer de stockage de messages en attente.	
6	Les pthreads issus d'un même processus partagent les variables globales du processus.	
7	Un sémaphore S est une variable utilisée par un processus P1 pour envoyer des données à un autre processus P2.	
8	Dans un modèle Lecteurs – Rédacteurs, plusieurs lecteurs peuvent lire en même temps dans un fichier F.	
9	Un signal temps-réel est appelé ainsi car il est reçu plus vite qu'un signal normal	
10	La commande Shell "trap" permet à un processus d'envoyer un signal à un autre processus.	