Projet (mini)Shell

Rapport

Etudiante : CHELLAF EL HAMMOUD Chaimae

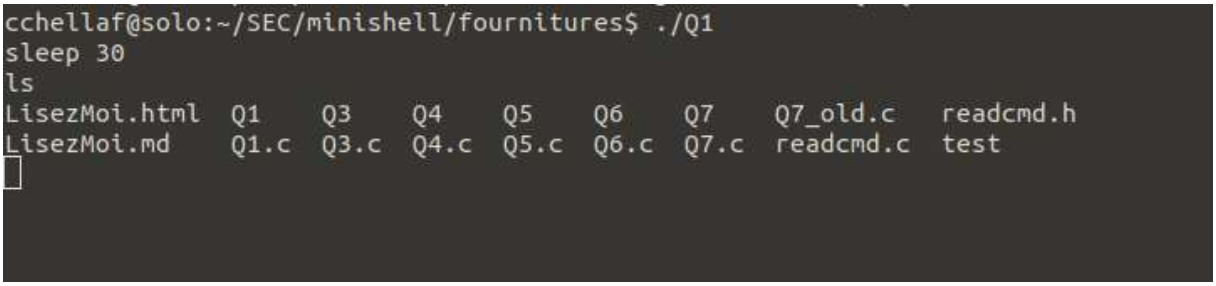
Groupe : F

Année scolaire : 2020/2021

Synchronisation entre le shell et ses fils :

Question 2 :

D'après la figure ci-dessous, l'affichage de l'invite se mêle à l'exécution du processus fils, puisque la commande **ls** est bien exécutée directement après la commande **sleep 30** avant que les 30 secondes ne se terminent.



Les choix opérés pour les questions 6 et 7 :

Question 6 :

Pour réaliser la commande jobs, j'ai créé un tableau qui contient les processus, chaque processus étant défini par son id, pid, état, la ligne de commandes lancée ainsi qu'un booléen « fini » indiquant si le processus est fini ou pas.

Pour les fonctions liées aux différentes opérations sur le tableau et les processus, je les ai placées dans deux fichiers ***tableauProc.h*** et ***tableauProc.c***.

Question 7 :

Pour réaliser la question 7, un traitant du signal SIGINT a été créé avant la boucle while pour permettre au père seul de traiter l'arrêt, tout en masquant ce signal pour tous les fils. Dans le traitant, on envoie un autre signal SIGKILL qui arrête le processus, mais en lui donnant juste le pid du processus en avant plan (pid_fg une variable globale dans le programme), et donc on n'arrête pas tous les processus mais juste le processus actif en avant plan.

Le même traitement a été fait pour le signal SIGTSTP dans la question 6.

Architecture de l'application :

Un fichier Qi.c traite la question i du projet. Pour les questions à partir de la question 6, on importe un autre fichier TableauProc.h contenant toutes les fonctions qui servent pour le traitement des tableaux et processus.

L'application est bien répartie en sous programmes facilitant la compréhension du code. Parmi les sous programmes implantés il y a :

- **commande_fg :** fonction qui exécute la commande fg
- **commande_bg :** fonction qui exécute la commande bg
- **commande_stop :** fonction qui exécute la commande stop

- **rediriger_entree :** fonction qui permet la redirection des fichiers vers l'entrée standard
- **rediriger_sortie :** fonction qui permet la redirection des fichiers vers la sortie standard
- **tubes :** fonction qui permet la composition des commandes en les reliant pas un tube

Il y a 3 traitants :

- le traitant du SIGCHLD nommé **suivi_fils**
- le traitant du SIGINT nommé **handler_ctrlC**
- le traitant du SIGTSTP nommé **handler_ctrlZ**

Quelques tests sur l'application :

✓ Test de la commande jobs et fg :

```
>>> commande : sleep 10
^Z
le fils de pid 21001 est suspendu
>>> commande : jobs
id [3]  |  pid [21001]  |  etat [SUSPENDU]  |  commande [sleep 10 ]
>>> commande : fg 3

le fils de pid 21001 est repris

le fils de pid 21001 est terminé
>>> commande :
```

✓ Test sur la suspension du processus en avant plan :

```
>>> commande : sleep 20 &
>>> commande : sleep 10
^Z
le fils de pid 21376 est suspendu
>>> commande : jobs
id [4]  |  pid [21367]  |  etat [ACTIF]  |  commande [sleep 20 &]
id [5]  |  pid [21376]  |  etat [SUSPENDU]  |  commande [sleep 10 ]
>>> commande :
```

✓ Test sur les redirections :

```
>>> commande : cat < hello > salut

le fils de pid 22244 est terminé
>>> commande : ls
hello          Q3    Q5.c        Q7.c      Q9_10.c        tableauProc.h
LisezMoi.html  Q3.c  Q6          Q7_old.c  readcmd.c      test
LisezMoi.md    Q4    Q6.c        Q8        readcmd.h      test.c
Q1             Q4.c  Q6_modif.c  Q8.c      salut          tester
Q1.c           Q5    Q7          Q9        tableauProc.c

le fils de pid 22251 est terminé
>>> commande :
```

- ✓ Test sur les tubes :

```
>>> commande : ls | wc -l
29
>>> commande :
```