

Rapport Projet Mini-Shell

Joël Roman Ky 1SN-M

18/05/2019

Question 1 :

Implémentation de la boucle principale avec la procédure **lire_commande()** qui affiche le nom du répertoire principale et récupère le répertoire courant

Question 2-3 :

Utilisation de la commande **waitpid** pour mettre en attente le processus père jusqu'à la fin de l'exécution du fils.

Question 4 :

Ajout des commandes internes **cd** à l'aide d'une procédure void en supposant que le paramètre `cmd->seq[0][0]` est égale à **cd** en distinguant deux cas :

-si `cmd->seq[0][1] = NULL` : le répertoire courant n'est pas **HOME**. On récupère le chemin de Home en utilisant `getenv("Home")` et on se déplace en utilisant `chdir`.

-si `cmd->seq[0][1] != NULL` : on utilise `chdir` pour remplacer le répertoire courant par celui donné en paramètre.

La commande interne **exit** se fait avec la procédure prédéfinie.

Question 5 :

-Si `cmd -> backgrounded == NULL` : Dans ce cas on attend le fils jusqu'à qu'il finit l'exécution de la commande en utilisant `wait(NULL)`. Nous avons essayé d'implémenter cette question avec la primitive **pause()** mais cela empêchait de passer à la ligne suivante.

Question 6 :

Implémentation des commandes externes **bg**, **fg**, **stop**, **jobs** traitées dans la boucle principal avec un **if...else if ..else**.

Les commandes **bg**, **fg** et **stop** consistent à retrouver le pid du processus et arrêter le processus en cours et mettre à jour la liste des processus après.

La commande **jobs** consiste à afficher la liste des processus ce qui est déjà fait par la procédure dans le module **Processus.h**

La gestion de la saisie **Ctrl+Z** a été réalisée en supposant que les signaux vont être redéfinies à l'aide d'un nouveau traitant qui va associer au signal **SIGTSTP**, le signal **SIGSTOP**.

Question 7 :

Comme pour la question 6, le signal **SIGINT** va être associé à un nouveau traitant qui associé au signal **SIGKILL**.

Une tentative d'implémentation avec **sigaction** a été tenté mais sans succès compte tenu de la difficulté de manipuler la **struct sigaction** ce qui ne permettait pas aux fils d'ignorer les signaux **SIGINT** et **SIGKILL** avec la primitive **sigprocmask**.

Ceci étant, nous avons juste implémenter le traitant qui associe de nouveaux signaux semblables aux signaux **SIGINT** et **SIGTSTP**

Question 8 :

La redirection a été implémentée avec deux fonctions qui redirigent l'entrée standard et la sortie standard **redir_entree** et **redir_sortie**.

Ces fonctions vérifient si la ligne de commande comporte un fichier de redirection (sortie ou entrée) ce qui sera défini par le fait que **cmd->in!** ou **cmd ->out** pointe sur le pointeur null ou pas.

Les redirections ont été intégrés dans les questions qui suivent.

Question 9 :

L'implémentation du pipe simple a été réalisé en vérifiant d'abord le nombre de paramètres entrés par la fonction qui doit être de deux ici.

Par la suite on applique la primitive **pipe** qui va créer le pipe avec ses deux descripteurs et on réalise la primitive **fork** qui va créer un 2^e fils.

Après avoir fermé les descripteurs superflus le père devra exécuter la première partie de la commande qui peut être éventuellement écrite dans un fichier et faire passer le résultat dans le pipe afin que le fils applique la seconde et fait passer le résultat vers la sortie standard qui pourra éventuellement être redirigé dans un fichier.

Le père attend la fin du fils avant de se terminer.

Question 10 :

L'implémentation de la structure en pipeline a été limitée à juste 2 pipes, l'implémentation de plusieurs pipes devenant un petit peu trop lourd à traiter .

Comme à la question précédente il faudra créer un processus père qui aura un fils qui lui même aura un fils.

Le père exécutera la 1^{ère} partie de la commande l'enverra au fils par le pipe, qui va lui appliquer la 2^e partie de la commande et la transférer à son fils qui va lui appliquer la 3^e partie de la commande et amener le résultat sur la sortie standard, résultat qui pourra être éventuellement redirigé vers un fichier.