

Rapport de projet Mini shell

Mamadou Lamine BAH – Farouma TALL

13 Février 2020

Introduction :

Ce projet consiste à implémenter quelques fonctionnalités similaires à celle d'un shell bash. L'objectif de l'exercice est de mettre à profit bon nombres de notions vues en cours à savoir le traitement des signaux, la gestion des processus, des redirections et des *pipes*, via des appels système. Le programme implémenté contient également d'autres fonctionnalités n'impliquant pas ces notions.

Modifications du programme existant :

- ajout du champ *nbCmd* qui compte le nombres de commandes situées dans *seq* pour simplifier le code par la suite.
- ajout du champ *bg* pour la gestion des tâches en arrière-plan.
- Organisation modulaire du code pour une meilleure lisibilité en créant notamment deux fichiers :
 - *executeCmd.c* qui contient l'implémentation de toutes les commandes qu'on a pu faire du shell
 - *executeCmd.h* qui contient la déclaration d'une structure *jobs* et les prototypes de toutes les fonctions correspondant ainsi qu'une documentation de chacune d'elles
 - Enfin le fichier *shell.c* ne contient que la fonction *main*

Réalisations effectuées :

- commande simple : fonction *executer_commande*. Utilisation des entrées stockées dans le champ *seq* de la structure *cmdline*. Fonction modifiée à chaque nouvelle fonctionnalité.
- implémentation des redirections en entrée/sortie : simple remplacement des E/S standards par les descripteurs de fichier cibles.
- implémentation des pipelines : Création d'un processus fils par commande

détectée. Utilisation d'un tableau de *pipes* pour la communication entre les fils. Ceci permet l'exécution en parallèle des différentes commandes.

- traitement des redirections E/S sur les pipelines : la redirection en entrée s'effectue sur la première commande tandis que la redirection en sortie est depuis le résultat obtenu après la dernière commande.
- Gestion des erreurs : Elle a été gérée principalement dans la fonction *executer_commande_dans_un_fils* notamment si une commande n'existe pas ou l'on essaye d'écrire ou de lire dans un fichier où l'on a pas des droits
- exécution de commandes en arrière plan : Utilisation du « else » de la fonction *executer_commande* et avec l'option WNOHANG.
- Changement de l'état du processus en premier plan et gestion du CTRL_C et du CTRL_Z: fonction *traite_signal* : gestion des signaux avec chacun un traitant de signal pour SIGINT et pour SIGTSTP. Quant à ce dernier, la fonction *ajoutJobs* est appelée pour quand une frappe de ctrlz est détectée, le processus est stoppé et ajouté dans la tableJobs.
- Gestion des zombies : modification de *traite_signal* pour prendre en compte le signal SIGCHLD. Gestion des zombies grâce à l'option WUNTRACED.
- Gestion des jobs:Création d'une structure jobs avec un numéro,un nom,un statut et un numéro de pid , un tableau de jobs,une fonction d'ajout de commande jobs et leur affichage.
- Gestion de la commande fg et stop:Utilisation des fonctions *cmdFg* et *cmdStop*

Description des tests :

- commande simple : commandes *ls*, *cat*, *ps*.
- implémentation des redirections en entrée/sortie : *ls -l > file*, *file > wc -c*.
- pipelines : *ls -l | wc -l | wc -c ± entrée/sortie*.
Test de la commande *find* sur une grosse quantité de données pour vérifier l'exécution des différentes commandes en parallèle.
- commandes en arrière plan/traitement des signaux : *xeyes &*, *cat*.
- Commandes jobs: *xeyes &*
xeyes &
jobs

- commande fg et stop : `xeyes & fg /stop`

Conclusion :

Presque toutes les fonctionnalités ont été réalisées. Les principales difficultés ont été rencontrées sur la gestion de *pipes* multiples, sur le traitement des signaux, nous avons également eu un souci avec la commande fg: le shell exécute bien la commande sauf que lorsqu'il ramène le processus en premier plan il ne redonne pas la main au shell. Sinon globalement, les appels systèmes ont été compris et le shell est bien fonctionnel. Il ne restait plus que l'implémentation de la commande bg dont on a compris le principe mais on ignore comment le tester sur le shell et enfin la dernière question sur l'utilisation des tildes et étoiles n'a pas été faite par manque de temps.