

Rapport Projet Minishell

Benjamin SCHLÖGEL 1SN-B

I / Questions du sujet	1
A / Question traités	1
B / Questions Partiellement traités	1
C / Questions non traitées	1
II / Architecture de l'application	1
III / Tests	2

I / Questions du sujet

A / Question traités

- Question 1
- Question 2
- Question 3
- Question 4
- Question 9

B / Questions Partiellement traités

- Question 5, cette question est complètement traitée, mais des erreurs de "segmentation fault" interviennent depuis l'ajout du module processus.c
- Question 6, "lj" fonctionne et parfois "sj" et "bg". Mais "fg" n'est pas au point.
- Question 7, j'ai mis en place le traitant du père, et les masques des fils qui doivent ignorer le signal. Mais "segmentation fault" arrête le père.
- Question 10 et 11, fonctionne correctement (exécute la commande entière) mais il y a un message d'erreur du waitpid du handler_SIGCHLD

C / Questions non traitées

- Question 8

II / Architecture de l'application

Mon application se compose de plusieurs fichiers :

- minishell.c qui contient les principales fonctions du minishell ainsi que le main. Pour l'exécuter, il faut rentrer les commandes suivantes dans le terminal :

1) gcc -Wall minishell.c readcmd.c processus.c -o minishell

2) ./minishell

J'ai scindé le main en beaucoup de fonctions, ce qui est certes pénalisant, car il faut chercher une à une les fonctions qui s'activent, mais permet d'avoir une bonne clarté dans le code.

- readcmd. et readcmd.h constitue le module qui s'occupe de lire une commande rentrée dans l'invite de commande du minishell, fourni par les enseignants.
- processus.c et processus.h constitue le module qui gère la liste de processus généré par le minishell. J'ai décidé de le faire en dehors du minishell.c pour garder de la cohérence et pour distinguer mieux les différentes parties de mon programme

III / Tests

- Question 1 : a été testé avec les commandes : ls, pwd
- Question 3 : rentrer plusieurs commandes à la suite
- Question 4 : exit fait quitter le processus du minishell, puis j'ai testé les commandes : cd .. , cd non_existant, cd fichier_existant, cd
- Question 5 : j'ai testé avec la commande : sleep 10 &
- Question 6 : j'ai testé lj, puis sj en mettant en sleep en arrière-plan, puis bg sur le processus suspendu
- Question 7 : j'ai testé le Ctrl+C avec un sleep en avant-plan et sans processus en avant-plan
- Question 8 : non testé
- Question 9 : j'ai testé avec les commandes : ls > fichier1, cat < fichier1 > fichier2
- Question 10 et 11 : j'ai testé avec les commandes : ls | wc -l, ls | grep mini | wc -l