

Rapport SEC

- Question 1 : Traitée intégralement
- Question 2 : Traitée intégralement
- Question 3 : Traitée intégralement
- Question 4 : Traitée intégralement
- Question 5 : Traitée intégralement
- Question 6 : Traitée intégralement

- Question 7 : Traitée partiellement

J'ai pu rendre désactivée l'action du Ctrl + Z "classique" et permettre d'arrêter un processus en avant plan dans le minishell. J'ai aussi pu rajouter la commande en avant-plan à la liste des jobs en mode suspendu. Néanmoins, je n'arrive pas à relancer le processus avec bg ou fg. Et, il semble que ma commande Ctrl + Z bloque aussi les processus en arrière plan.

- Question 8 : Traitée partiellement

J'ai pu rendre désactivée l'action du Ctrl + C "classique" et permettre d'arrêter un processus en avant plan dans le minishell. Néanmoins, je n'arrive pas à relancer le processus avec bg ou fg. Et, il semble que ma commande Ctrl + C bloque aussi les processus en arrière plan.

- Question 9 : Traitée partiellement

J'arrive à ouvrir / créer un fichier quand il y a des arguments en in ou out dans ma commande. Néanmoins la commande cat ne fonctionne pas comme il faudrait. J'arrive seulement à écrire le résultat de mes commandes (par exemple "ls" ou "pwd") dans un fichier externe et ne pas afficher les résultats sur le shell.

- Question 10 : Non traitée
- Question 11 : Non traitée

- Architecture du code :

J'ai essayé d'être fidèle aux remarques/conseils de Mr. Hamrouni. J'ai donc choisi d'épurer mon programme principal (main) en réalisant des sous-programmes.

J'ai essayé de structurer avec le maximum de commentaires possibles et de regrouper les fonctions utiles à telle ou telle partie entre elles (affichage, commande lj/sj/bg/fg, handler,...) pour aider à la clarté.

J'ai choisi d'utiliser un tableau de processus avec une capacité fixe de 200. Je me suis dit que c'était le plus pratique à utiliser surtout qu'on ne réalise jamais plus de 200 processus en arrière-plan en même temps.

Par rapport aux tests que j'ai effectués, je me suis beaucoup appuyé sur des affichages de printf à telle ou telle étape de mon code pour déboguer et comprendre ce qui n'allait pas. Je me suis aussi beaucoup servi de la commande "sleep" pour vérifier le comportement des actions lj, sj, fg, bg.

J'ai choisi de garder quelques variables en variables globales comme par exemple "commande_avant_plan" ce qui me permet de conserver le pid des commandes en avant plan et de le modifier dans le handler.

J'ai réussi à faire juste un seul waitpid comme recommandé par Mr. Hamrouni.

- Conclusion sur le projet :

Je suis plutôt satisfait de mon travail. Je n'ai certes pas fait toutes les questions, mais le temps pour réaliser était aussi limité par la date du rendu et tous les autres projets / partiels à réaliser / réviser. Je trouve que j'ai su rendre mon code structuré, épuré, lisible et compréhensible. J'ai pris le temps d'avoir un joli affichage relativement à l'utilisateur du minishell et je trouve cela aussi important. J'ai tout de même trouvé ce projet très difficile et il m'a fallu beaucoup travailler pour arriver à ce rendu. Mais le résultat me convient et j'espère qu'il vous conviendra aussi !