



Projet Minishell

Rapport Final

Hathoute Hamza

Groupe GH

Département Sciences du Numérique - Première année
2020-2021

Contenu

1	Architecture de l'application	3
2	Methodologie de tests	3
3	Question 2	3
4	Question 6	3
5	Question 7	3

1 Architecture de l'application

Le code source de l'application comporte deux fichiers sources: (**Q9.c** et **readcmd.c/h**). J'ai essayé de séparer la partie de gestion de processus dans un module séparé mais pour une raison inconnue j'obtenais des *segmentation fault*.

2 Methodologie de tests

Pour tester mon minishell; pour chaque modification au code, j'exécute des commandes que j'ai sauvegardé dans un fichier texte ainsi que le temps d'exécution et ce que la commande doit afficher.

J'ai aussi ajouté du code debug: en définissant **DEBUG**, le minishell affiche des informations utiles comme l'exécution d'un handler...

3 Question 2

Quand on exécute une commande qui n'est pas interne comme **echo**, **ls**, etc, l'affichage de l'invite précède le résultat de ces commandes.

```
sh-3.2$ ls
sh-3.2$ Calcul_Scientifique  LICENSE  Optimisation-EDT  PIM
Probabilité  README.md  SdC  Telecommunications  TOB

sh-3.2$ echo test
sh-3.2$ test
```

4 Question 6

La principale difficulté pour cette question était de gérer les processus fils et de mettre à jour leurs états. Pour répondre à cette problématique, j'ai procédé comme suit:

- Créer une structe qui englobe les informations sur le processus.
- Créer une liste qui regroupe les informations sur les processus fils.
- Implanter plusieurs fonctions qui gèrent ces processusses.

5 Question 7

Le travail étant fait en question 6, cette question necessite l'ajout d'un handler pour **SIGINT** et d'un masque qui bloque ce signal pour les fils du shell.