

Rapport

Projet (mini)Shell

ANAS TITAH



Introduction

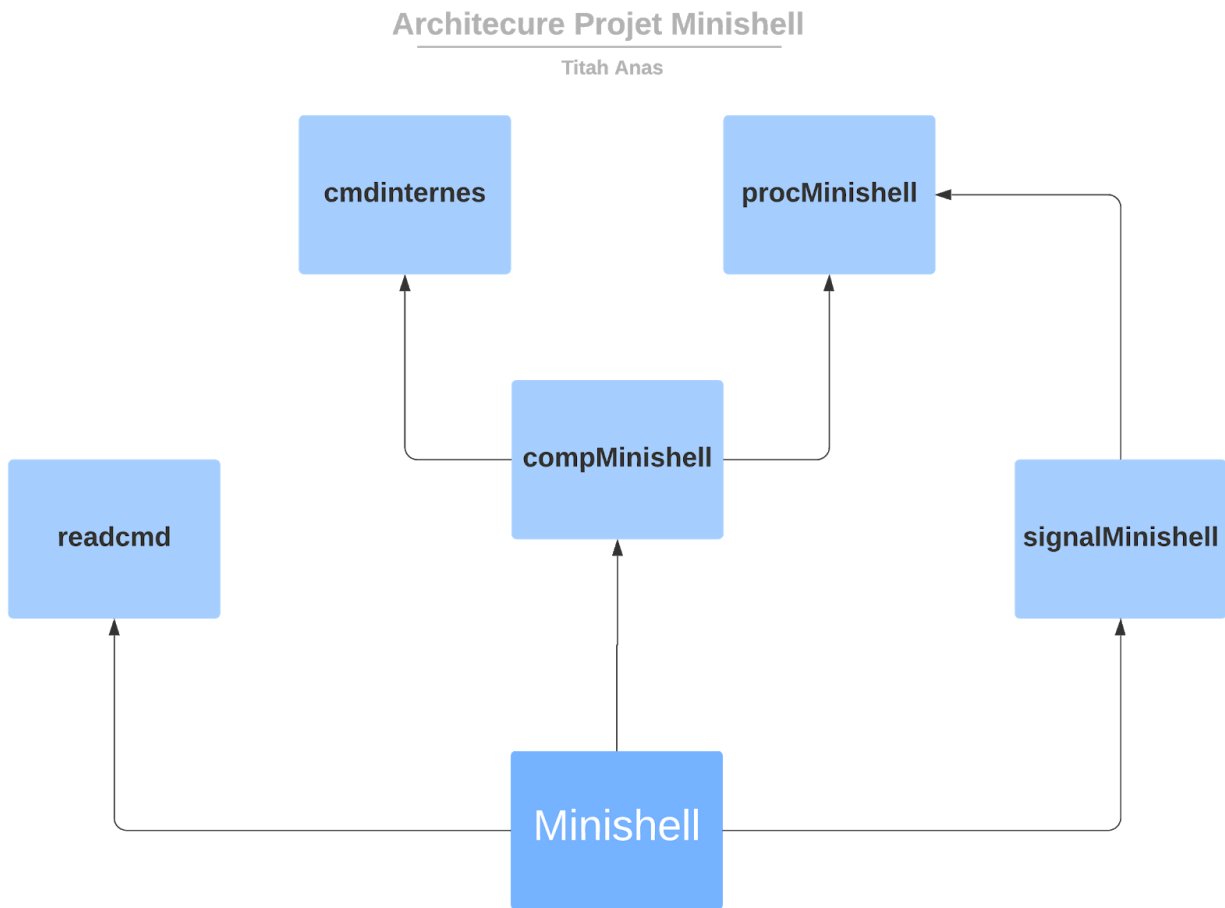
Ce projet a pour but la réalisation d'un minishell en mettant en pratique les notions de base vues en TD et TP, comme la gestion des processus, des signaux et même des entrées/sorties. Cet interpréteur de commandes développé à la fin offre plusieurs fonctionnalités de bases des shells comme celui de bash, powershell etc.

Architecture de l'application

Le minishell a été développé en plusieurs étapes et en construisant plusieurs fichiers afin de regrouper les procédures et fonctions de la même nature pour des raisons de simplification et de lisibilité, j'ai décidé de changer la structure de l'application pour qu'elle s'adapte du mieux possible avec les questions posées au cours du projet et pour qu'elle soit la plus proche possible de celle que j'avais à l'esprit.

Le programme se compose principalement du fichier source **minishell.c** où on trouve la structure de la lecture d'une commande et de son exécution, du fichier source **compMinishell.c** (et de son fichier header **compMinishell.h**) où on implémente les procédures principales utilisées par le minishell comme l'affichage du prompt de notre minishell, exécution des commandes simples, des commandes avec pipelines, et des commandes internes, le programme se compose aussi du fichier source **cmdinternes.c** (Header **cmdinternes.h**) où on définit les commandes internes (exit, cd, list, stop, bg et fg) , du fichier **signalMinishell.c** (Header **signalMinishell.h**) qui contient les traitants des signaux gérés par le minishell, du fichier **procMinishell.c** (header **procMinishell.h**) qui gère la liste des processus lancés depuis le shell et non encore terminés, et enfin du fichier **readcmd.c** (header **readcmd.h**) fourni avec le projet qui lit la commande entrée par l'utilisateur et l'interprète.

Voici une figure qui illustre l'architecture utilisée.



Réponses aux différentes questions / Conception

Question 1 - 2 - 3 : Le code est réalisé dans le fichier source **minishell.c**, ainsi que la commande d'exécution d'une commande simple (**execution_cmd_simple**) dans le fichier **compMinishell.c**.

Question 4 : On trouve les deux commandes **cd** et **exit** dans le fichier **cmdinternes.c**, et pour vérifier quelle commande à exécuter, on utilise la commande **execution_cmd_interne** dans le fichier source **compMinishell.c**.

Question 5 : Cette question est implantée dans la procédure **execution_cmd_simple** où on vérifie dans le père si la commande est une commande en tâche de fond et on passe alors à l'ajouter dans la liste des processus et lire une nouvelle commande, ou bien d'attendre sa fin si elle s'agit d'une commande en avant-plan.

Question 6 : Toutes les fonctions demandées se trouvent dans le fichier source **cmdinternes.c**, et on vérifie ainsi quelle commande à exécuter avec la commande **execution_cmd_interne**. Pour l'explication de la gestion de la suspension d'un processus, voir la question suivante.

Question 7 : La gestion de la frappe du ctrl-C et du suspension d'un processus (Ex : ctrl-Z) a été implantée dans le fichier **signalMinishell.c** qui contient un traitant gérant les signaux SIGINT et SIGTSTP (ainsi que le signal SIGCHLD), la conception a été réalisée de la manière suivante, on traite les deux signaux à chaque entrée dans la boucle en les redirigeant vers le processus en cours d'exécution en avant-plan, puis en les masquant pour les processus en tâche de fond et ne rien faire pour le processus du minishell.

Question 8 : Les redirections ont été bien implantées dans le programme, pour associer l'entrée standard à un fichier, on commence par ouvrir ce fichier en mode lecture, et on utilise la fonction **dup2** pour dupliquer le descripteur de ce fichier dans l'entrée, du même, pour l'association de la sortie standard à un fichier où on ouvre le fichier en mode écriture et on utilise de même la fonction **dup2**. On trouve cette implantation dans la procédure **execution_cmd_simple** et précisément dans le fils, et dans la procédure **execution_cmd_pipes** où on n'applique les redirections que pour la première commande en redirigeant l'entrée (si demandée), et pour la dernière commande en redirigeant la sortie (si demandée).

Question 9 - 10 : La réalisation de ces deux questions se trouve dans la procédure **execution_cmd_pipes**, le principe est d'utiliser une boucle while et de vérifier que la commande d'indice num_cmd (initialisé par 0) existe et est non nulle, ensuite, on crée un pipe et on lance un fils (avec fork), on redirige l'entrée vers celle mémorisée dans la variable entree (initialisé par 0), et la sortie vers l'entrée de l'écriture du pipe si une commande suivante existe, et on lance la commande courante, dans le père, on mémorise la sortie pour l'entrée de la prochaine commande (entree = p[0] avec p le pipe utilisé), on incrémente num_cmd et on rentre une autre fois dans la boucle while.

Méthodologie de tests et quelques tests significatifs

La méthodologie adaptée pour faire les tests était d'imaginer des scénarios d'utilisation du minishell et comparer le résultat avec un vrai shell (bash : le shell par défaut sur Ubuntu), pour certaines procédures et fonctions utilisées, j'ai utilisé une programmation défensive (Exemples : lecture ou ouverture des fichiers, commande introuvable, utilisation de certaines commandes internes sans arguments etc.)

Voici quelques tests significatifs :

(Test vérifiant les questions 1,2 et 3)

```
minishell@1.1:~/home/anas/Desktop/testcode$ ls -l
total 88
-rw-rw-r-- 1 anas anas 3322 mai 23 00:21 cmdinternes.c
-rw-rw-r-- 1 anas anas 612 mai 23 00:21 cmdinternes.h
-rw-rw-r-- 1 anas anas 9096 mai 23 00:23 compMinishell.c
-rw-rw-r-- 1 anas anas 1429 mai 23 00:23 compMinishell.h
-rw-rw-r-- 1 anas anas 2877 avril 28 04:50 LisezMoi.html
-rw-rw-r-- 1 anas anas 2063 avril 28 04:50 LisezMoi.md
-rwxr-xr-x 1 anas anas 23344 mai 23 02:10 minishell
-rw-rw-r-- 1 anas anas 1906 mai 23 00:30 minishell.c
-rw-rw-r-- 1 anas anas 3407 mai 23 00:28 procMinishell.c
-rw-rw-r-- 1 anas anas 977 mai 23 00:29 procMinishell.h
-rw-rw-r-- 1 anas anas 5033 avril 28 04:50 readcmd.c
-rw-rw-r-- 1 anas anas 1897 avril 28 04:50 readcmd.h
-rw-rw-r-- 1 anas anas 2937 mai 23 01:00 signalMinishell.c
-rw-rw-r-- 1 anas anas 516 mai 23 00:59 signalMinishell.h
minishell@1.1:~/home/anas/Desktop/testcode$ echo anas
anas
minishell@1.1:~/home/anas/Desktop/testcode$
```

(Test vérifiant la question 4)

```
minishell@1.1:~/home/anas/Desktop/testcode$ cd
minishell@1.1:~/home/anas/Desktop/testcode$ cd home/anas
minishell@1.1:~/home/anas/Desktop/testcode$ exit
anas@Hunter:~/Desktop/testcode$
```

(Test vérifiant la question 5, ainsi que l'utilisation d'une redirection et de la commande list pour vérifier rapidement le résultat)

```
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
minishell@1.1:~/home/anas/Desktop/testcode$ sleep 1000 &
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
18                5281     actif     sleep 1000
minishell@1.1:~/home/anas/Desktop/testcode$ ps -aux | grep sleep
anas              5281    0.0  0.0 14572  792 pts/2    S+   02:20   0:00 sleep 1000
minishell@1.1:~/home/anas/Desktop/testcode$
```

(Test vérifiant la question 6 (Les commandes internes et ctrl-Z))

```
minishell@1.1:~/home/anas/Desktop/testcode$ sleep 1000 &
minishell@1.1:~/home/anas/Desktop/testcode$ sleep 1500 &
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     actif     sleep 1000
23                5287     actif     sleep 1500
minishell@1.1:~/home/anas/Desktop/testcode$ stop 22
Suspension du processus : ( Id = 22 | Commande = sleep 1000 )
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     suspendu  sleep 1000
23                5287     actif     sleep 1500
minishell@1.1:~/home/anas/Desktop/testcode$ stop 23
Suspension du processus : ( Id = 23 | Commande = sleep 1500 )
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     suspendu  sleep 1000
23                5287     suspendu  sleep 1500
minishell@1.1:~/home/anas/Desktop/testcode$ bg 22
Reprise du processus en arrière-plan: ( Id = 22 | Commande = sleep 1000 )
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     actif     sleep 1000
23                5287     suspendu  sleep 1500
```



```

minishell@1.1:~/home/anas/Desktop/testcode$ bg 22
Reprise du processus en arrière-plan: ( Id = 22 | Commande = sleep 1000  )
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     actif     sleep 1000
23                5287     suspendu  sleep 1500
minishell@1.1:~/home/anas/Desktop/testcode$ fg 23
Reprise du processus en avant-plan: ( Id = 23 | Commande = sleep 1500  )
^Z
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     actif     sleep 1000
23                5287     suspendu  sleep 1500
minishell@1.1:~/home/anas/Desktop/testcode$ fg 22
Reprise du processus en avant-plan: ( Id = 22 | Commande = sleep 1000  )
^Z
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
22                5286     suspendu  sleep 1000
23                5287     suspendu  sleep 1500
minishell@1.1:~/home/anas/Desktop/testcode$

```

```

minishell@1.1:~/home/anas/Desktop/testcode$
minishell@1.1:~/home/anas/Desktop/testcode$ ^Z
Pour quitter le minishell, utilisez la commande exit

minishell@1.1:~/home/anas/Desktop/testcode$ ^Z
Pour quitter le minishell, utilisez la commande exit

minishell@1.1:~/home/anas/Desktop/testcode$ ^Z
Pour quitter le minishell, utilisez la commande exit

minishell@1.1:~/home/anas/Desktop/testcode$

```

(Test vérifiant la question 7)

```

minishell@1.1:~/home/anas/Desktop/testcode$ sleep 2000 &
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
37                5321     actif     sleep 2000
minishell@1.1:~/home/anas/Desktop/testcode$ fg 37
Reprise du processus en avant-plan: ( Id = 37 | Commande = sleep 2000  )
^C
minishell@1.1:~/home/anas/Desktop/testcode$ list
Id minishell      PID      etat      ligne de commande lancée
minishell@1.1:~/home/anas/Desktop/testcode$

```

```
minishell@1.1:~/home/anas/Desktop/testcode$  
minishell@1.1:~/home/anas/Desktop/testcode$ ^C  
Pour quitter le minishell, utilisez la commande exit  
  
minishell@1.1:~/home/anas/Desktop/testcode$ ^C  
Pour quitter le minishell, utilisez la commande exit  
  
minishell@1.1:~/home/anas/Desktop/testcode$ ^C  
Pour quitter le minishell, utilisez la commande exit  
  
minishell@1.1:~/home/anas/Desktop/testcode$ █
```

(Test vérifiant la question 8)

```
minishell@1.1:~/home/anas/Desktop/testcode$ ls -l  
total 88  
-rw-rw-r-- 1 anas anas 3322 mai 23 00:21 cmdinternes.c  
-rw-rw-r-- 1 anas anas 612 mai 23 00:21 cmdinternes.h  
-rw-rw-r-- 1 anas anas 9096 mai 23 00:23 compMinishell.c  
-rw-rw-r-- 1 anas anas 1429 mai 23 00:23 compMinishell.h  
-rw-rw-r-- 1 anas anas 2877 avril 28 04:50 LisezMoi.html  
-rw-rw-r-- 1 anas anas 2063 avril 28 04:50 LisezMoi.md  
-rwxr-xr-x 1 anas anas 23344 mai 23 02:16 minishell  
-rw-rw-r-- 1 anas anas 1906 mai 23 00:30 minishell.c  
-rw-rw-r-- 1 anas anas 3407 mai 23 00:28 procMinishell.c  
-rw-rw-r-- 1 anas anas 977 mai 23 00:29 procMinishell.h  
-rw-rw-r-- 1 anas anas 5033 avril 28 04:50 readcmd.c  
-rw-rw-r-- 1 anas anas 1897 avril 28 04:50 readcmd.h  
-rw-rw-r-- 1 anas anas 2937 mai 23 01:00 signalMinishell.c  
-rw-rw-r-- 1 anas anas 516 mai 23 00:59 signalMinishell.h  
minishell@1.1:~/home/anas/Desktop/testcode$ echo anas titah > test1.txt  
minishell@1.1:~/home/anas/Desktop/testcode$ cat <test1.txt >test2.txt  
minishell@1.1:~/home/anas/Desktop/testcode$ cat <test1.txt  
anas titah  
minishell@1.1:~/home/anas/Desktop/testcode$ cat <test2.txt  
anas titah  
minishell@1.1:~/home/anas/Desktop/testcode$ █
```



```

minishell@1.1:~/home/anas/Desktop/testcode$ ls -l
total 96
-rw-rw-r-- 1 anas anas 3322 mai 23 00:21 cmdinternes.c
-rw-rw-r-- 1 anas anas 612 mai 23 00:21 cmdinternes.h
-rw-rw-r-- 1 anas anas 9096 mai 23 00:23 compMinishell.c
-rw-rw-r-- 1 anas anas 1429 mai 23 00:23 compMinishell.h
-rw-rw-r-- 1 anas anas 2877 avril 28 04:50 LisezMoi.html
-rw-rw-r-- 1 anas anas 2063 avril 28 04:50 LisezMoi.md
-rwxr-xr-x 1 anas anas 23344 mai 23 02:16 minishell
-rw-rw-r-- 1 anas anas 1906 mai 23 00:30 minishell.c
-rw-rw-r-- 1 anas anas 3407 mai 23 00:28 procMinishell.c
-rw-rw-r-- 1 anas anas 977 mai 23 00:29 procMinishell.h
-rw-rw-r-- 1 anas anas 5033 avril 28 04:50 readcmd.c
-rw-rw-r-- 1 anas anas 1897 avril 28 04:50 readcmd.h
-rw-rw-r-- 1 anas anas 2937 mai 23 01:00 signalMinishell.c
-rw-rw-r-- 1 anas anas 516 mai 23 00:59 signalMinishell.h
-rwx----- 1 anas anas 11 mai 23 02:36 test1.txt
-rwx----- 1 anas anas 11 mai 23 02:37 test2.txt
minishell@1.1:~/home/anas/Desktop/testcode$ cat <test3.txt
Erreur d'ouverture: No such file or directory
minishell@1.1:~/home/anas/Desktop/testcode$

```

(Test vérifiant les questions 9 et 10)

```

minishell@1.1:~/home/anas/Desktop/testcode$ ls
cmdinternes.c  compMinishell.h  minishell  procMinishell.h  signalMinishell.c  test2.txt
cmdinternes.h  LisezMoi.html  minishell.c  readcmd.c  signalMinishell.h
compMinishell.c  LisezMoi.md  procMinishell.c  readcmd.h  test1.txt
minishell@1.1:~/home/anas/Desktop/testcode$ ls | wc -l
16
minishell@1.1:~/home/anas/Desktop/testcode$ ls | wc -l > resultat.txt
minishell@1.1:~/home/anas/Desktop/testcode$ cat <resultat.txt
16
minishell@1.1:~/home/anas/Desktop/testcode$

```

```

minishell@1.1:~/home/anas/Desktop/testcode$ ls
cmdinternes.c  compMinishell.h  minishell  procMinishell.h  resultat.txt  test1.txt
cmdinternes.h  LisezMoi.html  minishell.c  readcmd.c  signalMinishell.c  test2.txt
compMinishell.c  LisezMoi.md  procMinishell.c  readcmd.h  signalMinishell.h
minishell@1.1:~/home/anas/Desktop/testcode$ ls > temp.txt
minishell@1.1:~/home/anas/Desktop/testcode$ cat <temp.txt | grep shell | wc -l
8
minishell@1.1:~/home/anas/Desktop/testcode$ cat <temp.txt | grep shell | wc -l > resultat2.txt
minishell@1.1:~/home/anas/Desktop/testcode$ cat <resultat2.txt
8
minishell@1.1:~/home/anas/Desktop/testcode$ cat <temp.txt >temp2.txt
minishell@1.1:~/home/anas/Desktop/testcode$ cat temp.txt temp2.txt | grep shell | wc -l
16
minishell@1.1:~/home/anas/Desktop/testcode$

```

Conclusion

Ce projet est un bon exercice pour tester les connaissances requises dans les séances du TD et TP, ainsi, il peut être amélioré en ajoutant d'autres fonctionnalités utiles comme les variables et les commandes comme history, et on peut aussi penser à réaliser ce que les flèches haut/bas et droite/gauche font dans un shell standard.