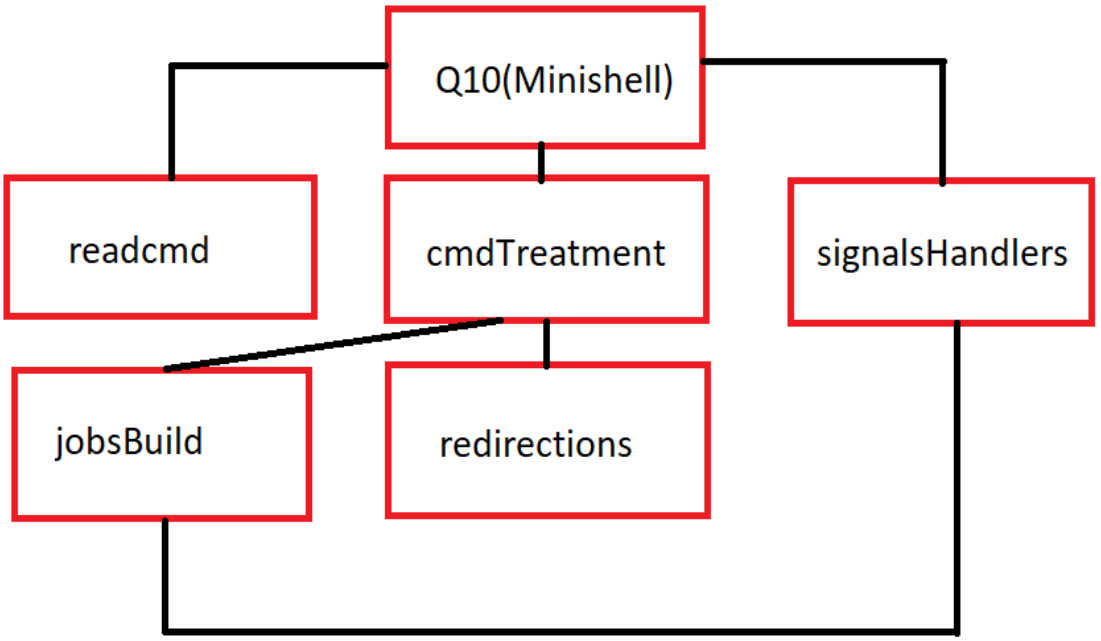
CHOUKRANI Youssef

Rapport MiniShell

Département Sciences du numérique 2020-2021

Choix de conception :



Les fichiers présents dans l'architecture :

--Q10 : représente la version finale du minishell et c'est le code qui contient le main.

--cmdTreatment : s'occupe de traiter la commande lu au clavier.

--signalsHandlers : contient le traitement des différents signaux.

--redirections : contient les méthodes pour rediriger les entrées et sorties.

--jobsBuild : Permet de construire un tableau qui va servir à stocker les informations à propos des processus en cours.

La méthodologie de test suivie :

Pour chaque manipulation, il fallait comparer le résultat de mon minishell avec le résultat du shell effectif de ubuntu.

Je tiens à vous informer que le code n'était pas propre avant ma réponse à la dernière question où j'ai essayé de tout bien mettre en place et surtout avoir un main avec peu de lignes. En effet, j'avais un peu baclé le travail avant d'atteindre la dernière question. Et après avoir finalisé cette dernière, j'ai réorganisé toute l'architecture et pris en considération les retours de mon professeur pour corriger ce qui m'a été signalé, m'ai j'ai appliqué celà que au code de la question 10.

Les Tests réalisés :

J’ai créé tout d’abord une commande “demc” qui sleep et en même temps print les chiffres de 0 à 9 pour vérifier le comportement du minishell.

--Pour bg :
Je lance la commande “demc”, j’effectue un “ctrl+z”, ensuite je lance “bg + id” du processus, à ce moment là pendant que demc continue à s’exécuter, je lance “ls” et le résultat de “ls” s’affiche instantanément, ce qui signifie que “demc” s'exécutait bel et bien en arrière plan.

```
choukrani@sec:/home/ychoukra/sectp/secprojetef ./demc
0
1
2
^Z
Processus suspendu
choukrani@sec:/home/ychoukra/sectp/secprojetef jobs
Id       Pid     Etat           Commande
[1]      18696   Suspendu       ./demc

choukrani@sec:/home/ychoukra/sectp/secprojetef bg 1
Repris en arrière-plan
choukrani@sec:/home/ychoukra/sectp/secprojetef 3
4
5
ls
choukrani.youssef.tar  f1dp         jobsBuild.h  Q1.c    Q3c   Q5.c    Q67c   Q8c    Q9c            redirections.c
cmdTreatment.c         f2           lulu.c       Q2c     Q3.c  Q5kac   Q67.c  Q8.c   Q9.c           redirections.h
cmdTreatment.h         fournitures  Q10c         Q2.c    Q4c   Q5ka.c  Q6c    Q92c   rapport+67.pdf signalsHandlers.c
demc                   f.tar        Q10.c        Q2c.c   Q4.c  Q67ac   Q8ac   Q92.c  readcmd.c      signalsHandlers.h
dormir_et_marquer.c    jobsBuild.c  Q1c          Q2.pdf  Q5c   Q67a.c  Q8a.c  Q9ac   readcmd.h      toto.c

choukrani@sec:/home/ychoukra/sectp/secprojetef 6
7
8
9
jobs
La liste est vide !

choukrani@sec:/home/ychoukra/sectp/secprojetef
```

--Pour fg:

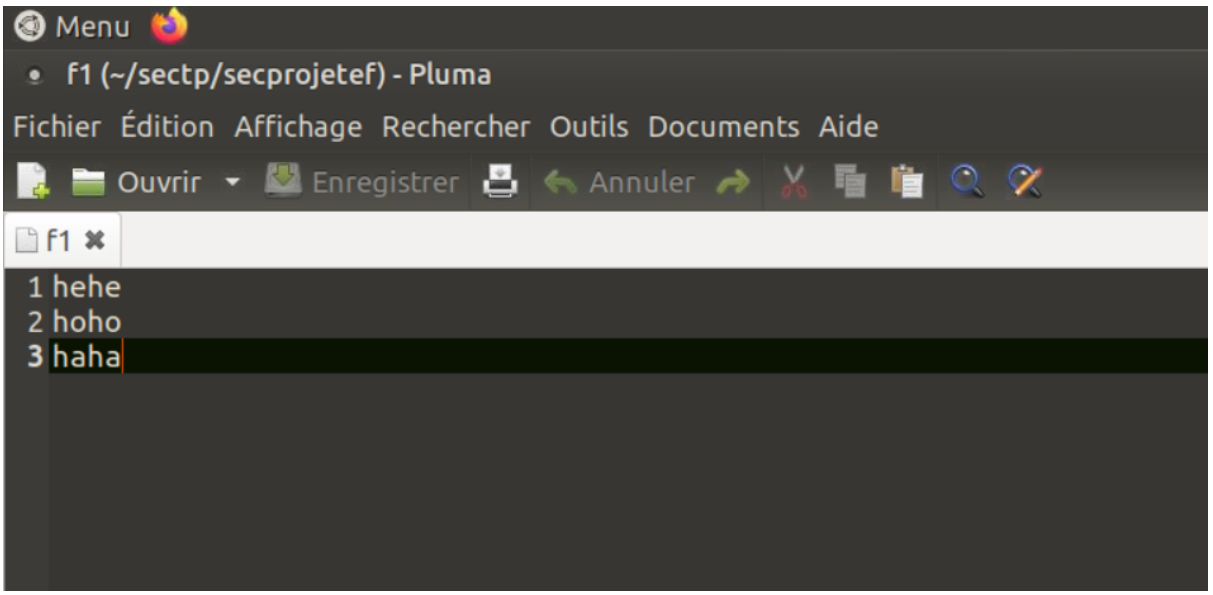
Même manipulation que bg, mais cette fois, le résultat de “ls” ne s’affiche pas instantanément et attend la fin de l’exécution de “demc” pour ensuite s’afficher.

```
ychoukra@bulbizarre:~/sectp/secprojetef$ ./Q10c

choukrani@sec:/home/ychoukra/sectp/secprojetef ./demc
0
1
2
^Z
Processus suspendu
choukrani@sec:/home/ychoukra/sectp/secprojetef jobs
Id       Pid     Etat           Commande
[1]      22377   Suspendu       ./demc

choukrani@sec:/home/ychoukra/sectp/secprojetef fg 1
3
4
ls
5
6
7
8
9
Repris en Avant-plan
choukrani@sec:/home/ychoukra/sectp/secprojetef choukrani.youssef.tar  f1dp         jobsBuild.h  Q1.c    Q3c   Q5.c    Q67c   Q8c    Q9c
    redirections.c
cmdTreatment.c         f2           lulu.c       Q2c     Q3.c  Q5kac   Q67.c  Q8.c   Q9.c           redirections.h
cmdTreatment.h         fournitures  Q10c         Q2.c    Q4c   Q5ka.c  Q6c    Q92c   rapport+67.pdf signalsHandlers.c
demc                   f.tar        Q10.c        Q2c.c   Q4.c  Q67ac   Q8ac   Q92.c  readcmd.c      signalsHandlers.h
dormir_et_marquer.c    jobsBuild.c  Q1c          Q2.pdf  Q5c   Q67a.c  Q8a.c  Q9ac   readcmd.h      toto.c

choukrani@sec:/home/ychoukra/sectp/secprojetef
```

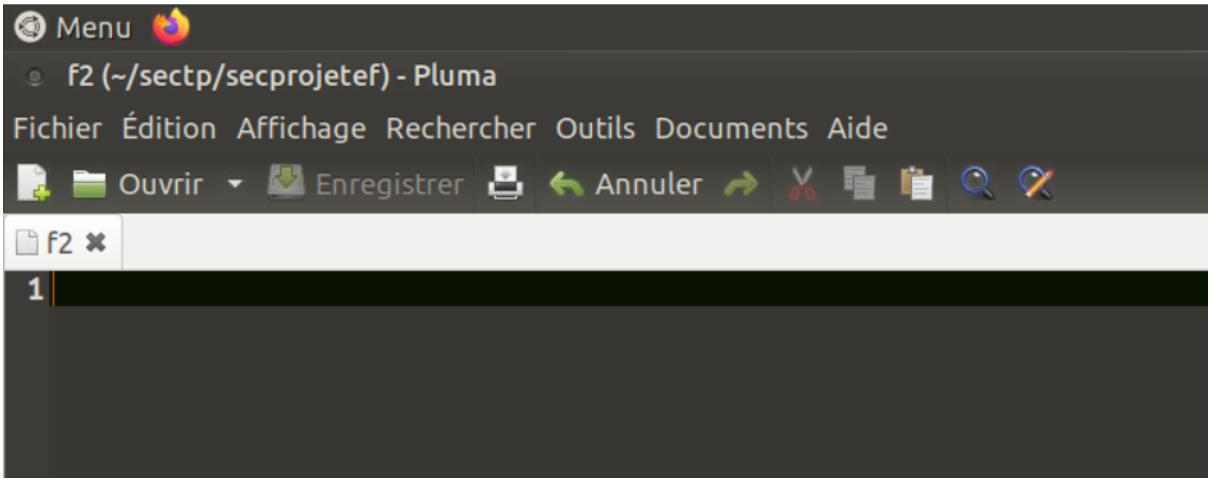
--Pour les redirections:

J’ai crée deux fichiers : f1 et f2 et j’ai lancé la commande cat <f1> f2, le résultat est le suivant:

Le fichier f1



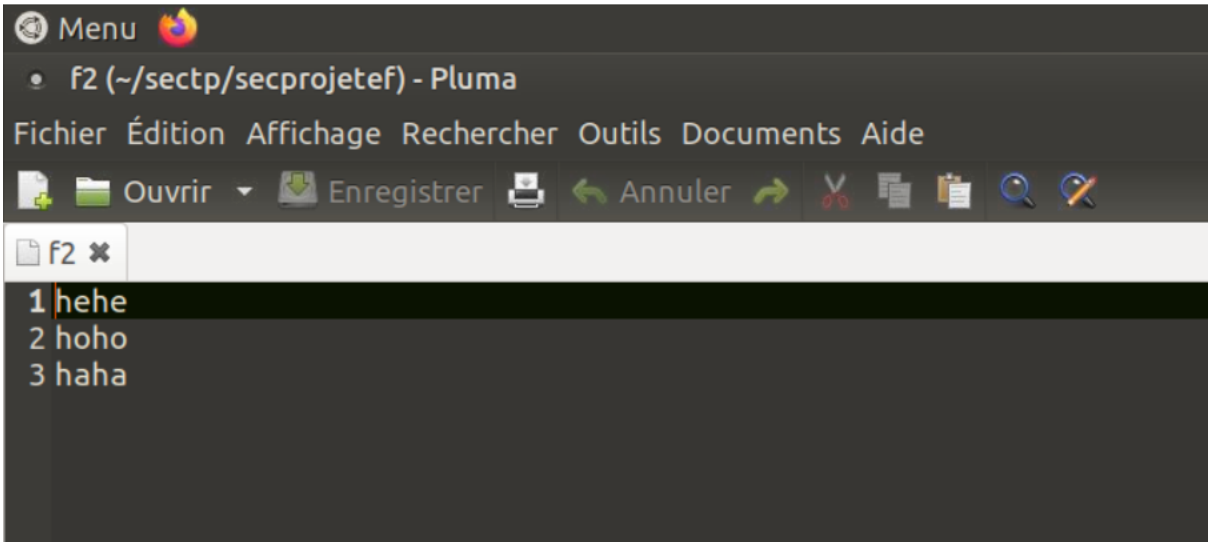
Le fichier f2



La commande :

```
choukrani@sec:/home/ychoukra/sectp/secprojetef cat <f1> f2
```

Le résultat:



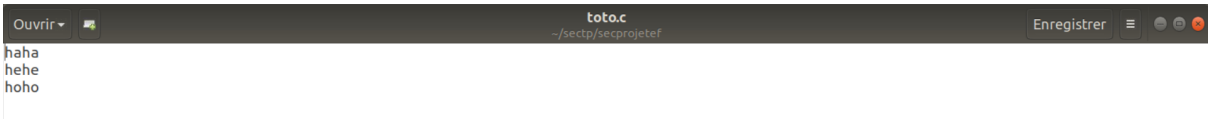
--Pour les tubes :

J’ai testé les deux commandes présentes dans le sujet et les résultats sont les suivant

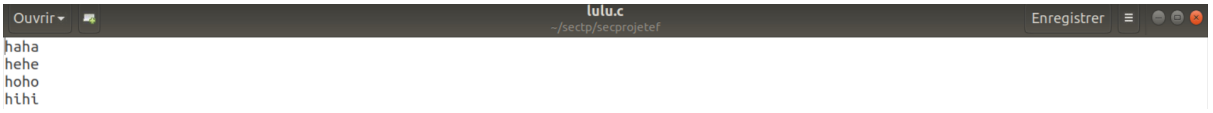
```
choukrani@sec:/home/ychoukra/sectp/secprojetef ls
choukrani.youssef.tar  f1           jobsBuild.h  Q1.c    Q3c   Q5.c    Q67c  Q8c   Q9c             redirections.c
cmdTreatment.c         f2           lulu.c       Q2c     Q3.c  Q5kac   Q67.c Q8.c  Q9.c            redirections.h
cmdTreatment.h         fournitures  Q10c         Q2.c    Q4c   Q5ka.c  Q6c   Q92c  rapport+67.pdf  signalsHandlers.c
demc                   f.tar        Q10.c        Q2c.c   Q4.c  Q67ac   Q8ac  Q92.c readcmd.c       signalsHandlers.h
dormir_et_marquer.c    jobsBuild.c  Q1c          Q2.pdf  Q5c   Q67a.c  Q8a.c Q9ac  readcmd.h       toto.c

choukrani@sec:/home/ychoukra/sectp/secprojetef ls | wc -l
50

choukrani@sec:/home/ychoukra/sectp/secprojetef
```

```
choukrani@sec:/home/ychoukra/sectp/secprojetef cat toto.c lulu.c | grep int | wc -l
0

choukrani@sec:/home/ychoukra/sectp/secprojetef
```

Fichier toto.c :



Fichier lulu.c :



Question 2:

Avec le code écrit dans la question 1, le processus shell lance un fils, puis se met immédiatement en attente de lecture de la prochaine ligne de commande.
Il est possible alors que l'affichage de l'invite de commande précède ou se mêle à l'exécution du processus fils comme le montre la situation suivante :
J'ai créé une commande (./demc) qui permet de sleep et en même temps de marquer chaque seconde qui passe, et dont le code est le suivant :

```c
int main() {
        int s=10;
        int i =0;
        while (i<s) {
                sleep(1);
                printf("%d\n",i);
                i++;
        }
}
```

En exécutant le minishell conçu à l'issue de la première question , et en lançant la commande ./demc, j'ai toujours la possibilité d'exécuter d'autre commande comme le ls, et c'est ce qui est représenté dans la figure suivante :

On remarque que la ligne que propose le shell s'affiche avant même l'exécution de la commande pour toutes les commandes qui suivent la première (./demc).
⇒ L'affichage de l'invite de commande précède/se mêle à l'exécution de la commande.

Question 6 et 7 : Pour cette question, il est nécessaire de stocker les commandes lancées pour les afficher et pouvoir manipuler par la suite, d'où l'idée de créer un tableau qui stock l'identifiant de la commande, son pid, son état, et la commande en soit. J'ai créé donc un fichier jobsBuild contenant les différentes fonctions et méthodes qui permettent de manipuler cette liste afin de pouvoir l'utiliser proprement dans le programme main.
En ce qui concerne les commandes ajoutées (jobs, stop, bg, fg), j'ai délégué leur traitement au processus père (le shell).
Ensuite j'ai utilisé un handler de sigchld comme ça été conseillé pour alerter le père à chaque fois du changement d'état de ses fils, et j'ai masqué le ctrl+z et ctrl+c pour les processus en arrière plan.