



*Rapport de projet Systèmes d'Exploitation Centralisés*  
**Réalisation d'un minishell**

Priscilia Gonthier  
Groupe M

25 Mai 2022

Département Sciences du Numérique - Première année  
2021-2022

## Table des matières

Introduction . . . . .	2
1 Architecture de l'Application . . . . .	2
2 Réponses aux questions et conception. . . . .	2
3 Méthodologie des tests et tests significatifs . . . . .	4

# Introduction

Ce projet a pour but de mettre en pratique les notions vues en TD et TP autour de la gestion des processus, des signaux et des Entrées/Sorties au travers du développement d'un interpréteur de commande simplifié (minishell).

## 1 Architecture de l'Application

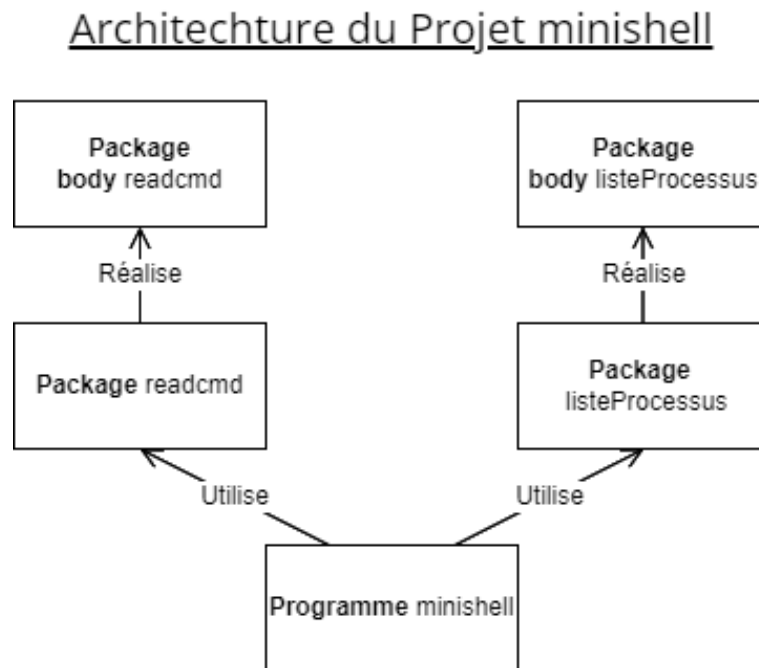


FIGURE 1 – Architecture du projet minishell

## 2 Réponses aux questions et conception

### Questions 1 à 3 :

Afin de réaliser le minishell, j'ai utilisé une boucle infinie. À chaque tour de boucle, le prompt est affiché, puis l'entrée au clavier est lu par la commande **readcmd()**. La commande est exécuter en faisant tout d'abord un **fork()**, puis dans le processus fils un **execvp()**.

Avec cette manière de faire, le minishell n'attendait pas la fin d'exécution de ses processus fils, et il pouvait afficher le prompteur avant ou au milieu de l'affichage du processus fils comme dans la figure 2.

Le minishell va donc attendre la fin du processus fils lancé grâce à la fonction **waitpid()** pour éviter que l'affichage du prompteur chevauche celui de l'exécution.

```

Minishell:/home/pgonthie/Bureau/minishell_final>>> ls
Minishell:/home/pgonthie/Bureau/minishell_final>>> copie.txt    listeProcessus.h  Makefile    minishell.c  Q2.pdf
      readcmd.h  test.txt
listeProcessus.c  listeProcessus.o  minishell  minishell.o  readcmd.c  readcmd.o
pwd
Minishell:/home/pgonthie/Bureau/minishell_final>>> /home/pgonthie/Bureau/minishell_final
sleep 10
Minishell:/home/pgonthie/Bureau/minishell_final>>> sds
Minishell:/home/pgonthie/Bureau/minishell_final>>> sds
Minishell:/home/pgonthie/Bureau/minishell_final>>> cd
[i] Processus arrêté sleep 10
Minishell:/home/pgonthie>>> █

```

FIGURE 2 – Affichage abberant du prompteur

#### Question 4 :

J'ai ensuite rajouté la commande interne **exit**, qui met la variable de boucle à **false** afin de sortir du minishell et de quitter le programme. Pour la commande interne **cd**, j'ai utilisé la fonction **chdir()**, et géré comme dans le shell le retour à HOME avec la fonction **getenv()** qui retourne un pointeur vers l'environnement dont le nom est donné en paramètre.

#### Question 5 :

Pour lancer des commandes en tâche de fond, j'ai utilisé l'attribut **backgrounded** de la l'enregistrement **cmdline**, et si cette attribut est nul, la commande est lancé en premier plan et le père attend la fin du fils, sinon la commande est lancé en arrière plan et le père n'attend pas la fin du fils pour continuer son execution.

#### Question 6 :

Pour implémenter les commandes internes **lj**, **sj**, **bg** et **fg**, j'ai créé un module **listeProcessus**, qui regroupe les fonctions que j'ai utilisé pour la gestion des processus. J'ai aussi utiliser un handler du signal **SIGCHLD** pour gérer la liste de processus, lorsqu'un processus en arrière plan se termine. Au niveau de la liste de processus, j'ai fait le choix d'une liste de structure contenant l'id, le pid, l'tat et la commande du processus. Je n'ai pas réussi à supprimé les warnings et erreurs valgrind concernant l'allocation des listes, même en essayant de l'allouer classiquement (cela me rajoutait des erreurs de segmentation) et non dynamiquement, je l'ai donc laissé telle quel.

#### Questions 7 et 8 :

Pour faire fonctionner les commandes **ctrl-c** et **ctrl-z** seulement sur les processus au premier plan, j'ai créé des handlers pour les signaux **SIGINT** et **SIGTSTP**. Cela a permis que le minishell continue de fonctionner et que le processus au premier plan s'arrete ou se suspend. Cependant j'avais toujours le problème, que les processus en arrière plan s'arretent ou se suspendent en même temps. J'ai donc du masquer les signaux **SIGINT** et **SIGTSTP** au niveau du lancement des processus fils. Cela permet que les fils ne traitent pas ces signaux, c'est seulement le minishell qui le gère avec les handlers que j'ai défini. Pour suspendre le minishell, j'ai créé la commande interne **susp**, qui utilise **getpid()** pour obtenir le pid du minishell et effectuer un **kill()** avec le signal **SIGSTOP**. Pour reprendre le minishell, il y a une erreur que je n'ai pas

réussit à corriger et lorsque l'on rentre **kill -18 identifiant\_du\_minishell** (avec 18 correspondant au signal SIGCONT) on devrait pouvoir reprendre la minishell, sauf qu'il se remet en suspension.

### Question 9 :

J'ai implémenté les redirections en créant 2 sous-fonctions **redirigerSortie()** et **redirigerEntree()**. Pour rediriger la sortie standard vers un fichier, on ouvre le fichier avec la possibilité d'écriture, puis avec la fonction **dup2()** on duplique le descripteur de fichier vers la sortie standard. On fait de manière similaire la redirection de l'entrée en ouvrant un fichier en lecture, puis en dupliquant le descripteur de fichier vers l'entrée standard grâce à **dup2()**.

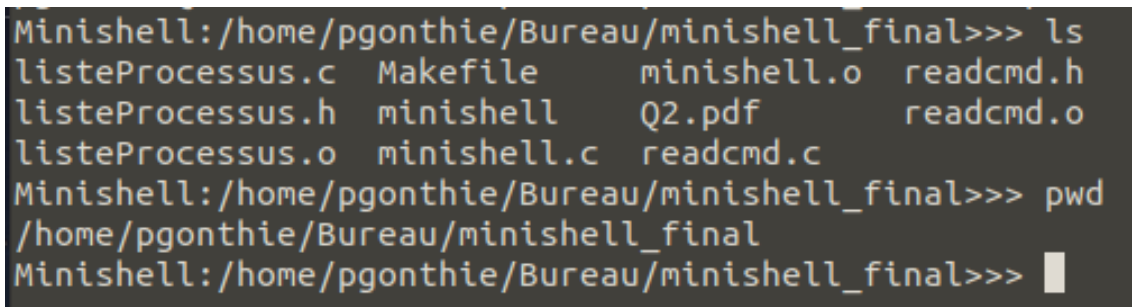
### Questions 10 et 11 :

Pour permettre l'utilisation de tubes, je parcours tout d'abord le nombre de commandes pour connaître le nombre de tubes nécessaires. Je crée ensuite les tubes avec la commande **pipe()**. Je parcours ensuite les commandes, et crée des processus fils. Puis je redirige les entrées et sorties vers les différents tubes ou vers des fichiers grâce à la commande **dup2()** d'exécuter les commandes. Le processus père quand à lui ferme tous les tubes et attends tous ses fils.

## 3 Méthodologie des tests et tests significatifs

Afin de tester les différentes fonctions de mon minishell, j'ai lancé les différentes commandes au fur et à mesure de leur implémentation pour vérifier leur fonctionnement.

### 3.1 Test des questions 1 à 3 :



```
Minishell:/home/pgonthie/Bureau/minishell_final>>> ls
listeProcessus.c  Makefile      minishell.o  readcmd.h
listeProcessus.h  minishell    Q2.pdf      readcmd.o
listeProcessus.o  minishell.c  readcmd.c
Minishell:/home/pgonthie/Bureau/minishell_final>>> pwd
/home/pgonthie/Bureau/minishell_final
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 3 – Test de lancement des commandes et d'attente de la fin des fils

### 3.2 Test de la question 4 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> cd ..
Minishell:/home/pgonthie/Bureau>>> pwd
/home/pgonthie/Bureau
Minishell:/home/pgonthie/Bureau>>> cd minishell_final
Minishell:/home/pgonthie/Bureau/minishell_final>>> cd
Minishell:/home/pgonthie>>> exit
pgonthie@n7-ens-lnx018:~/Bureau/minishell_final$
```

FIGURE 4 – Test des commandes internes exit et cd

### 3.3 Test de la questions 5 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID    PENDING   BLOCKED   IGNORED   CAUGHT  STAT  TTY      TIME COMMAND
50224  101218   00000000   00010000   00380004  4b817efb Ss    pts/1    0:00 bash
50224  105443   00000000   00000000   00000000   00090002 S+    pts/1    0:00 ./minishell
50224  105473   00000000   00080002   00000000   00000000 S+    pts/1    0:00 sleep 100
50224  105487   00000000   00080002   00000000   <f3d1fef9 R+    pts/1    0:00 ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> pwd
/home/pgonthie/Bureau/minishell_final
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 200 &
[2] 105516
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID    PENDING   BLOCKED   IGNORED   CAUGHT  STAT  TTY      TIME COMMAND
50224  101218   00000000   00010000   00380004  4b817efb Ss    pts/1    0:00 bash
50224  105443   00000000   00000000   00000000   00090002 S+    pts/1    0:00 ./minishell
50224  105473   00000000   00080002   00000000   00000000 S+    pts/1    0:00 sleep 100
50224  105516   00000000   00080002   00000000   00000000 S+    pts/1    0:00 sleep 200
50224  105530   00000000   00080002   00000000   <f3d1fef9 R+    pts/1    0:00 ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>>
```

FIGURE 5 – Test de lancement de commandes en tâche de fond

### 3.4 Tests de la question 6 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 10 &
[1] 105765
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 20 &
[2] 105778
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      105765     ACTIF     sleep 10
  2      105778     ACTIF     sleep 20
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 6 – Test de la commande lj

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 100 &
[1] 144990
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 200 &
[2] 145027
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      144990     ACTIF     sleep 100
  2      145027     ACTIF     sleep 200
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING  BLOCKED  IGNORED          CAUGHT  STAT  TTY      TIME COMMAND
50224  143663  0000000000000000  000000000010000  0000000000380004  0000000004b817efb Ss   pts/1    0:00 bash
50224  144954  0000000000000000  0000000000000000  0000000000000000  0000000000900002 S+   pts/1    0:00 ./minishell
50224  144990  0000000000000000  0000000000080002  0000000000000000  0000000000000000 S+   pts/1    0:00 sleep 100
50224  145027  0000000000000000  0000000000080002  0000000000000000  0000000000000000 S+   pts/1    0:00 sleep 200
50224  145040  0000000000000000  0000000000080002  0000000000000000  00000001f3d1fef9 R+   pts/1    0:00 ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> sj 1
[1] Processus en arrière plan suspendu : sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      144990     SUSPENDU  sleep 100
  2      145027     ACTIF     sleep 200
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING  BLOCKED  IGNORED          CAUGHT  STAT  TTY      TIME COMMAND
50224  143663  0000000000000000  000000000010000  0000000000380004  0000000004b817efb Ss   pts/1    0:00 bash
50224  144954  0000000000000000  0000000000000000  0000000000000000  0000000000900002 S+   pts/1    0:00 ./minishell
50224  144990  0000000000000000  0000000000080002  0000000000000000  0000000000000000 T+   pts/1    0:00 sleep 100
50224  145027  0000000000000000  0000000000080002  0000000000000000  0000000000000000 S+   pts/1    0:00 sleep 200
50224  145087  0000000000000000  0000000000080002  0000000000000000  00000001f3d1fef9 R+   pts/1    0:00 ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 7 – Test de la commande sj

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 100 &
[1] 144406
Minishell:/home/pgonthie/Bureau/minishell_final>>> sj 1
[1] Processus en arrière plan suspendu : sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      144406     SUSPENDU  sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING  BLOCKED  IGNORED          CAUGHT  STAT  TTY      TIME COMMAND
50224  143663  0000000000000000  000000000010000  0000000000380004  0000000004b817efb Ss   pts/1    0:00 bash
50224  144390  0000000000000000  0000000000000000  0000000000000000  0000000000900002 S+   pts/1    0:00 ./minishell
50224  144406  0000000000000000  0000000000080002  0000000000000000  0000000000000000 T+   pts/1    0:00 sleep 100
50224  144439  0000000000000000  0000000000080002  0000000000000000  00000001f3d1fef9 R+   pts/1    0:00 ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> bg 1
[1] Processus repris en arrière plan : sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      144406     ACTIF     sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING  BLOCKED  IGNORED          CAUGHT  STAT  TTY      TIME COMMAND
50224  143663  0000000000000000  000000000010000  0000000000380004  0000000004b817efb Ss   pts/1    0:00 bash
50224  144390  0000000000000000  0000000000000000  0000000000000000  0000000000900002 S+   pts/1    0:00 ./minishell
50224  144406  0000000000000000  0000000000080002  0000000000000000  0000000000000000 S+   pts/1    0:00 sleep 100
50224  144489  0000000000000000  0000000000080002  0000000000000000  00000001f3d1fef9 R+   pts/1    0:00 ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 8 – Test de la commande bg

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 20 &
[1] 144762
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      144762      ACTIF      sleep 20
Minishell:/home/pgonthie/Bureau/minishell_final>>> sj 1
[1] Processus en arrière plan suspendu : sleep 20
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING      BLOCKED      IGNORED      CAUGHT      STAT      TTY      TIME      COMMAND
50224  143663  000000000  000000000010000  0000000000380004  000000004b817efb  Ss      pts/1      0:00      bash
50224  144731  000000000  000000000000000  0000000000000000  000000000090002  S+      pts/1      0:00      ./minishell
50224  144762  000000000  000000000080002  0000000000000000  000000000000000  T+      pts/1      0:00      sleep 20
50224  144800  000000000  000000000080002  0000000000000000  000000001f3d1fef9  R+      pts/1      0:00      ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> fg 1
sleep 20
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 9 – Test de la commande fg

### 3.5 Tests de la question 7 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 100
^Z
[1]+ Suspension 107535 sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      107535      SUSPENDU      sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING      BLOCKED      IGNORED      CAUGHT      STAT      TTY      TIME      COMMAND
50224  101218  000000000  000100000  003800004  4b817efb  Ss      pts/1      0:00      bash
50224  107513  000000000  000000000  000000000  000900002  S+      pts/1      0:00      ./minishell
50224  107535  000800000  000800002  000000000  000000000  T+      pts/1      0:00      sleep 100
50224  107557  000000000  000800002  000000000  <f3d1fef9  R+      pts/1      0:00      ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 10 – Test de la commande ctrl-z

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> ls
listeProcessus.c  Makefile      minishell.o  readcmd.h
listeProcessus.h  minishell     Q2.pdf       readcmd.o
listeProcessus.o  minishell.c   readcmd.c
Minishell:/home/pgonthie/Bureau/minishell_final>>> susp

[1]+ Arrêté ./minishell
pgonthie@n7-ens-lnx018:~/Bureau/minishell_final$ ps -s
  UID      PID      PENDING      BLOCKED      IGNORED      CAUGHT      STAT      TTY      TIME      COMMAND
50224  110978  000000000  000100000  003800004  4b817efb  Ss      pts/0      0:00      bash
50224  111026  000000000  000000000  000000000  000900002  T       pts/0      0:00      ./minishell
50224  111061  000000000  000000000  000000000  <f3d1fef9  R+      pts/0      0:00      ps -s
pgonthie@n7-ens-lnx018:~/Bureau/minishell_final$ kill -18 111026
pgonthie@n7-ens-lnx018:~/Bureau/minishell_final$ Reprise du minishell
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 11 – Test de la commande susp



### 3.6 Test de la question 8 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 100 &
[1] 111491
Minishell:/home/pgonthie/Bureau/minishell_final>>> sleep 10
^C
Minishell:/home/pgonthie/Bureau/minishell_final>>> lj
  ID      PID      ETAT      COMMANDE
  1      111491     ACTIF     sleep 100
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING      BLOCKED      IGNORED      CAUGHT      STAT      TTY      TIME      COMMAND
50224  110978  00000000  00010000  00380004  4b817efb  Ss      pts/0      0:00      bash
50224  111474  00000000  00000000  00000000  00090002  S+      pts/0      0:00      ./minishell
50224  111491  00000002  00080002  00000000  00000000  S+      pts/0      0:00      sleep 100
50224  111528  00000000  00080002  00000000  <f3d1fef9  R+      pts/0      0:00      ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> ^C
Minishell:/home/pgonthie/Bureau/minishell_final>>> ps -s
  UID      PID      PENDING      BLOCKED      IGNORED      CAUGHT      STAT      TTY      TIME      COMMAND
50224  110978  00000000  00010000  00380004  4b817efb  Ss      pts/0      0:00      bash
50224  111474  00000000  00000000  00000000  00090002  S+      pts/0      0:00      ./minishell
50224  111491  00000002  00080002  00000000  00000000  S+      pts/0      0:00      sleep 100
50224  111690  00000000  00080002  00000000  <f3d1fef9  R+      pts/0      0:00      ps -s
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 12 – Test de la commande ctrl-c

### 3.7 Test de la question 9 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> ls
listeProcessus.c  Makefile      minishell.o  readcmd.h
listeProcessus.h  minishell     Q2.pdf       readcmd.o
listeProcessus.o  minishell.c   readcmd.c
Minishell:/home/pgonthie/Bureau/minishell_final>>> echo Bonjour >test.txt
Minishell:/home/pgonthie/Bureau/minishell_final>>> cat <test.txt >copie.txt
Minishell:/home/pgonthie/Bureau/minishell_final>>> cat <test.txt
Bonjour
Minishell:/home/pgonthie/Bureau/minishell_final>>> cat <copie.txt
Bonjour
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 13 – Test de la redirection

### 3.8 Test de la question 10 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> ls | wc -l
13
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 14 – Test des tubes simples



### 3.9 Test de la question 11 :

```
Minishell:/home/pgonthie/Bureau/minishell_final>>> cat minishell.c test.txt | grep int | wc -l  
65  
Minishell:/home/pgonthie/Bureau/minishell_final>>> █
```

FIGURE 15 – Test des pipelines