

Rapport S.E.C

Etape 1: n'importe quel commander ou exit pour sortir

Etape 2:

> ls

```
> ls
Makefile      copier.o      mac2.txt      minishell.o   readcmd.h     test_readcmd.c tube3
cat           dest         minishell     output        readcmd.o     tube1         tube4
copier.c      mac1.txt     minishell.c   readcmd.c     test          tube2         tube5
Terminaison du fils de pid : 2648
> █
```

> pwd

```
> pwd
/Users/sakunwithanage/Desktop/minishell
Terminaison du fils de pid : 2661
> █
```

> cat minishell.c

```
> cat minishell.c
#include <stdio.h>
#include <stdlib.h>
#include "readcmd.h"
#include <stdbool.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
#include <signal.h>
#include <sys/fcntl.h>

void affichePrompt() {
    printf("> ");
}

void traitementSIGCHLD(int S) {
    int status;
    pid_t pid = waitpid(-1, &status, WNOHANG);
    if (pid == -1) {
        printf("Erreur de terminaison du fils\n");
    }
}
```

On voit que les commandes sont bien exécutées et les résultats correspondent bien à ceux du terminal normal.

Etape 3:

```
(base) benjamin@benjamin:~/minishell$ sleep 10 &
> sleep 10&
> ls
Makefile      copier.o      mac2.txt      minishell.o   readcmd.h     test_readcmd.c tube3
cat           dest         minishell     output        readcmd.o     tube1          tube4
copier.c      mac1.txt     minishell.c   readcmd.c     test          tube2          tube5
Terminaison du fils de pid : 2785
> █
```

On voit que “sleep 10” est lancé en arrière plan et on peut quand même exécuter la commande ls.

Etape 4:

```
> sleep 10&
> sleep 50
```

Etape 6:

```
> sleep 10&
> ls
```

Affiche la fin de ls en premier puis celui de sleep

Etape 7:

```
> ls
Makefile      copier.o      mac2.txt      minishell.o   readcmd.h     test_readcmd.c tube3
cat           dest         minishell     output        readcmd.o     tube1          tube4
copier.c      mac1.txt     minishell.c   readcmd.c     test          tube2          tube5
Le processus fils avec PID 4210 s'est terminé avec le code de sortie 0.
> █
```

```
> sleep 10
^ZLe processus fils avec PID 4193 a été suspendu.
Ctrl + Z
^C> Ctrl + C
Le processus fils avec PID 4193 s'est terminé suite à un signal 2.
```

Etape 10:

Comme sur le capture d’écran haut dessus on affiche que le processus a été suspendu. Même si à ce moment là Ctrl Z et Ctrl C ne fonctionnait pas de la même manière. Ainsi si on avait utilisé ces commandes cela aurait arrêté le programme minishell.

Etape 12: > sleep 10
> Ctrl C

Tout le programme minishell s'arrête.

Etape 13:

```
> sleep 10
^C Ctrl + C
Le processus fils avec PID 4431 s'est terminé suite à un signal 2.
> sleep 10
^Z Le processus fils avec PID 4438 a été suspendu.
Ctrl + Z
```

Etape 14:

```
> sleep 50&
> sleep 10
^C Ctrl + C
Le processus fils avec PID 4521 s'est terminé suite à un signal 2.
>
```

Etape 16:

ls > test.txt : liste le contenu du répertoire courant dans liste.txt

Etape 18:

```
> ls | wc -l
Le processus fils avec PID 4765 s'est terminé avec le code de sortie 0.
22
Le processus fils avec PID 4766 s'est terminé avec le code de sortie 0.
```

Le résultat coïncide avec celui du terminal.

Etape 19:

```
> cat minishell.c readcmd.c | grep int | wc -l
Le processus fils avec PID 4797 s'est terminé avec le code de sortie 0.
Le processus fils avec PID 4798 s'est terminé avec le code de sortie 0.
32
Le processus fils avec PID 4799 s'est terminé avec le code de sortie 0.
>
```

