

(mini) projet (mini) shell

Mdaa Saad

May 14, 2020

Question 2 - Question 3

```
saad@pop ~/D/S/m/fournitures> ./minishell0
minishell -> ls
minishell -> LisezMoi.html minishell minishell0.c readcmd.c test
LisezMoi.md minishell0 minishell.c readcmd.h
pwd
minishell -> /home/saad/Desktop/SX/miniprojet/fournitures
ls
minishell -> LisezMoi.html minishell minishell0.c readcmd.c test
LisezMoi.md minishell0 minishell.c readcmd.h
sleep(5)
minishell -> minishell -> █
```

On remarque que sans le *waitpid* le processus père n'attend pas la terminaison du fils et donc on obtient un chevauchement des valeurs de retour sur le *stdout* du père et son fils.

Question 4

Pour la commande interne *cd* on utilise la fonction *chdir* et pour la commande interne *exit* on utilise *exit(0)*

Question 6

Pout gérer les processus lancés depuis le minishell j'ai défini le module **process.c** qui permet de créer une liste chaînée de type *process* qui est un enregistrement d'attribut **name** , **etat** (A pour actif , S pour suspendu) , **id** (id propre au minishell) , **pid** pid du processus .

Question 10

Pour traiter les Pipelines j'ai défini la fonction **ncmd** qui retourne le nombre des commandes dans un pipeline , et l'exécution se fait dans une boucle **for** jusqu'au l'avant-dernière commande puis on traite la redirection pour la dernière commande ce qui permet aussi de traiter le cas d'une simple commande .

méthodologie de tests

- la compilation est automatisé il suffit de taper *make*
- Test de gestion des processus
- Test de redirection

```
/home/saad/Desktop/minishell $ cat process.c > f1
/home/saad/Desktop/minishell $ cat f1
#include "process.h"
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>

int lenght(process process){
    if (process == NULL){
        return 0;
    } else {
        return 1 + lenght(process -> next);
    }
}

int isin_proc(pid_t pid,process process){
    if (process == NULL){
        return 0;
    }
}
```

- Test de tube simple

```
}
}/home/saad/Desktop/minishell $ ls | wc -l
13
```

- Test de Pipelines

```
/home/saad/Desktop/minishell $ cat minishell.c | grep int | wc -l
31
/home/saad/Desktop/minishell $
```

- Test de la gestion des processus

- on lance deux processus en fond de tâche (sleep 500) et (sleep 600) et on teste la commande interne list

```
/home/saad/Desktop/minishell $ sleep 500 &
/home/saad/Desktop/minishell $ list

Name : sleep
Pid = 152015
Id = 0
etat = A
/home/saad/Desktop/minishell $
```

```
/home/saad/Desktop/minishell $ list

Name : sleep
Pid = 152015
Id = 0
etat = A

Name : sleep
Pid = 152171
Id = 1
etat = A
/home/saad/Desktop/minishell $
```

On teste le fonctionnement de la commande stop sur le processus id 0 et id 1

151883	151893	151893	151893	pts/0	152008	Ss	1000	0:00	bash
151893	152008	152008	151893	pts/0	152008	S+	1000	0:00	_ ./minishe
152008	152015	152008	151893	pts/0	152008	T+	1000	0:00	_ sleep
152008	152171	152008	151893	pts/0	152008	T+	1000	0:00	_ sleep

On teste le fonctionnement de la commande fg sur le processus id 1

```

151883 151893 151893 151893 pts/0      152008 Ss   1000   0:00 bash
151893 152008 152008 151893 pts/0      152008 S+   1000   0:00 \_ ./minishell
152008 152015 152008 151893 pts/0      152008 S+   1000   0:00 \_ sleep 500
152008 152171 152008 151893 pts/0      152008 T+   1000   0:00 \_ sleep 600

```

On teste le traitement du signal sigint

```

/home/saad/Desktop/minishell $ stop 0
/home/saad/Desktop/minishell $ stop 1
/home/saad/Desktop/minishell $ bg 0
/home/saad/Desktop/minishell $ fg 1
^C
killed by SIGINT
/home/saad/Desktop/minishell $ █

```