

Systèmes et Réseaux : Réalisations d'un mini shell

Grand Maxence, Muller Lucie

17/02/2017

Table des matières

| | | |
|----------|--|-----------|
| 1 | Partie 1 | 3 |
| 1.1 | Commandes simples | 3 |
| 1.2 | Commande avec redirections d'entrée ou de sortie | 5 |
| 1.3 | Séquence de commandes composée de deux commandes reliés par un tube | 7 |
| 1.4 | Séquence de commandes composée de plusieurs commandes et plusieurs tubes | 8 |
| 2 | Partie Bonus | 11 |
| 2.1 | Exécution de commandes en arrière plan | 11 |
| 2.2 | Changer l'état du processus au premier plan | 11 |
| 2.3 | Gestion des zombies | 12 |
| 2.4 | Commande intégrée jobs | 13 |
| 2.5 | Agir sur les commandes en arrière plan | 13 |

Introduction

Chapitre 1

Partie 1

1.1 Commandes simples

Dans cette partie nous devons implémenter une fonction permettant d'exécuter une commande simple en shell. Pour tester notre fonction nous avons tester plusieurs commande n'utilisant ni les pipes ni les redirections dans un shell et dans notre programme shell.

Code :

```
void commande_simple(struct cmdline *l){
    int pid = Fork();
    int status;
    if(pid == 0){
        execvp(l->seq[0][0], l->seq[0]);
        exit(0);
    }
    else{
        waitpid(pid, &status, 0);
    }
}
```

Test

SHELL

```
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ ls
csapp.c  Makefile      rapport.log  rapport.tex  readcmd.c  README.
md       shell.c      test      test3  tst      tube_simple.c
csapp.h  rapport.aux  rapport.pdf  rapport.toc  readcmd.h  shell
        sujet.pdf  test2      text      tst.c  waitpid1.c
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ ls -l
total 2688
-rwx----- 1 maxence maxence 20259 dÃ©c. 25 2014 csapp.c
-rwx----- 1 maxence maxence 6105 mars 16 2014 csapp.h
-rwx----- 1 maxence maxence 136 oct. 6 2009 Makefile
-rwx----- 1 maxence maxence 1499 fÃ©vr. 8 15:58 rapport.aux
-rwx----- 1 maxence maxence 30949 fÃ©vr. 8 15:58 rapport.log
```

```

-rwx----- 1 maxence maxence 128494 fÃ©vr. 8 15:58 rapport.pdf
-rwx----- 1 maxence maxence 2558 fÃ©vr. 8 16:02 rapport.tex
-rwx----- 1 maxence maxence 933 fÃ©vr. 8 15:58 rapport.toc
-rwx----- 1 maxence maxence 4699 fÃ©vr. 8 14:07 readcmd.c
-rwx----- 1 maxence maxence 1029 fÃ©vr. 8 14:02 readcmd.h
-rwx----- 1 maxence maxence 14 fÃ©vr. 7 18:02 README.md
-rwx----- 1 maxence maxence 41024 fÃ©vr. 8 15:45 shell
-rwx----- 1 maxence maxence 4338 fÃ©vr. 8 16:03 shell.c
-rwx----- 1 maxence maxence 94332 fÃ©vr. 7 12:01 sujet.pdf
-rwx----- 1 maxence maxence 150 fÃ©vr. 8 13:57 test
-rwx----- 1 maxence maxence 3 fÃ©vr. 8 13:58 test2
-rwx----- 1 maxence maxence 3 fÃ©vr. 7 15:51 test3
-rwx----- 1 maxence maxence 0 fÃ©vr. 7 15:36 text
-rwx----- 1 maxence maxence 18008 fÃ©vr. 8 12:08 tst
-rwx----- 1 maxence maxence 723 fÃ©vr. 7 13:55 tst.c
-rwx----- 1 maxence maxence 815 janv. 31 15:18 tube_simple.c
-rwx----- 1 maxence maxence 657 janv. 20 2009 waitpid1.c
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ cat test
19
pp.c
csapp.h
Makefile
readcmd.c
readcmd.h
README.md
shell
shell.c
shell.o
sujet.pdf
test
test2
test3
text
tst
tst.c
tst.o
tube_simple.c
waitpid1.c

```

shell.c

```

shell> ls
csapp.c  Makefile  rapport.log  rapport.tex  readcmd.c  README.
md  shell.c  test  test3  tst  tube_simple.c
csapp.h  rapport.aux  rapport.pdf  rapport.toc  readcmd.h  shell
sujet.pdf  test2  text  tst.c  waitpid1.c
shell> ls -l
total 2688
-rwx----- 1 maxence maxence 20259 dÃ©c. 25 2014 csapp.c
-rwx----- 1 maxence maxence 6105 mars 16 2014 csapp.h
-rwx----- 1 maxence maxence 136 oct. 6 2009 Makefile
-rwx----- 1 maxence maxence 1499 fÃ©vr. 8 15:58 rapport.aux
-rwx----- 1 maxence maxence 30949 fÃ©vr. 8 15:58 rapport.log
-rwx----- 1 maxence maxence 128494 fÃ©vr. 8 15:58 rapport.pdf
-rwx----- 1 maxence maxence 4558 fÃ©vr. 8 16:05 rapport.tex
-rwx----- 1 maxence maxence 933 fÃ©vr. 8 15:58 rapport.toc
-rwx----- 1 maxence maxence 4699 fÃ©vr. 8 14:07 readcmd.c

```

```

-rwx----- 1 maxence maxence 1029 fÃ©vr. 8 14:02 readcmd.h
-rwx----- 1 maxence maxence 14 fÃ©vr. 7 18:02 README.md
-rwx----- 1 maxence maxence 41024 fÃ©vr. 8 16:06 shell
-rwx----- 1 maxence maxence 4338 fÃ©vr. 8 16:03 shell.c
-rwx----- 1 maxence maxence 94332 fÃ©vr. 7 12:01 sujet.pdf
-rwx----- 1 maxence maxence 150 fÃ©vr. 8 13:57 test
-rwx----- 1 maxence maxence 3 fÃ©vr. 8 13:58 test2
-rwx----- 1 maxence maxence 3 fÃ©vr. 7 15:51 test3
-rwx----- 1 maxence maxence 0 fÃ©vr. 7 15:36 text
-rwx----- 1 maxence maxence 18008 fÃ©vr. 8 12:08 tst
-rwx----- 1 maxence maxence 723 fÃ©vr. 7 13:55 tst.c
-rwx----- 1 maxence maxence 815 janv. 31 15:18 tube_simple.c
-rwx----- 1 maxence maxence 657 janv. 20 2009 waitpid1.c
shell> cat test
19
pp.c
csapp.h
Makefile
readcmd.c
readcmd.h
README.md
shell
shell.c
shell.o
sujet.pdf
test
test2
test3
text
tst
tst.c
tst.o
tube_simple.c
waitpid1.c
shell> echo "test"
"test"
shell>

```

1.2 Commande avec redirections d'entrée ou de sortie

Dans cette section nous devons implémenter une fonction permettant de gérer les redirections d'entrée ou de sortie. Nos tests ont suivi les mêmes principes que l'étape précédente à l'exception que seuls les commandes avec pipes ne seront pas testées ici.

Code

```

void commande_redirection(struct cmdline *l){
    int fOut = 1; int fIn = 0;

    if(l->in != NULL)
        fIn = open(l->in, O_RDONLY, 0);
    if(l->out != NULL)

```

```

        fOut = open(l->out, O_WRONLY | O_CREAT, 0700) ;

int pid = Fork(); int status;
if(pid == 0){
    if(fOut != 1){
        close(1);
        dup2(fOut, 1);
    }
    if(fIn == 0){
        execvp(l->seq[0][0], l->seq[0]);
        exit(0);
    }else{
        close(0);
        dup2(fIn, 0);
        execvp(l->seq[0][0], l->seq[0]);
        exit(0);
    }
}

}
else{
    waitpid(pid, &status, 0);
}
}
}

```

Test
SHELL

```

maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ ls -l > test
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ cat test
total 2688
-rwx----- 1 maxence maxence 20259 dÃ©c. 25 2014 csapp.c
-rwx----- 1 maxence maxence 6105 mars 16 2014 csapp.h
-rwx----- 1 maxence maxence 136 oct. 6 2009 Makefile
-rwx----- 1 maxence maxence 1499 fÃ©vr. 8 16:10 rapport.aux
-rwx----- 1 maxence maxence 31866 fÃ©vr. 8 16:10 rapport.log
-rwx----- 1 maxence maxence 218712 fÃ©vr. 8 16:10 rapport.pdf
-rwx----- 1 maxence maxence 7408 fÃ©vr. 8 16:16 rapport.tex
-rwx----- 1 maxence maxence 933 fÃ©vr. 8 16:10 rapport.toc
-rwx----- 1 maxence maxence 4699 fÃ©vr. 8 14:07 readcmd.c
-rwx----- 1 maxence maxence 1029 fÃ©vr. 8 14:02 readcmd.h
-rwx----- 1 maxence maxence 14 fÃ©vr. 7 18:02 README.md
-rwx----- 1 maxence maxence 41024 fÃ©vr. 8 16:06 shell
-rwx----- 1 maxence maxence 4338 fÃ©vr. 8 16:03 shell.c
-rwx----- 1 maxence maxence 94332 fÃ©vr. 7 12:01 sujet.pdf
-rwx----- 1 maxence maxence 0 fÃ©vr. 8 16:16 test
-rwx----- 1 maxence maxence 3 fÃ©vr. 8 13:58 test2
-rwx----- 1 maxence maxence 3 fÃ©vr. 7 15:51 test3
-rwx----- 1 maxence maxence 0 fÃ©vr. 7 15:36 text
-rwx----- 1 maxence maxence 18008 fÃ©vr. 8 12:08 tst
-rwx----- 1 maxence maxence 723 fÃ©vr. 7 13:55 tst.c
-rwx----- 1 maxence maxence 815 janv. 31 15:18 tube_simple.c
-rwx----- 1 maxence maxence 657 janv. 20 2009 waitpid1.c
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ wc -l < test
23

```

```

maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ wc -c < test
1347
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ wc - < test > test2
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ cat test2
23 200 1347 -

```

```

shell> ls -l > test
shell> cat test
total 2816
-rwx----- 1 maxence maxence 20259 dÃ©c. 25 2014 csapp.c
-rwx----- 1 maxence maxence 6105 mars 16 2014 csapp.h
-rwx----- 1 maxence maxence 136 oct. 6 2009 Makefile
-rwx----- 1 maxence maxence 1499 fÃ©vr. 8 16:10 rapport.aux
-rwx----- 1 maxence maxence 31866 fÃ©vr. 8 16:10 rapport.log
-rwx----- 1 maxence maxence 218712 fÃ©vr. 8 16:10 rapport.pdf
-rwx----- 1 maxence maxence 9314 fÃ©vr. 8 16:22 rapport.tex
-rwx----- 1 maxence maxence 933 fÃ©vr. 8 16:10 rapport.toc
-rwx----- 1 maxence maxence 4699 fÃ©vr. 8 14:07 readcmd.c
-rwx----- 1 maxence maxence 1029 fÃ©vr. 8 14:02 readcmd.h
-rwx----- 1 maxence maxence 14 fÃ©vr. 7 18:02 README.md
-rwx----- 1 maxence maxence 41024 fÃ©vr. 8 16:23 shell
-rwx----- 1 maxence maxence 4343 fÃ©vr. 8 16:23 shell.c
-rwx----- 1 maxence maxence 94332 fÃ©vr. 7 12:01 sujet.pdf
-rwx----- 1 maxence maxence 1 fÃ©vr. 8 16:17 test
-rwx----- 1 maxence maxence 1 fÃ©vr. 8 16:17 test2
-rwx----- 1 maxence maxence 3 fÃ©vr. 7 15:51 test3
-rwx----- 1 maxence maxence 0 fÃ©vr. 7 15:36 text
-rwx----- 1 maxence maxence 18008 fÃ©vr. 8 12:08 tst
-rwx----- 1 maxence maxence 723 fÃ©vr. 7 13:55 tst.c
-rwx----- 1 maxence maxence 815 janv. 31 15:18 tube_simple.c
-rwx----- 1 maxence maxence 657 janv. 20 2009 waitpid1.c
shell> wc -l < test
23
shell> wc -c < test
1347
shell> wc - < test > test2
shell> cat test2
23 200 1347 -

```

1.3 Séquence de commandes composée de deux commandes reliés par un tube

```

void commande_pipe(struct cmdline *l){
    int tailleSeq; int tmp;
    for(tailleSeq=0; l->seq[tailleSeq+1]!=0; tailleSeq++);
    int pid = Fork(); int status; int i=0; int desc[2];
    if(pid == 0){
        pipe(desc);
        pid=Fork();
        if(pid != 0){

```



```
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ cat tmp2
-rwx----- 1 maxence maxence    136 oct.   6  2009 Makefile
-rwx----- 1 maxence maxence   1499 fÃ©vr.  8 16:10 rapport.aux
```

```
shell> head -5 < test | tail -2 | sort > tmp
shell> cat tmp
-rwx----- 1 maxence maxence    136 oct.   6  2009 Makefile
-rwx----- 1 maxence maxence   1499 fÃ©vr.  8 16:10 rapport.aux
```

Chapitre 2

Partie Bonus

2.1 Exécution de commandes en arrière plan

```
void commande_bg(struct cmdline *l){
    int pid, status;

    pid = Fork();

    if(pid == 0){
        execvp(l->seq[0][0], l->seq[0]);
        exit(0);
    }else{
        if(l->bg);
        else{
            waitpid(pid, &status, 0);
        }
    }
}
```

2.2 Changer l'état du processus au premier plan

```
void stop(int sig){
    printf("Fini\n");
    kill(child, SIGKILL);
}
void suspend(int sig){
    printf("Suspendu\n");
    kill(child, SIGSTOP);
}
void commande_signaux(struct cmdline *l){
    int pid = Fork(); int status;
```

```

Signal(SIGINT, stop);
Signal(SIGTSTP, suspend);

if(pid == 0){
    execvp(l->seq[0][0], l->seq[0]);
    exit(0);
}else{
    child = pid;
    Signal(SIGINT, stop);
    Signal(SIGTSTP, suspend);
    while (waitpid(pid, &status, 0) != pid);
}
}

```

```

maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ gedit
^C
[2]+  Fini                                okular
maxence@Sybil:/media/maxence/MyLinuxLive1/cours/l3_info/s06/SR/
projetSHELL$ gedit
^Z
[1]+  ArrÃªtÃ©                             gedit

```

```

shell> gedit
^CFini

```

2.3 Gestion des zombis

```

void zombi(int sig)
{
    pid_t pid;

    if ((pid = waitpid(-1, NULL, WNOHANG|WUNTRACED)) < 0)
        unix_error("waitpid error");
    printf("Handler reaped child %d\n", (int)pid);
    return;
}

void commande_zombi(struct cmdline *l){
    Signal( SIGCHLD, zombi);
    int pid = Fork();
    int status;
    if(pid == 0){
        execvp(l->seq[0][0], l->seq[0]);
        exit(0);
    }
}

```

2.4 Commande intégrée jobs

| |
|--|
| |
| |
| |

2.5 Agir sur les commandes en arrière plan

| |
|--|
| |
| |
| |