

# CSE-3033

## Operating Systems

### Project2

Süleyman Barış ESER - 150116055  
Murat Şenol – 150117039

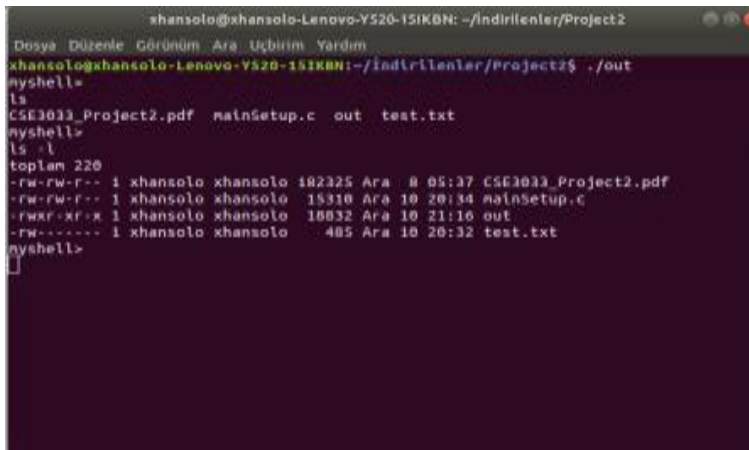
In this project, our aim is to create a User Interface Shell in Linux by using c.

#### Part A)

To execute a command with `execv(commandPath, args)` function, there is need to command exact path. To get this path, there are several different method but we used `popen('printenv PATH', 'r')` in the `commandPath(char **)` method. After getting command Path, fork a child and control status. If the status is true for `fork() == 0`, then we can notice code is in the child process. Then, `execv` function can be used and show output.

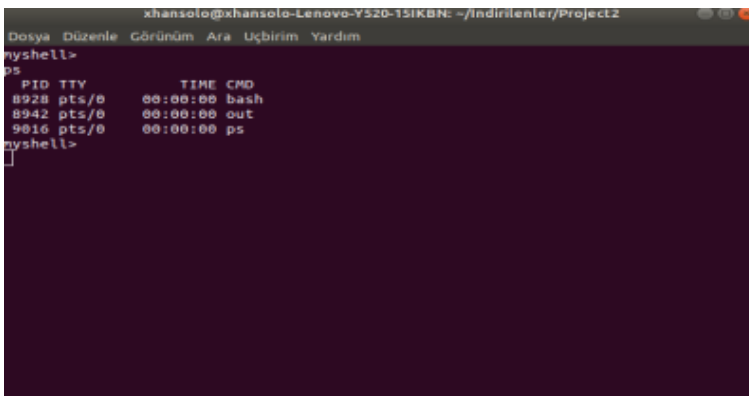
Examples;

**ls and ls -l:**



```
khansolo@khansolo-Lenovo-Y520-15IKBN: ~/Indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
khansolo@khansolo-Lenovo-Y520-15IKBN:~/Indirilenler/Project2$ ./out
myshell>
ls
CSE3033_Project2.pdf mainSetup.c out test.txt
myshell>
ls -l
toplam 220
-rw-rw-r-- 1 khansolo khansolo 102325 Ara  8 05:37 CSE3033_Project2.pdf
-rw-rw-r-- 1 khansolo khansolo  15310 Ara 10 20:34 mainSetup.c
-rwxr-xr-x 1 khansolo khansolo  10032 Ara 10 21:10 out
-rw----- 1 khansolo khansolo   405 Ara 10 20:32 test.txt
myshell>
```

**ps:**



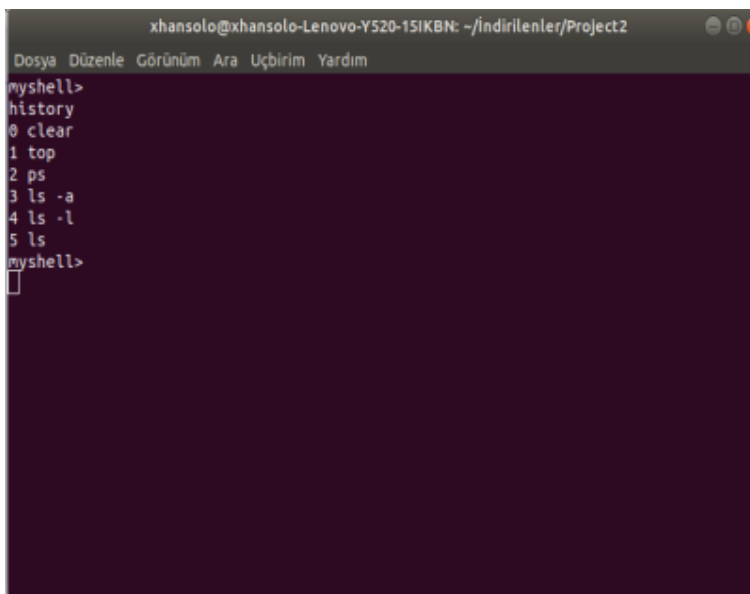
```
khansolo@khansolo-Lenovo-Y520-15IKBN: ~/Indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
myshell>
ps
  PID TTY          TIME CMD
 8928 pts/0    00:00:00 bash
 8942 pts/0    00:00:00 out
 9016 pts/0    00:00:00 ps
myshell>
```

and more command can be executed in this part, for example, `clear`, `htop`, `echo`, etc. And processes can run in background by putting `'&'` to the end of command.

## **Part B)**

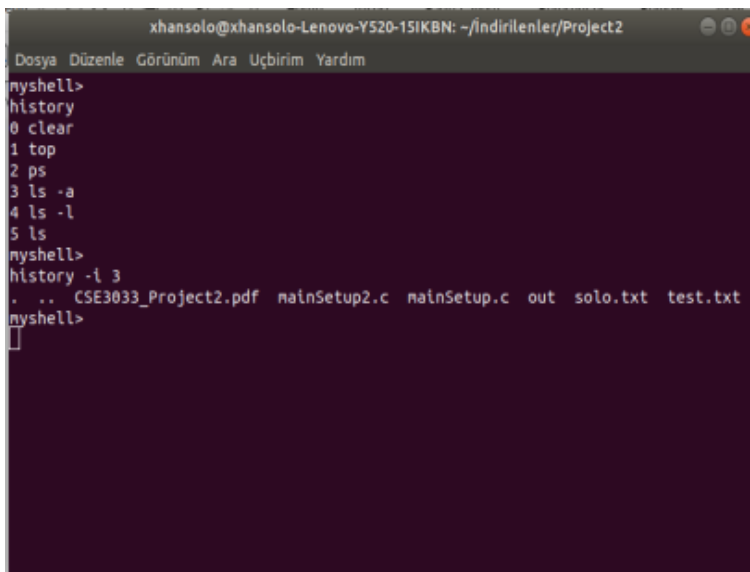
In this Part, to keep all old command in the code, there is a history array. At each process, the commands are adding to history array. If the user want to see the command history, he/she can print 'history' in the USS terminal. If the user want to re-execute the an old command, the user can print 'history -i num' , 'num' is the place of the the commandin the history array.

**history:**

A terminal window with a dark purple background. The title bar shows 'xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2'. The menu bar includes 'Dosya', 'Düzenle', 'Görünüm', 'Ara', 'Uçbirim', and 'Yardım'. The terminal content shows a 'myshell>' prompt, followed by the 'history' command. The output lists commands with indices: '0 clear', '1 top', '2 ps', '3 ls -a', '4 ls -l', and '5 ls'. The prompt returns to 'myshell>' with a cursor on the next line.

```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
myshell>
history
0 clear
1 top
2 ps
3 ls -a
4 ls -l
5 ls
myshell>
█
```

**history -i num:**

A terminal window with a dark purple background. The title bar shows 'xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2'. The menu bar includes 'Dosya', 'Düzenle', 'Görünüm', 'Ara', 'Uçbirim', and 'Yardım'. The terminal content shows a 'myshell>' prompt, followed by the 'history' command. The output lists commands with indices: '0 clear', '1 top', '2 ps', '3 ls -a', '4 ls -l', and '5 ls'. Then, the 'history -i 3' command is entered, which re-executes the command at index 3 ('ls -a'). The output shows the directory listing: '. .. CSE3033\_Project2.pdf mainSetup2.c mainSetup.c out solo.txt test.txt'. The prompt returns to 'myshell>' with a cursor on the next line.

```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
myshell>
history
0 clear
1 top
2 ps
3 ls -a
4 ls -l
5 ls
myshell>
history -i 3
. .. CSE3033_Project2.pdf mainSetup2.c mainSetup.c out solo.txt test.txt
myshell>
█
```

if the user press Control + Z then, code checks wheter there is still a foreground process or not by handling the SIGTSTP signal. If there is then stop it, else then, do nothing.

**^Z:**

```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/İndirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
myshell>
^Z
Ctrl-Z was pressed , no foreground procces found
Parçalama arızası (çekirdek döküldü)
xhansolo@xhansolo-Lenovo-Y520-15IKBN:~/İndirilenler/Project2$ ./out
myshell>
ls
CSE3033_Project2.pdf  mainSetup2.c  mainSetup.c  out  solo.txt  test.txt
myshell>
ls -l
toplam 248
-rw-rw-r-- 1 xhansolo xhansolo 182325 Ara  8 05:37 CSE3033_Project2.pdf
-rw-rw-r-- 1 xhansolo xhansolo 17916 Ara 10 22:17 mainSetup2.c
-rw-rw-r-- 1 xhansolo xhansolo 15304 Ara 10 22:08 mainSetup.c
-rwxr-xr-x 1 xhansolo xhansolo 22928 Ara 10 22:21 out
-rw----- 1 xhansolo xhansolo  55 Ara 10 21:35 solo.txt
-rw----- 1 xhansolo xhansolo  485 Ara 10 20:32 test.txt
myshell>
^Z
Ctrl-Z was pressed , no foreground procces found
Parçalama arızası (çekirdek döküldü)
xhansolo@xhansolo-Lenovo-Y520-15IKBN:~/İndirilenler/Project2$
```

But in this part, we can detect the the press and do the operations but, often the segmentation error occurs.

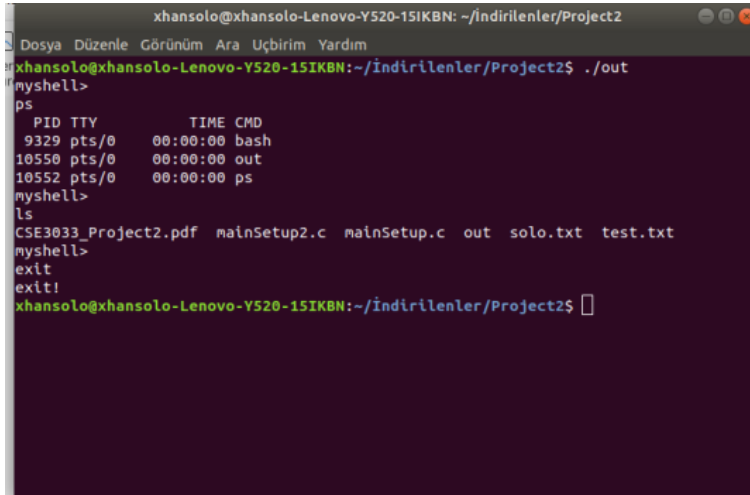
If the user print path, the executable files that added by the user shown in the terminal. The user can add or delete any path from the path array.

**path:**

```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/İndirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
path
myshell>
clear
myshell>
exit
exit!
xhansolo@xhansolo-Lenovo-Y520-15IKBN:~/İndirilenler/Project2$ ./out
myshell>
path + a
myshell>
path + b
myshell>
path
a:b:
myshell>
path + c
myshell>
path - a
myshell>
path
b:c:
myshell>
```

If the user print exit, then the exitShell function by using waitpid(-1, NULL, WNOHANG) controls the number of child processes that run in background. The Function stops them and exit from code.

**exit:**



```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
xhansolo@xhansolo-Lenovo-Y520-15IKBN:~/indirilenler/Project2$ ./out
myshell>
ps
  PID TTY          TIME CMD
 9329 pts/0    00:00:00 bash
10550 pts/0    00:00:00 out
10552 pts/0    00:00:00 ps
myshell>
ls
CSE3033_Project2.pdf mainSetup2.c mainSetup.c out solo.txt test.txt
myshell>
exit
exit!
xhansolo@xhansolo-Lenovo-Y520-15IKBN:~/indirilenler/Project2$
```

### Part C)

If the user print redirection command, then in the execShell function controls the redirection type. Then, the function create a pipe in and pipe out to connect the out file or in file with the program. Then, the function create a dup2() and write or read the file.

**command > a.txt:**



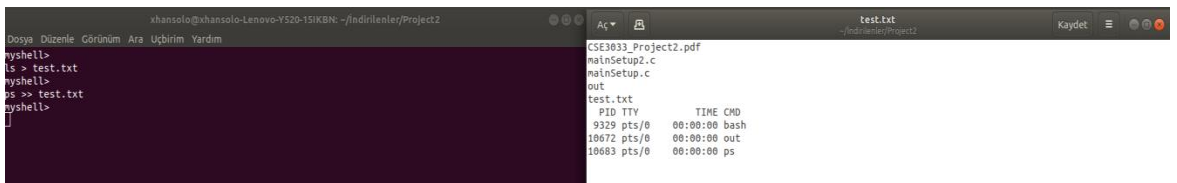
```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
ls -l > test.txt
myshell>

```

```
test.txt
~/indirilenler/Project2
Kaydet

toplamlar 240
-rw-rw-r-- 1 xhansolo xhansolo 182325 Ara  8 05:37 CSE3033_Project2.pdf
-rw-rw-r-- 1 xhansolo xhansolo 17916 Ara 10 22:17 mainSetup2.c
-rw-rw-r-- 1 xhansolo xhansolo 15304 Ara 10 22:08 mainSetup.c
-rwxr-xr-x 1 xhansolo xhansolo 22928 Ara 10 22:21 out
-rw-rw-r-- 1 xhansolo xhansolo 0 Ara 10 22:35 test.txt
```

**command >> a.txt:**



```
xhansolo@xhansolo-Lenovo-Y520-15IKBN: ~/indirilenler/Project2
Dosya Düzenle Görünüm Ara Uçbirim Yardım
myshell>
ls > test.txt
myshell>
ps >> test.txt
myshell>

```

```
test.txt
~/indirilenler/Project2
Kaydet

CSE3033_Project2.pdf
mainSetup2.c
mainSetup.c
out
test.txt
  PID TTY          TIME CMD
 9329 pts/0    00:00:00 bash
10672 pts/0    00:00:00 out
10683 pts/0    00:00:00 ps
```

And, can do 'program < file-in > file-out', 'program < file-in' and 'program 2> file-out'.

### Part Bonus -)

Split the args from ‘;’, and adding them to an array. Then, execute them one by one.

**command 1 ; command 2:**

```
myshell>
ls -l ; ps -l
toplam 244
-rw-rw-r-- 1 xhansolo xhansolo 182325 Ara  8 05:37 CSE3033_Project2.pdf
-rw-rw-r-- 1 xhansolo xhansolo 17916 Ara 10 22:17 mainSetup2.c
-rw-rw-r-- 1 xhansolo xhansolo 15304 Ara 10 22:08 mainSetup.c
-rwxr-xr-x 1 xhansolo xhansolo 22928 Ara 10 22:21 out
-rw----- 1 xhansolo xhansolo 171 Ara 10 22:36 test.txt
F S  UID  PID  PPID  C  PRI  NI  ADDR  SZ  WCHAN  TTY  TIME CMD
0 S  1000  9329  9319  0  80   0 -  7797 wait  pts/0  00:00:00 bash
0 S  1000  10672  9329  0  80   0 -  1129 wait  pts/0  00:00:00 out
4 R  1000  10718  10672  0  80   0 -  9360 -    pts/0  00:00:00 ps
```

\*\*\*But in this part, when we use a for loop, the segmentation error occurs and code exits abnormally. So, code execute them one by one. It works for just 2 command.