Marmara University Faculty of Engineering
Computer Engineering

CSE3033- OPERATING SYSTEMS	

STUDENT ID-NAME-SURNAME

150116841 HALE ŞAHİN
150114856 ÖZGE YILDIRIM

INTRODUCTION

Written a simple shell in C. Under Linux, quick and practical programming is the shortest way to program shell. Operating the system's natural commands and no additional interpreters to request the shell script (script) language is the most important factor separating from others. You can use a code that you write without making changes to all Linux / UNIX systems. Shell under Linux programming can be done with many different shell.

EXPLANATION OF PART A

Made a simple shell and it takes the command and executes in a new process. Created a new process using fork system call when the program takes the program name. Executed the program by the new process. The child use the **execl()** function for executing a new program. There is a foreground and background processes. When the process is executed in foreground, the shell waits for the task for completing. Then prompts the user for another command.When the process is executed in background, the shell doesn't wait for the task for completing. Then prompts the user for another command.

TEST SCRIPTS AND SOME CONTROLS OF PART A

Shell runs the command pwd and works properly.

```
myshell: cp mine.c test.txt
myshell: ls
myshell:  CSE3033_Project2.pdf    processes              'redirect&pipes'   threads
 mainSetup.c              ProcessesInUnix.pdf     signals
 mine.c                   Project2.c              Signals.pdf
 my.o                    'Redirection & Pipes.pdf'  test.txt
```

Ls command list the files properly.

```
myshell: ls -l
toplam 1460
-rw-rw-r--. 1 halesahin halesahin 283678 Kas 13 13:48  CSE3033_Project2.pdf
-rwxr--r--. 1 halesahin halesahin   4322 Ara  8 16:07  mainSetup.c
-rw-r--r--. 1 halesahin halesahin   6283 Ara  8 12:54  mine.c
-rwxrwxr-x. 1 halesahin halesahin   8928 Ara  8 16:00  my.o
drwxrwxr-x. 2 halesahin halesahin   4096 Ara  1 23:29  processes
-rw-rw-r--. 1 halesahin halesahin 239965 Ara  1 16:59  ProcessesInUnix.pdf
-rw-rw-r--. 1 halesahin halesahin   7073 Ara  1 16:30  Project2.c
-rw-rw-r--. 1 halesahin halesahin 802364 Ara  1 16:59 'Redirection & Pipes.pdf'
drwxrwxr-x. 2 halesahin halesahin   4096 Eki 30 15:22 'redirect&pipes'
drwxrwxr-x. 2 halesahin halesahin   4096 Kas 13 15:03  signals
-rw-rw-r--. 1 halesahin halesahin 103176 Ara  1 16:58  Signals.pdf
drwxr-xr-x. 2 halesahin halesahin   4096 Ara  8 16:08  test.txt
drwxrwxr-x. 2 halesahin halesahin   4096 Kas 22 15:24  threads
myshell:
```
Here we see ls -l is working in our shell code too.

```
myshell: rm mine.c
myshell: ls
 CSE3033_Project2.pdf    processes              'Redirection & Pipes.pdf'    Signals.pdf
 mainSetup.c             ProcessesInUnix.pdf    'redirect&pipes'             test.txt
 my.o                    Project2.c              signals                     threads
myshell:
```

Rm file function is also working. When we list the new version, we see that the file removed.

EXPLANATIOIN OF PART B

Made a simple shell and it takes the command as input and executes that in a new process. Fork a child process using fork() and the child executes the program.

TEST SCRIPTS AND SOME CONTROLS OF PART B

Ctrl Z : Stops the foregound processes when signal is on, even if there is child pids.

```
myshell: ls -l
toplam 1456
-rw-rw-r--. 1 halesahin halesahin 283678 Kas 13 13:48  CSE3033_Project2.pdf
-rwxr--r--. 1 halesahin halesahin   4552 Ara  8 17:02  mainSetup.c
-rwxrwxr-x. 1 halesahin halesahin  13024 Ara  8 16:20  my.o
drwxrwxr-x. 2 halesahin halesahin   4096 Ara  1 23:29  processes
-rw-rw-r--. 1 halesahin halesahin 239965 Ara  1 16:59  ProcessesInUnix.pdf
-rw-rw-r--. 1 halesahin halesahin   7073 Ara  1 16:30  Project2.c
-rw-rw-r--. 1 halesahin halesahin 802364 Ara  1 16:59 'Redirection & Pipes.pdf'
drwxrwxr-x. 2 halesahin halesahin   4096 Eki 30 15:22 'redirect&pipes'
drwxrwxr-x. 2 halesahin halesahin   4096 Kas 13 15:03  signals
-rw-rw-r--. 1 halesahin halesahin 103176 Ara  1 16:58  Signals.pdf
drwxr-xr-x. 2 halesahin halesahin   4096 Ara  8 16:08  test.txt
drwxrwxr-x. 2 halesahin halesahin   4096 Kas 22 15:24  threads
myshell: ^Z
[2]+  Durdu                      ./my.o
[halesahin@localhost İndirilenler]$
```

Clr: syste clear function is triggerred, and clears the terminal screen.

```
[halesahin@localhost İndirilenler]$ ./my.o
myshell: clr
```

Here we see it cleared.

```
Dosya  Düzenle  Görünüm  Ara  U
myshell:
```

Exit function is prompt that user wants to exit.

```
myshell: pwd
/home/halesahin/İndirilenler
myshell: exit
[halesahin@localhost İndirilenler]$
```

EXPLANATIOIN OF PART C

Made a simple shell that supports I/O redirection and it includes arguments. (It can write or append the standard error or output of x.c file to x.out file.)

TEST SCRIPTS AND SOME CONTROLS OF PART C

1.   First query is list the files in the current directory, write them into file.out, even that file is not exist, create one. If exist overwrite it.

```
[halesahin@localhost Indirilenler]$ ./my.o
myshell: ls > file.out
myshell: █
```

Result of file.out

```
Aç ▾   ▣                                          file.out
                                                  ~/Indirilenler
CSE3033_Project2.pdf
file.in
file.out
mainSetup.c
my.o
processes
ProcessesInUnix.pdf
Project2.c
Redirection & Pipes.pdf
redirect&pipes
signals
Signals.pdf
test.txt
threads
```

2. Second query appends te command execution

```
myshell: ls -al >> file.out
myshell: █
```

Aç ▾ 🗗

```
CSE3033_Project2.pdf
file.in
file.out
mainSetup.c
my.o
processes
ProcessesInUnix.pdf
Project2.c
Redirection & Pipes.pdf
redirect&pipes
signals
Signals.pdf
test.txt
threads
toplam 1472
drwxr-xr-x.  7 halesahin halesahin   4096 Ara  8 19:53 .
drwx------. 16 halesahin halesahin   4096 Ara  8 16:06 ..
-rw-rw-r--.  1 halesahin halesahin 283678 Kas 13 13:48 CSE3033_Project2.pdf
-rw-r--r--.  1 halesahin halesahin    166 Ara  8 19:45 file.in
-rw-rw-r--.  1 halesahin halesahin    172 Ara  8 19:53 file.out
-rwxr--r--.  1 halesahin halesahin   6354 Ara  8 19:41 mainSetup.c
-rwxrwxr-x.  1 halesahin halesahin  13280 Ara  8 19:47 my.o
drwxrwxr-x.  2 halesahin halesahin   4096 Ara  1 23:29 processes
-rw-rw-r--.  1 halesahin halesahin 239965 Ara  1 16:59 ProcessesInUnix.pdf
-rw-rw-r--.  1 halesahin halesahin   7073 Ara  1 16:30 Project2.c
-rw-rw-r--.  1 halesahin halesahin 802364 Ara  1 16:59 Redirection & Pipes.pdf
drwxrwxr-x.  2 halesahin halesahin   4096 Eki 30 15:22 redirect&pipes
drwxrwxr-x.  2 halesahin halesahin   4096 Kas 13 15:03 signals
-rw-rw-r--.  1 halesahin halesahin 103176 Ara  1 16:58 Signals.pdf
drwxr-xr-x.  2 halesahin halesahin   4096 Ara  8 16:08 test.txt
drwxrwxr-x.  2 halesahin halesahin   4096 Kas 22 15:24 threads
```

3. Third is taking input file and do the command.

Aç ▾ 🗗

| mainSetup.c | × | *mainSetup.c |

```
CSE3033_Project2.pdf
file.out
mainSetup.c
my.o
processes
ProcessesInUnix.pdf
Project2.c
Redirection & Pipes.pdf
redirect&pipes
signals
Signals.pdf
test.txt
threads
```

```
myshell: ls < file.in
< redirection
ls: 'Redirection''e erişilemedi: No such file or directory
ls: '&''e erişilemedi: No such file or directory
ls: 'Pipes.pdf''e erişilemedi: No such file or directory
 CSE3033_Project2.pdf   mainSetup.c    ProcessesInUnix.pdf    Signals.pdf
 file.out               my.o           Project2.c

processes:
badprocessID.c  execcmd.o  fanwait.c  parentwaitpid.c  simplechain.o  simplefan.out
badprocessID.o  execls.c   fanwait.o  parentwaitpid.o  simplefan.c    twoprocs.c
execcmd.c       execls.o   fnwait.o   simplechain.c    simplefan.o    twoprocs.o

'redirect&pipes':
redirect.c  simpleredirect.c

signals:
signalterminate.c

test.txt:
mine.c

threads:
hello_arg2.c  hello.c  join.c  thread_create2.c  thread_create.c  thread_join.c
myshell: 
```

It may append to existing file.

4. Writes error into the file

```
myshell: cmd 2> file.out
myshell: 
```

Aç ▾   🗗   file.out
                                                             ~/İndirilenler
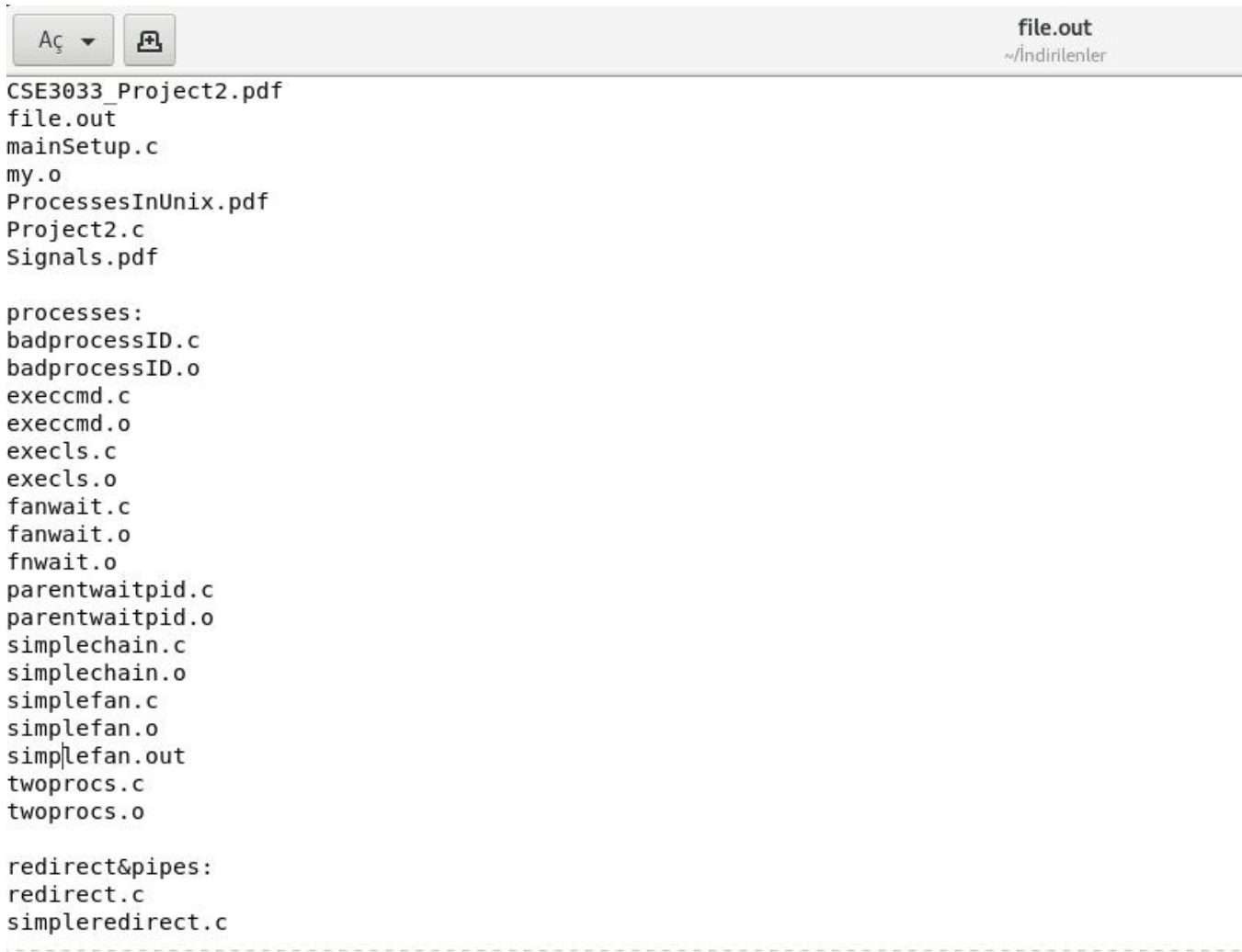
/bin/sh: cmd: komut yok

5.

Takes input and apply command then writes result to the output.

```
myshell: ls < file.in > file.out
ls: ls: 'Redirection''e erişilemedi: No such file or directory
ls: '&''e erişilemedi: No such file or directory
ls: 'Pipes.pdf''e erişilemedi: No such file or directory
'Redirection''e erişilemedi: No such file or directory
ls: '&''e erişilemedi: No such file or directory
ls: 'Pipes.pdf''e erişilemedi: No such file or directory
[halesahin@localhost İndirilenler]$ 
```

```
Aç      🔼

CSE3033_Project2.pdf
file.out
mainSetup.c
my.o
ProcessesInUnix.pdf
Project2.c
Signals.pdf

processes:
badprocessID.c
badprocessID.o
execcmd.c
execcmd.o
execls.c
execls.o
fanwait.c
fanwait.o
fnwait.o
parentwaitpid.c
parentwaitpid.o
simplechain.c
simplechain.o
simplefan.c
simplefan.o
simplefan.out
twoprocs.c
twoprocs.o

redirect&pipes:
redirect.c
simpleredirect.c
```

Pipeline

Pipeline is executing first command then takes the result to execute second command.

```
[halesahin@localhost İndirilenler]$ ./my.o
myshell: ls | sort
 CSE3033_Project2.pdf    mainSetup.c    ProcessesInUnix.pdf        'redirect&pipes'    test.txt
 file.in                 my.o           Project2.c                 signals             threads
 file.out                processes      'Redirection & Pipes.pdf'  Signals.pdf
[halesahin@localhost İndirilenler]$
```

```
[halesahin@localhost İndirilenler]$ ./my.o
myshell: ls -al | sort
 CSE3033_Project2.pdf    mainSetup.c    ProcessesInUnix.pdf          'redirect&pipes'    test.txt
 file.in                 my.o          Project2.c                    signals             threads
 file.out                processes     'Redirection & Pipes.pdf'     Signals.pdf
[halesahin@localhost İndirilenler]$ ./my.o
myshell: ls -l | sort
 CSE3033_Project2.pdf    mainSetup.c    ProcessesInUnix.pdf          'redirect&pipes'    test.txt
 file.in                 my.o          Project2.c                    signals             threads
 file.out                processes     'Redirection & Pipes.pdf'     Signals.pdf
[halesahin@localhost İndirilenler]$
```