

PROJECT 2 REPORT

150117042	150117061	150117510
Ezgi Doğruer	İsra Nur Alperen	Elif Gökpınar

STEP A:

- background :

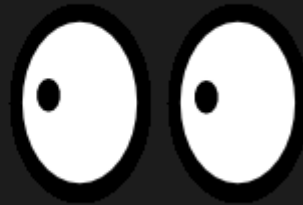
```
myshell: xeyes &
args 0 = xeyes
args 1 = &
args 2 = (null)
[1] Pid: [10976] Command: xeyes
myshell: ls &
args 0 = ls
args 1 = &
args 2 = (null)
[2] Pid: [10977] Command: ls process started..
a          exo-terminal-emulator.desktop kesin_son.o opera.desktop son.c
CSE3033_Project2.zip kesin_son.c Lab Project_1 son.o
myshell: ps_all
args 0 = ps_all
args 1 = (null)
Running:
[1] Pid: [10976] Command: xeyes
Finished:
[2] Pid: [10977] Command: ls
myshell: 
```



```
myshell: pwd &
args 0 = pwd
args 1 = &
args 2 = (null)
[3] Pid: [10978] Command: pwd process started..
/home/ezgidogruer/Masaüstü
myshell: ps_all
args 0 = ps_all
args 1 = (null)
Running:
Finished:
[1] Pid: [10976] Command: xeyes
[3] Pid: [10978] Command: pwd
myshell:
```

- **foreground** : Runs when a command is run as foreground and does not output such as "Running" or "Finished" when executed in "ps_all" command.

```
myshell: xeyes
args 0 = xeyes
args 1 = (null)
pwd
ls
█
```



After the xeyes is closed , other commands execute :

```
myshell: xeyes
args 0 = xeyes
args 1 = (null)
pwd
ls
myshell: args 0 = pwd
args 1 = (null)
/home/ezgidogruer/Masaüstü
myshell: args 0 = ls
args 1 = (null)
a          exo-terminal-emulator.desktop  kesin_son.o  opera.desktop  son.c
CSE3033_Project2.zip  kesin_son.c  Lab          Project_1      son.o
myshell:
```

STEP B:

- **ps_all**

- ❖ Shows the process situation.
- ❖ If the background process is finished, ps_all refresh the list.

Ps_all shows the added bacground process.

```
myshell: xeyes &
[1] Pid: [41072] Command: xeyes
myshell: firefox &
[2] Pid: [41074] Command: firefox
myshell: ps_all
Running:
[1] Pid: [41072] Command: xeyes
[2] Pid: [41074] Command: firefox
Finished:
myshell: ^C
```

If the process is finished , ps_all shows this process in the finished part.

```
elif@elif-SATELLITE-L830:~/Desktop$ ./test
myshell: xeyes &
[1] Pid: [41423] Command: xeyes
myshell: xeyes &
[2] Pid: [41424] Command: xeyes
myshell: ps_all
Running:
[1] Pid: [41423] Command: xeyes
[2] Pid: [41424] Command: xeyes
Finished:
myshell: █
```



```
elif@elif-SATELLITE-L830:~/Desktop$ ./test
myshell: xeyes &
[1] Pid: [41423] Command: xeyes
myshell: xeyes &
[2] Pid: [41424] Command: xeyes
myshell: ps_all
Running:
[1] Pid: [41423] Command: xeyes
[2] Pid: [41424] Command: xeyes
Finished:
myshell: ps_all

*****Running:
[2] Pid: [41424] Command: xeyes
Finished:
[1] Pid: [41423] Command: xeyes
myshell: █
```



- **bookmark :**

At this stage, we created a linkedlist named Bookmark to keep the commands entered. First, we check if the bookmark command is entered as args [0] in the main method. Then we call the bookmarkCommand function. We make checks on the commands entered with this method. If only the transaction command is entered without entering the -l, -d or -i special command, the first thing to do is add these commands to our linkedlist with the addingBookmark method.

If the entered comment is -l, we print all commands in our linked list as a list on the terminal. Also, when the index of the command in the list enters with the -d command, we delete the element in that index from the linkedlist where we keep the commands, and move the next element to the index where we deleted it. For this process, we send the index of the element to wanted be deleted to the deleteBookmark method, and it happens in this method.

In addition to these, we run that command when the index of the element to be processed is entered with the -i command. We put the command we want to run for this process to an array and run this command with the function execvp ().

```
myshell: bookmark "ps -a"
args 0 = bookmark
args 1 = "ps
args 2 = -a"
args 3 = (null)
myshell: bookmark "ls -l | wc -l"
args 0 = bookmark
args 1 = "ls
args 2 = -l
args 3 = |
args 4 = wc
args 5 = -l"
args 6 = (null)
myshell: bookmark "ls -l"
args 0 = bookmark
args 1 = "ls
args 2 = -l"
args 3 = (null)
```

```
myshell: bookmark -l
args 0 = bookmark
args 1 = -l
args 2 = (null)

0 "ps -a"

1 "ls -l | wc -l"

2 "ls -l"
myshell: bookmark -i 0
args 0 = bookmark
args 1 = -i
args 2 = 0
args 3 = (null)
      PID TTY          TIME CMD
  10561 pts/0        00:00:00 son.o
  10562 pts/0        00:00:00 ps
myshell: █
```

```
myshell: bookmark -d 1
args 0 = bookmark
args 1 = -d
args 2 = 1
args 3 = (null)
myshell: bookmark -l
args 0 = bookmark
args 1 = -l
args 2 = (null)

0 "ps -a"

1 "ls -l"
myshell: █
```

- **search :**

First part : User can command the search and word, It tries to find the word in current subdirectory.

Second part : If user command `-r`, the word is searched every subdirectories recursively.

search "main"

```
myshell: search "main"
3: ./apple.c => main
1: ./d.C => main();
825: ./asa.c => int main(void)
104: ./s.C => int main(int argc, char *argv[])
174: ./son.c => //printf("The main process is not wait child anymore.\n");
839: ./son.c => int main(void)
819: ./e.c => int main(void)
69: ./learn.c => int main(){
```

search -r "main"

```
myshell: search -r "main"
3: ./apple.c => main
1: ./d.C => main();
81: ./CSE3033_Project2/mainSetup.c => int main(void)
825: ./asa.c => int main(void)
176: ./mainSetup.c => //printf("The main process is not wait child anymore.\n");
841: ./mainSetup.c => int main(void)
104: ./s.C => int main(int argc, char *argv[])
61: ./son.c => //printf("The main process is not wait child anymore.\n");
805: ./son.c => int main(void)
819: ./e.c => int main(void)
300: ./Oncekiler/errorCheck/Subsub.c => int main(void)
300: ./Oncekiler/asa.c => int main(void)
104: ./Oncekiler/sub.h => int main(int argc, char *argv[])
104: ./Eskiler/s.h => int main(int argc, char *argv[])
164: ./Eskiler/s.c => int main(int argc, char *argv[])
445: ./Eskiler/main.c => int main(void) {
69: ./learn.c => int main(){
174: ./gecebitti/gece.c => //printf("The main process is not wait child anymore.\n");
839: ./gecebitti/gece.c => int main(void)
myshell: 
```

Error check :

```
myshell: search main
You should use quote each two side of the word!Try again.
myshell: search -r main
You should use quote each two side of the word!Try again.
```

Error check :

```
myshell: search -r main
You should use quote each two side of the word!Try again.
myshell: search -s main

Wrong input, try again.
myshell: 
```

- ^Z

We have 2 methods named `cathCtrlZ ()` and `checkCtrlZ ()` for ^Z operation. First we define a sigaction named `ctrlZ`. Then we do synchronization with the values in this sigaction struct. When pressed with the `SIGTSTP` statement in the if control, we check the system to stop and shut down and perform this operation. Also, we stop the process that has the pid we got with `cathCtrlz` function. We call these functions in the `run` method where we run foreground operations.

- **exit**

If there are no background process, system will exit.

```
elif@elif-SATELLITE-L830:~/Desktop$ ./test
myshell: ls
apple.c  CSE3033_Project2  e.c      file.c  Öncekiler  s.C
asa.c    d.C                  Eskiler  learn.c  project1   son.c
myshell: exit
elif@elif-SATELLITE-L830:~/Desktop$
```

If there are background process, system will not exit.

```
elif@elif-SATELLITE-L830:~/Desktop$ ./test
myshell: ls
apple.c  CSE3033_Project2  e.c      file.c  Öncekiler  s.C
asa.c    d.C                  Eskiler  learn.c  project1   son.c
myshell: xeyes &
[1] Pid: [40317] Command: xeyes
myshell: exit
There are still background process, you can not exit!
myshell:
```