

# LAB 1

## SIMPLE SHELL

<b>Name</b>	Ranime Ahmed Elsayed Shehata.
<b>ID</b>	21010531.

## Simple Shell (Multi-Processing)

- **Objectives:**

1. Familiarity with system calls in Unix environment.
  2. Introduction to processes and multi-processing.
  3. Introduction to signal handling in Unix environment.
- 

- **Problem Statement:**

It is required to implement a Unix shell program. A shell is simply a program that conveniently allows you to run other programs. Your shell must support the following commands:

1. The internal shell command "exit" which terminates the shell
2. A command with no arguments

Example: ls, cp, rm

3. A command with arguments

Example: ls -l

4. A command, with or without arguments, executed in the background using &.

Example: gedit &

Details: In this case, your shell must execute the command and return immediately, not blocking until the command finishes.

- ✦ Requirements: You have to show that the opened process will be nested as a child process to the shell program via opening the task manager found in the operating system. Additionally, you have to write in a log file (basic text file) when a child process is terminated (main application will be interrupted by a SIGCHLD signal). So you have to implement an interrupt handler to handle this interrupt and do the corresponding action to it.

5. Shell builtin commands

Commands: cd & echo

✦ cd: Cover all the following cases (assume no spaces in path):

- \* cd

- \* cd ~

- \* cd ..

- \* cd absolute\_path

- \* cd relative\_path\_to\_current\_working\_directory

✦ echo: Prints the input after evaluating all expressions (assume input to echo must be within double quotations).

## 6. Expression evaluation

Commands: export

Details: Set values to variables and print variables values. No mathematical operations is needed.

Export Details: Accept input of two forms, either a string without spaces, or a full string inside double quotations.

---

- **Functions:**

- 1) int main(): It sets up a signal handler for child process termination, call the setup function and starts the main shell loop.
- 2) Void setup\_environment(): This function sets up the shell environment by getting the current working directory, printing a prompt, and reading user input. It then removes the newline character and calls "parse\_input" function to process the input.
- 3) Void parse\_input(): This function parses the user input, identifies built-in commands (echo, cd, export), and then calls the appropriate function to execute the command. It also executes the command "exit" which terminates the shell.
- 4) Void shell(): This function runs the main shell loop, repeatedly calling "setup\_environment()".
- 5) Void execute\_shell\_builtin(): This function handles the execution of built-in shell commands (cd, echo, export) using a switch case based on the input.

- 6) Void `execute_command()`: This function is responsible for executing external non-builtin commands (`mkdir`, `ls`, `rm`, `pwd`, ..) by forking a child process and calling “`execvp`” in the child.
  - 7) void `on_child_exit(pid_t pid)`: This function handles the termination of a child process, waiting for it and writing information to a log file.
  - 8) void `on_signal_exit()`: This function is a signal handler for child process termination. It calls “`write_to_log_file`” function to log the termination.
  - 9) void `write_to_log_file(pid_t pid)`: This function writes information about the terminated child process to a log file.
  - 10) void `remove_quotes(char *str)`: This function removes quotes from a string.
- 

- **Sample Runs:**

- Compile and execute the c file.

```
ranime@Nemo: ~/Desktop/Term6/OS/Labs/Lab1
ranime@Nemo:~/Desktop/Term6/OS/Labs/Lab1$ gcc -o exe main.c
ranime@Nemo:~/Desktop/Term6/OS/Labs/Lab1$ ./exe
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>S
```

- `ls`, `mkdir` commands.

```
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls
cmake-build-debug      exe          logs.txt      main.c
'demo test case.webm'  'log file.png'  main        'System monitor.png'
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>mkdir test
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls
cmake-build-debug      exe          logs.txt      main.c      test
'demo test case.webm'  'log file.png'  main        'System monitor.png'
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>
```

➤ ls -a, -l, -h

```
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls -a -l -h
total 8.3M
drwxrwxr-x 5 ranime ranime 4.0K Mar  5 17:23 .
drwxrwxr-x 4 ranime ranime 4.0K Mar  5 15:53 ..
drwxrwxr-x 3 ranime ranime 4.0K Mar  5 15:38 cmake-build-debug
-rw-rw-r-- 1 ranime ranime 8.0M Mar  5 17:07 'demo test case.webm'
-rwxrwxr-x 1 ranime ranime 18K Mar  5 17:21 exe
drwxrwxr-x 2 ranime ranime 4.0K Mar  5 17:20 .idea
-rw-rw-r-- 1 ranime ranime 46K Mar  5 17:08 'log file.png'
-rw-rw-r-- 1 ranime ranime 593 Mar  5 17:23 logs.txt
-rwxrwxr-x 1 ranime ranime 18K Mar  5 17:22 main
-rw-rw-r-- 1 ranime ranime 6.4K Mar  5 16:57 main.c
-rw-rw-r-- 1 ranime ranime 186K Mar  5 17:11 'System monitor.png'
drwxrwxr-x 2 ranime ranime 4.0K Mar  5 17:23 test
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>
```

➤ export & ls

```
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>export x="-a -l -h"
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls $x
total 8.3M
drwxrwxr-x 5 ranime ranime 4.0K Mar  5 17:23 .
drwxrwxr-x 4 ranime ranime 4.0K Mar  5 15:53 ..
drwxrwxr-x 3 ranime ranime 4.0K Mar  5 15:38 cmake-build-debug
-rw-rw-r-- 1 ranime ranime 8.0M Mar  5 17:07 'demo test case.webm'
-rwxrwxr-x 1 ranime ranime 18K Mar  5 17:21 exe
drwxrwxr-x 2 ranime ranime 4.0K Mar  5 17:20 .idea
-rw-rw-r-- 1 ranime ranime 46K Mar  5 17:08 'log file.png'
-rw-rw-r-- 1 ranime ranime 626 Mar  5 17:23 logs.txt
-rwxrwxr-x 1 ranime ranime 18K Mar  5 17:22 main
-rw-rw-r-- 1 ranime ranime 6.4K Mar  5 16:57 main.c
-rw-rw-r-- 1 ranime ranime 186K Mar  5 17:11 'System monitor.png'
drwxrwxr-x 2 ranime ranime 4.0K Mar  5 17:23 test
```



- pwd & cd commands.

```
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>pwd
/home/ranime/Desktop/Term6/OS/Labs/Lab1
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>cd ..
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs >>pwd
/home/ranime/Desktop/Term6/OS/Labs
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs >>cd ..
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS >>pwd
/home/ranime/Desktop/Term6/OS
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS >>cd
SIMPLE-SHELL:/home/ranime >>cd /home/ranime/Desktop/Term6
SIMPLE-SHELL:/home/ranime/Desktop/Term6 >>pwd
/home/ranime/Desktop/Term6
SIMPLE-SHELL:/home/ranime/Desktop/Term6 >>cd ~
SIMPLE-SHELL:/home/ranime >>pwd
/home/ranime
SIMPLE-SHELL:/home/ranime >>cd ./Desktop
SIMPLE-SHELL:/home/ranime/Desktop >>pwd
/home/ranime/Desktop
SIMPLE-SHELL:/home/ranime/Desktop >>|
```

- export & echo commands.

```
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>export x=22
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>echo $x
22
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>echo ranime
ranime
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>
```

- opening the “gedit” application, but the shell is stuck and cannot execute other commands while gedit is open.

```
SIMPLE-SHELL:/home/ranime/Desktop >>gedit
```

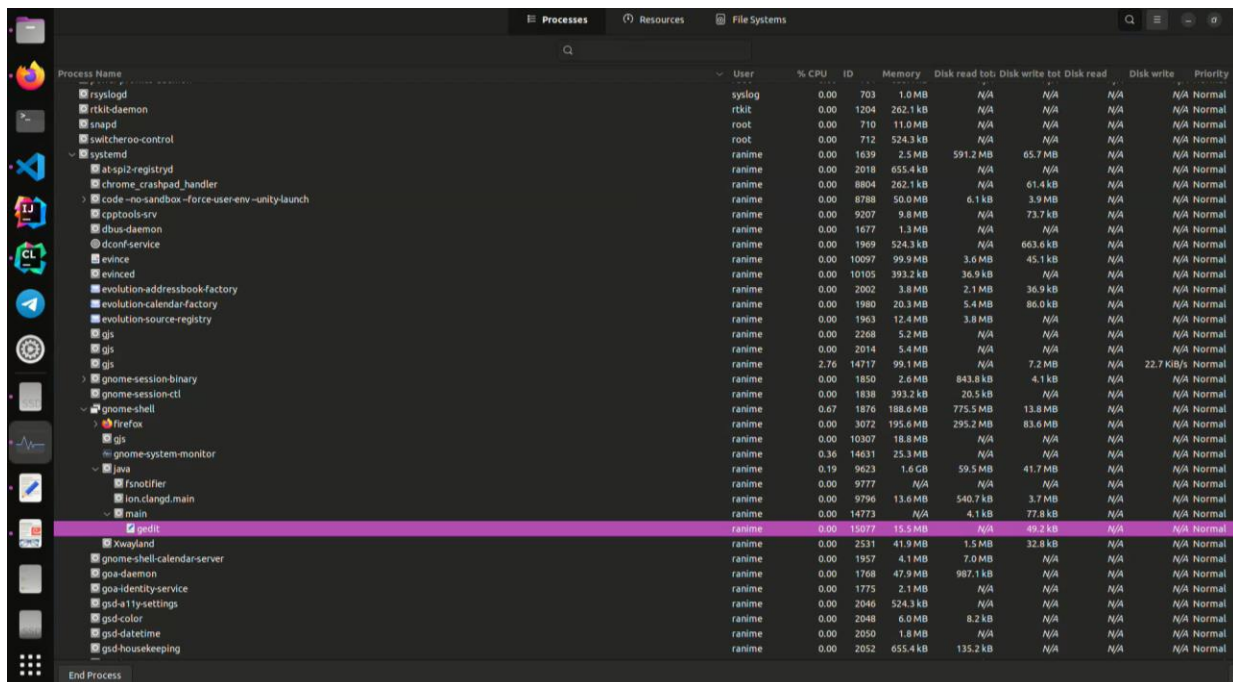
- opening the “gedit” application, but the shell is not stuck and can execute other commands while gedit is open.

```
SIMPLE-SHELL:/home/ranime >>gedit &  
SIMPLE-SHELL:/home/ranime >>
```

- The command “rm”.

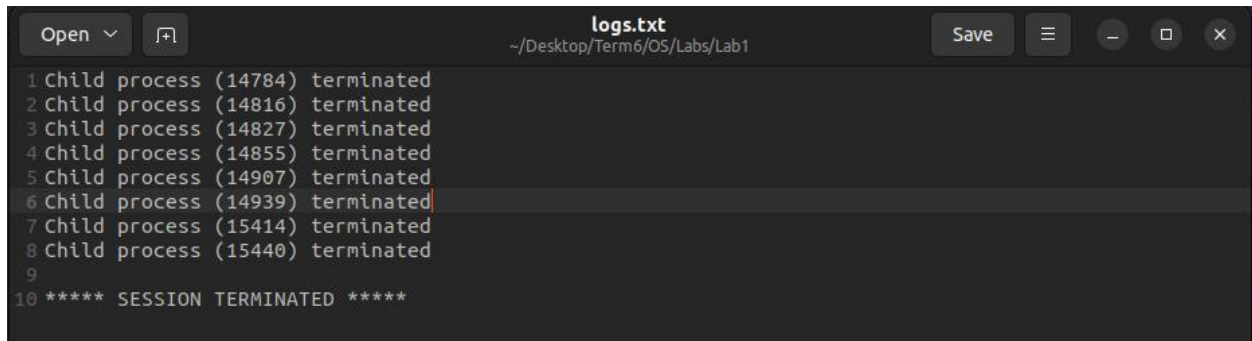
```
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls  
cmake-build-debug exe logs.txt main main.c media test  
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>gedit  
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls  
cmake-build-debug exe file logs.txt main main.c media test  
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>rm file  
SIMPLE-SHELL:/home/ranime/Desktop/Term6/OS/Labs/Lab1 >>ls  
cmake-build-debug exe logs.txt main main.c media test
```

- Here is the system monitor showing that the application (gedit) is a child process of the parent which is genomshell.



Process Name	User	% CPU	ID	Memory	Disk read tot	Disk write tot	Disk read	Disk write	Priority
rsyslogd	syslog	0.00	703	1.0 MB	N/A	N/A	N/A	N/A	Normal
rtkit-daemon	rtkit	0.00	1204	262.1 KB	N/A	N/A	N/A	N/A	Normal
snappy	root	0.00	710	11.0 MB	N/A	N/A	N/A	N/A	Normal
switcheroo-control	root	0.00	712	524.3 KB	N/A	N/A	N/A	N/A	Normal
systemd	ranime	0.00	1639	2.5 MB	591.2 MB	65.7 MB	N/A	N/A	Normal
at-spi2-registr	ranime	0.00	2018	655.4 KB	N/A	N/A	N/A	N/A	Normal
chrome_crashpad_handler	ranime	0.00	8804	262.1 KB	N/A	61.4 KB	N/A	N/A	Normal
code --no-sandbox --force-user-env-unity-launch	ranime	0.00	8788	50.0 MB	6.1 KB	3.9 MB	N/A	N/A	Normal
cpptools-srv	ranime	0.00	9207	9.8 MB	N/A	73.7 KB	N/A	N/A	Normal
dbus-daemon	ranime	0.00	1677	1.3 MB	N/A	N/A	N/A	N/A	Normal
dconf-service	ranime	0.00	1969	524.3 KB	N/A	663.6 KB	N/A	N/A	Normal
evince	ranime	0.00	10097	99.9 MB	3.6 MB	45.1 KB	N/A	N/A	Normal
evince	ranime	0.00	10105	393.2 KB	36.9 KB	N/A	N/A	N/A	Normal
evolution-addressbook-factory	ranime	0.00	2002	3.8 MB	2.1 MB	36.9 KB	N/A	N/A	Normal
evolution-calendar-factory	ranime	0.00	1980	20.3 MB	5.4 MB	86.0 KB	N/A	N/A	Normal
evolution-source-registry	ranime	0.00	1963	12.4 MB	3.8 MB	N/A	N/A	N/A	Normal
gjs	ranime	0.00	2258	5.2 MB	N/A	N/A	N/A	N/A	Normal
gjs	ranime	0.00	2014	5.4 MB	N/A	N/A	N/A	N/A	Normal
gjs	ranime	2.76	14717	99.1 MB	N/A	72.2 KB	N/A	22.7 KB/s	Normal
gnome-session-binary	ranime	0.00	1850	2.6 MB	843.8 KB	4.1 KB	N/A	N/A	Normal
gnome-session-ctl	ranime	0.00	1838	393.2 KB	20.5 KB	N/A	N/A	N/A	Normal
gnome-shell	ranime	0.67	1876	188.6 MB	775.5 MB	13.8 MB	N/A	N/A	Normal
Firefox	ranime	0.00	3072	195.6 MB	295.2 MB	83.6 MB	N/A	N/A	Normal
gjs	ranime	0.00	10307	18.8 MB	N/A	N/A	N/A	N/A	Normal
gnome-system-monitor	ranime	0.36	14631	25.3 MB	N/A	N/A	N/A	N/A	Normal
java	ranime	0.19	9623	1.6 GB	59.5 MB	41.7 MB	N/A	N/A	Normal
fsnotifier	ranime	0.00	9777	N/A	N/A	N/A	N/A	N/A	Normal
ion clangd.main	ranime	0.00	9796	13.6 MB	540.7 KB	3.7 MB	N/A	N/A	Normal
main	ranime	0.00	14773	N/A	4.1 KB	77.8 KB	N/A	N/A	Normal
gedit	ranime	0.00	15077	15.5 MB	N/A	49.2 KB	N/A	N/A	Normal
xwayland	ranime	0.00	2531	41.9 MB	1.5 MB	32.8 KB	N/A	N/A	Normal
gnome-shell-calendar-server	ranime	0.00	1957	4.1 MB	7.0 MB	N/A	N/A	N/A	Normal
go-daemon	ranime	0.00	1768	47.9 MB	987.1 KB	N/A	N/A	N/A	Normal
go-identity-service	ranime	0.00	1775	2.1 MB	N/A	N/A	N/A	N/A	Normal
pid-11ty-settings	ranime	0.00	2046	524.3 KB	N/A	N/A	N/A	N/A	Normal
pid-color	ranime	0.00	2048	6.0 MB	8.2 KB	N/A	N/A	N/A	Normal
pid-datetime	ranime	0.00	2050	1.8 MB	N/A	N/A	N/A	N/A	Normal
pid-housekeeping	ranime	0.00	2052	655.4 KB	135.2 KB	N/A	N/A	N/A	Normal

- Open the log file to see logs of different child processes that were terminated with their process id (PID).



```
logs.txt
~/Desktop/Term6/OS/Labs/Lab1

1 Child process (14784) terminated
2 Child process (14816) terminated
3 Child process (14827) terminated
4 Child process (14855) terminated
5 Child process (14907) terminated
6 Child process (14939) terminated
7 Child process (15414) terminated
8 Child process (15440) terminated
9
10 ***** SESSION TERMINATED *****
```

---

- **Video Link:**

[https://drive.google.com/file/d/105h\\_jtopoDvG\\_mMRzVVr\\_0vf82u06T95/view?usp=sharing](https://drive.google.com/file/d/105h_jtopoDvG_mMRzVVr_0vf82u06T95/view?usp=sharing)

- **Github repo Link:**

<https://github.com/ranimeshehata/Simple-Shell>