Alexandria University - Faculty of Engineering

Computer Systems Engineering Department

Lap Assignment 1 Report
Shell and System Calls

Name: Abanoub Milad Nassief

Seat Num. : 6

Code Organization:

Functional programming paradiam is adapted. Command line input is read and parsed into a command and parameters. Command is checked and executed in a new process (system call) . Parent process waits or continue operating based on a given parameter “&” at the end of command line input.

Functions:

int get_shell_variable(char * variable)

get shell variable value, check if shell variable exists in the variables array
return index of value in the values array if found else -1

bool is_env_variable(char * variable)

check if a given variable is an environmental variable or not based on a predifined list of environmental variables.

void display_history()

open history file, display commands history

void append_history(char *command)

add a command to the end of history file

void append_log()

add a log to the end of log file

bool is_blank(char str[])

check if string is blank or empty

int parse_command(char * line)

parse a line into a command and parameters "if found"
line : (input) the command line
return 1 if command valid 0 otherwise

void handle_command()

determine the command type

void handle_files_names()

handle file names of executables files

void handle_cd()

execute the change directory command with its parameters

void handle_expression(char* sub)

execute the expression assignment commands

bool handle_variables()

replaces the $ variables (shell or environmental) by their equivalent values

void exec_command()

execute the command with its parameters

void start_interactive_mode()

start the interactive mode procedures, take user's input, call parser and executer functions

void start_batch_mode(const char *file_name)

start the batch mode procedures, take batch file, read line, append to history and finally call parser and executer functions

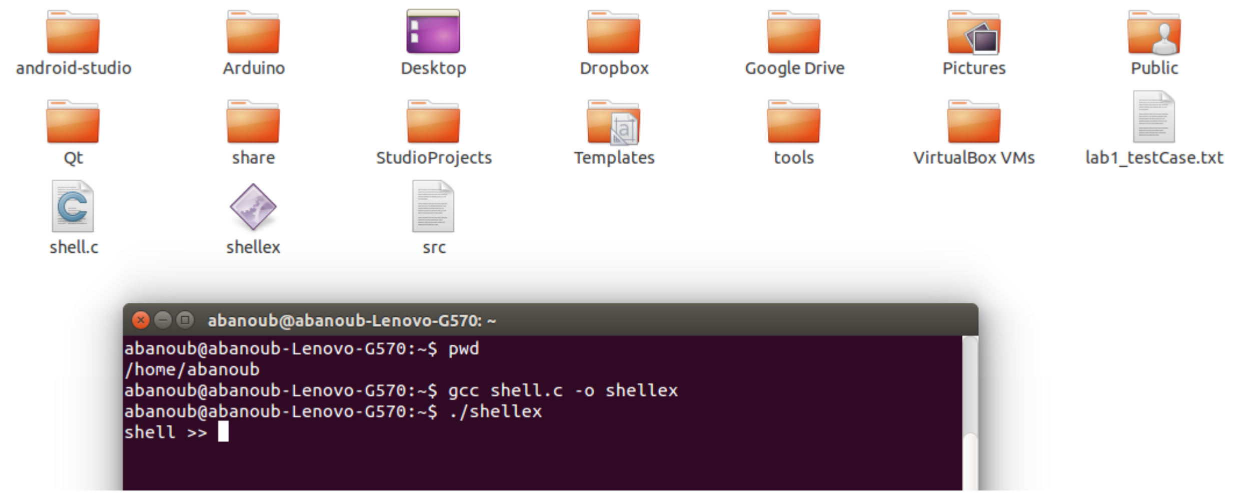
void initialize()

initialize variables and values array, counter and
handle history and log file directroy

main

checks the operation mode interactive or batch based on the arguments.

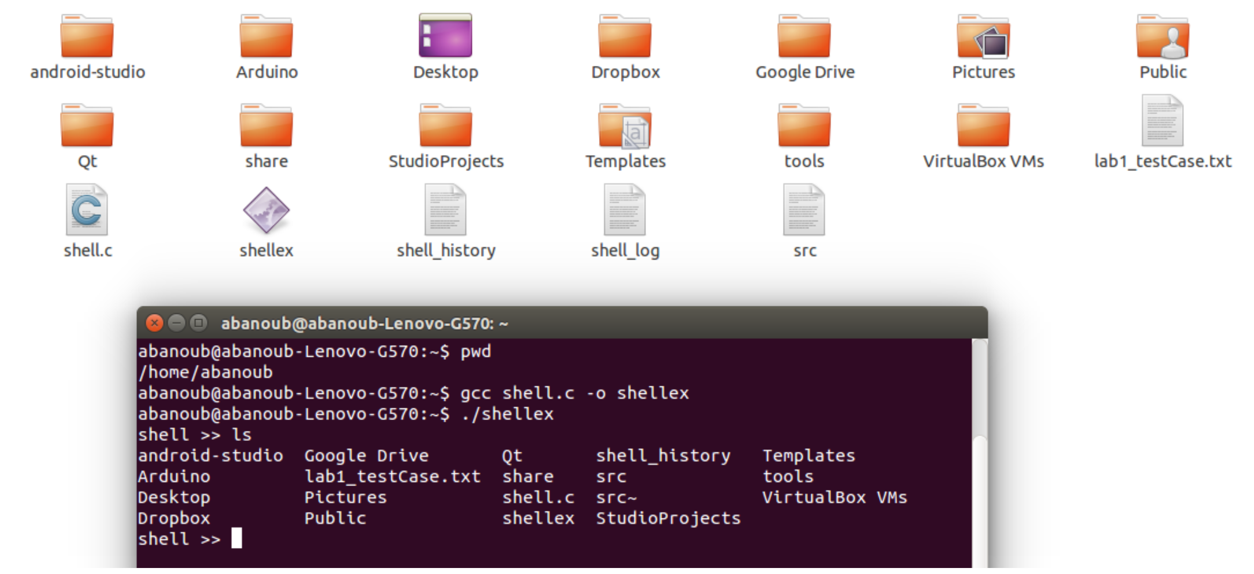
Compiling and runing shell:



```
abanoub@abanoub-Lenovo-G570:~$ pwd
/home/abanoub
abanoub@abanoub-Lenovo-G570:~$ gcc shell.c -o shellex
abanoub@abanoub-Lenovo-G570:~$ ./shellex
shell >> 
```

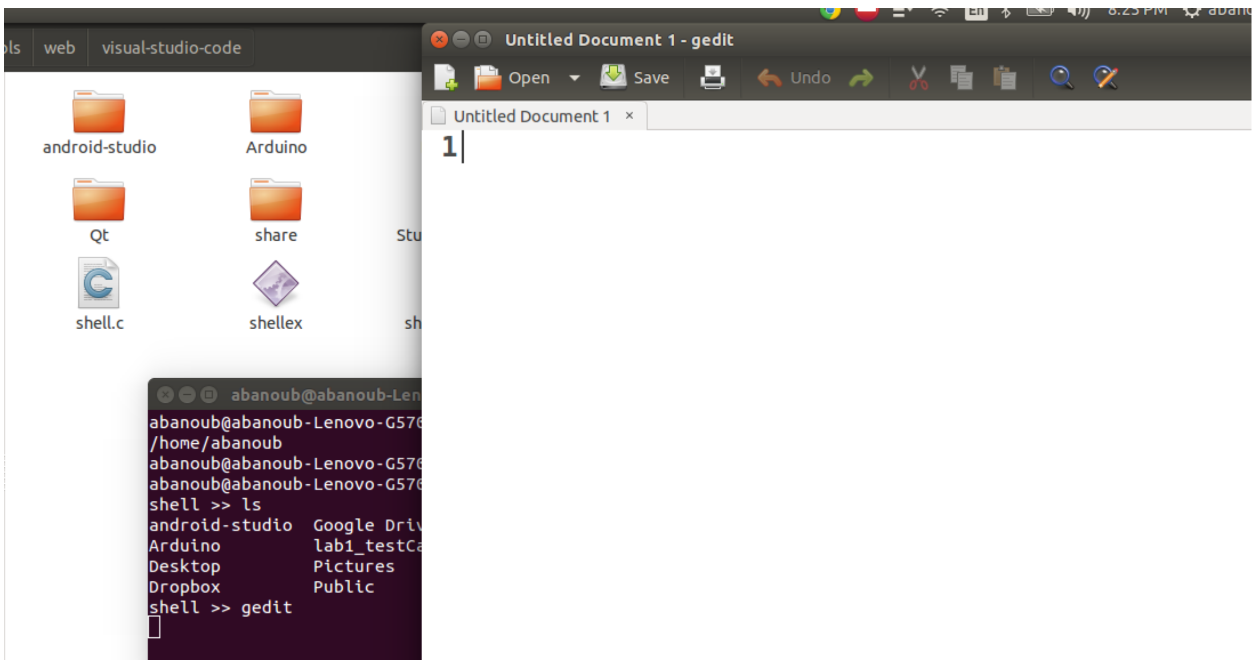
Figure 1 compiling and running shell

Sample Runs:



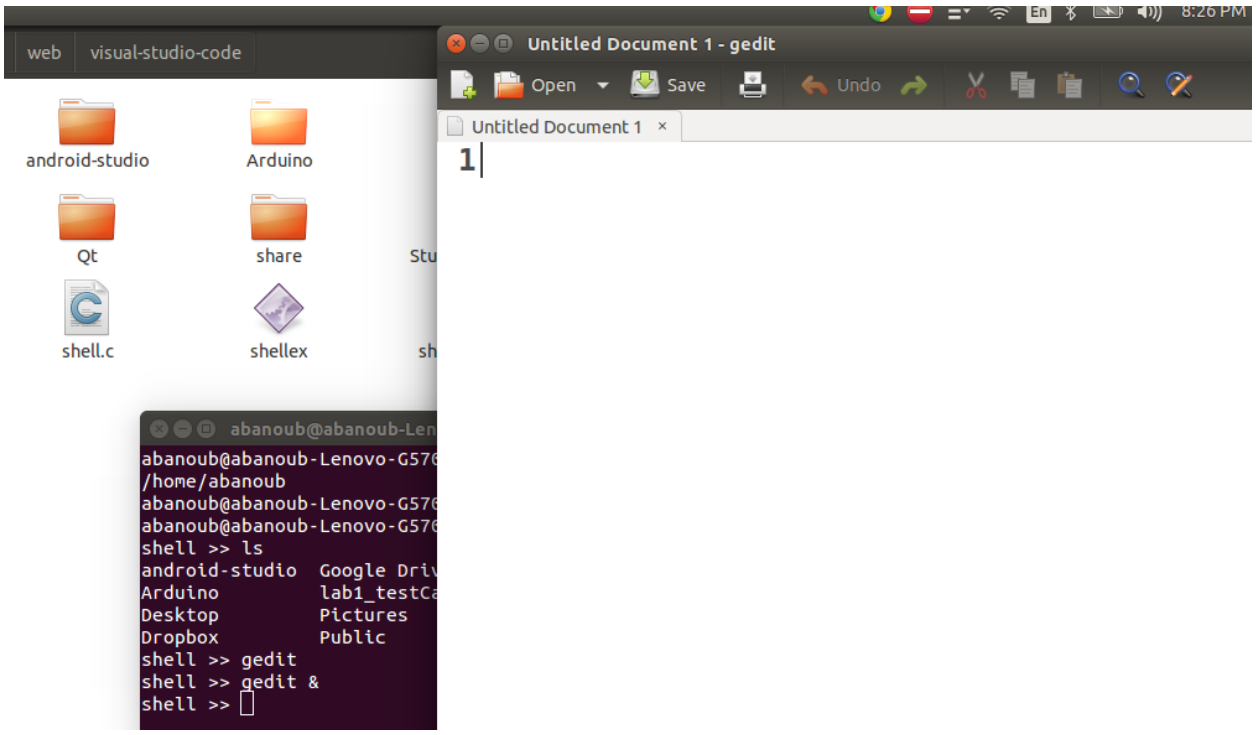
```
abanoub@abanoub-Lenovo-G570:~$ pwd
/home/abanoub
abanoub@abanoub-Lenovo-G570:~$ gcc shell.c -o shellex
abanoub@abanoub-Lenovo-G570:~$ ./shellex
shell >> ls
android-studio  Google Drive       Qt       shell_history   Templates
Arduino         lab1_testCase.txt  share    src             tools
Desktop         Pictures           shell.c  src~            VirtualBox VMs
Dropbox         Public             shellex  StudioProjects
shell >> 
```

Figure 2 ls command



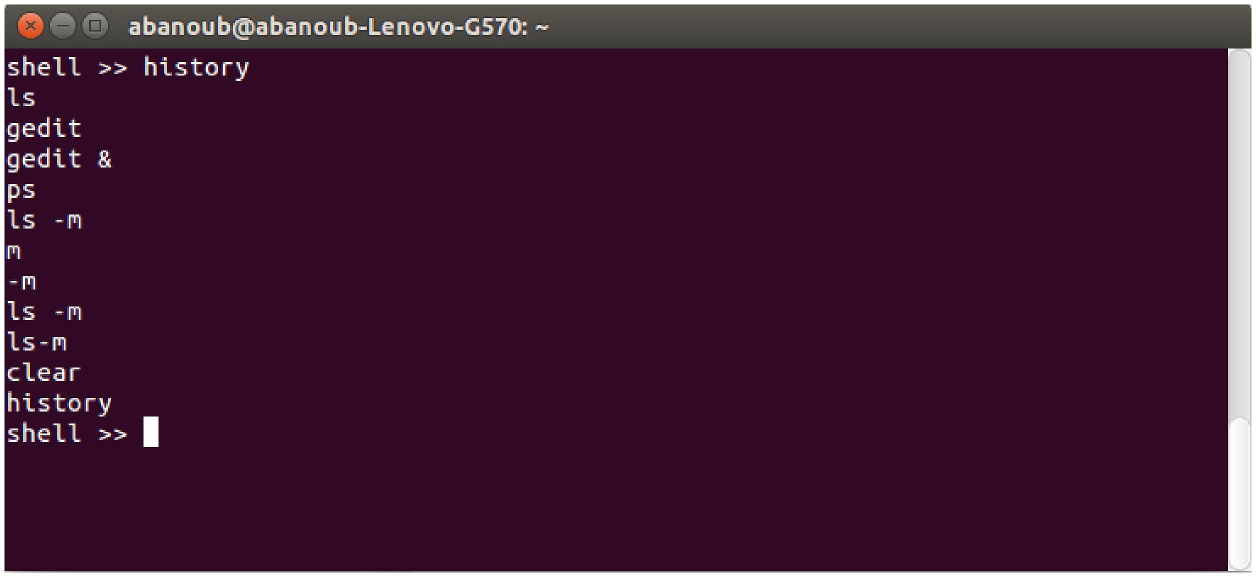
```
abanoub@abanoub-Lenovo-G57
/home/abanoub
abanoub@abanoub-Lenovo-G57
abanoub@abanoub-Lenovo-G57
shell >> ls
android-studio  Google Dri
Arduino         lab1_testCa
Desktop         Pictures
Dropbox         Public
shell >> gedit
```

Figure 3 running gedit in foreground



```
abanoub@abanoub-Lenovo-G57
/home/abanoub
abanoub@abanoub-Lenovo-G57
abanoub@abanoub-Lenovo-G57
shell >> ls
android-studio  Google Dri
Arduino         lab1_testC
Desktop         Pictures
Dropbox         Public
shell >> gedit
shell >> gedit &
shell >>
```

Figure 4 running gedit in background



abanoub@abanoub-Lenovo-G570: ~

```
shell >> history
ls
gedit
gedit &
ps
ls -m
m
-m
ls -m
ls-m
clear
history
shell >>
```

Figure 5 history command

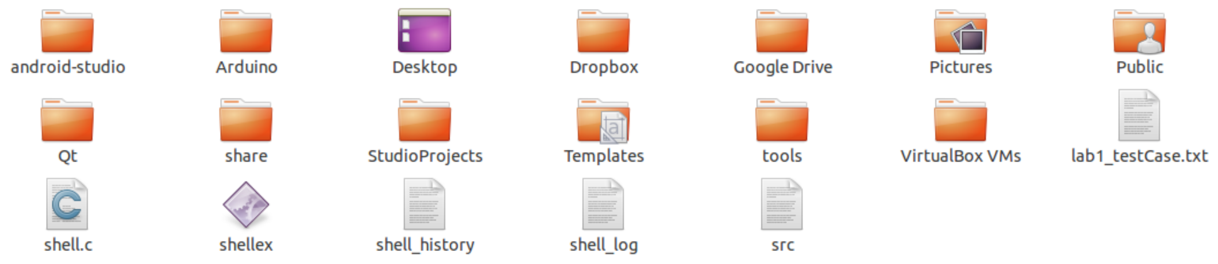
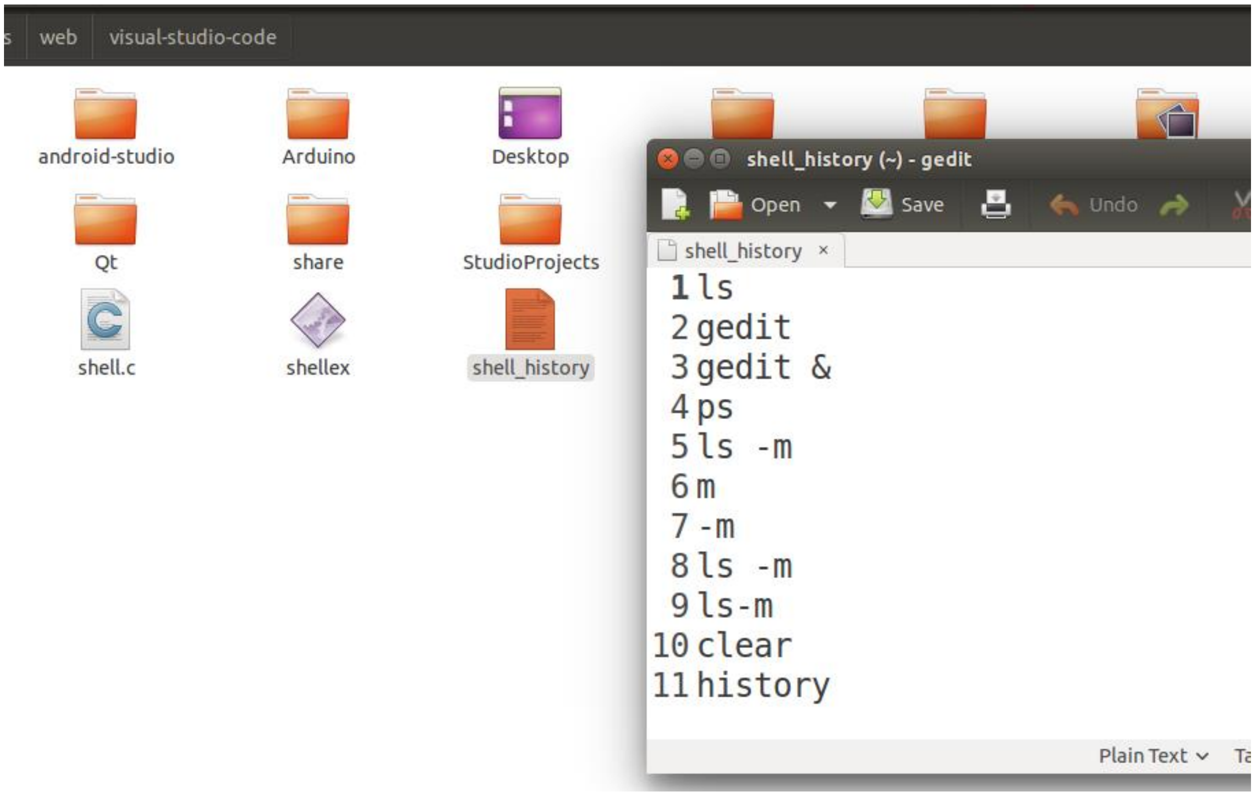
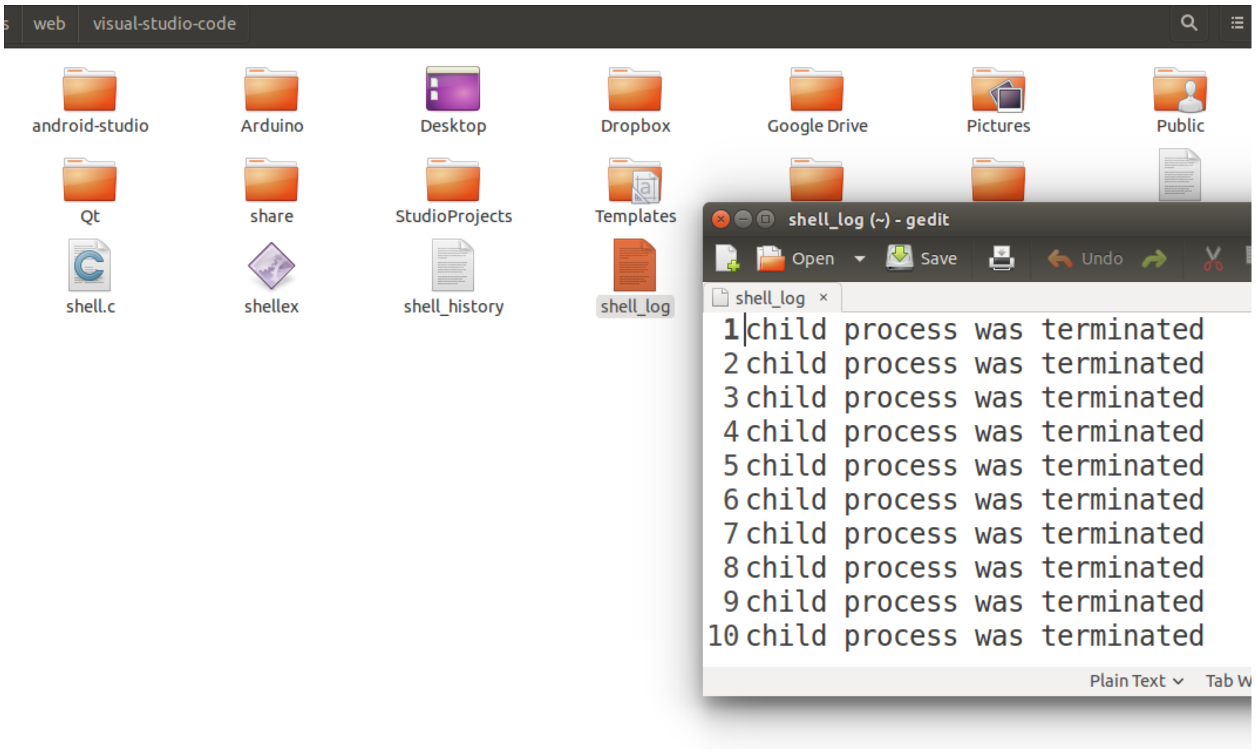


Figure 6 history and log files created after running the shell



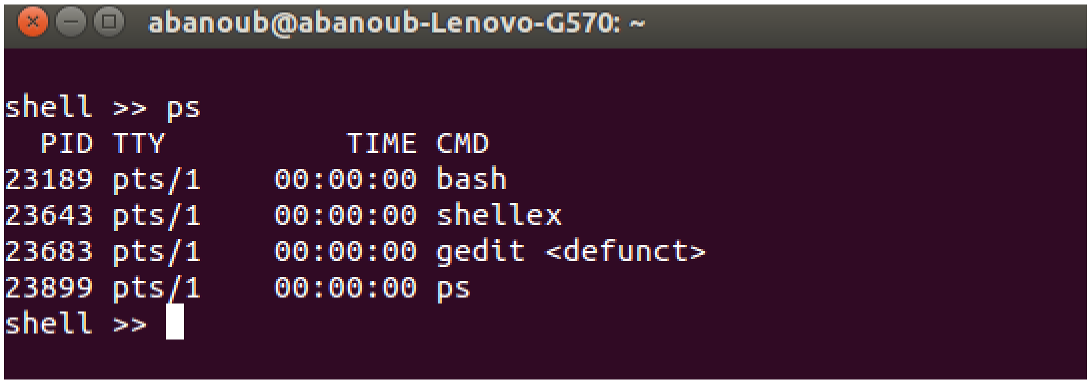
```
1 ls
2 gedit
3 gedit &
4 ps
5 ls -m
6 m
7 -m
8 ls -m
9 ls-m
10 clear
11 history
```

Figure 7 history file



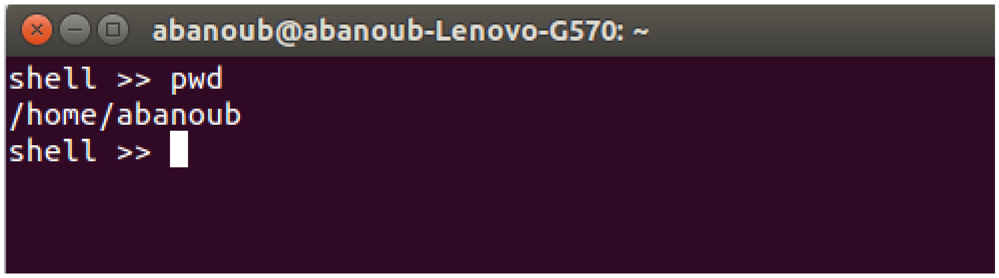
```
child process was terminated
child process was terminated
child process was terminated
child process was terminated
child process was terminated
child process was terminated
child process was terminated
child process was terminated
child process was terminated
child process was terminated
```

Figure 8 log file



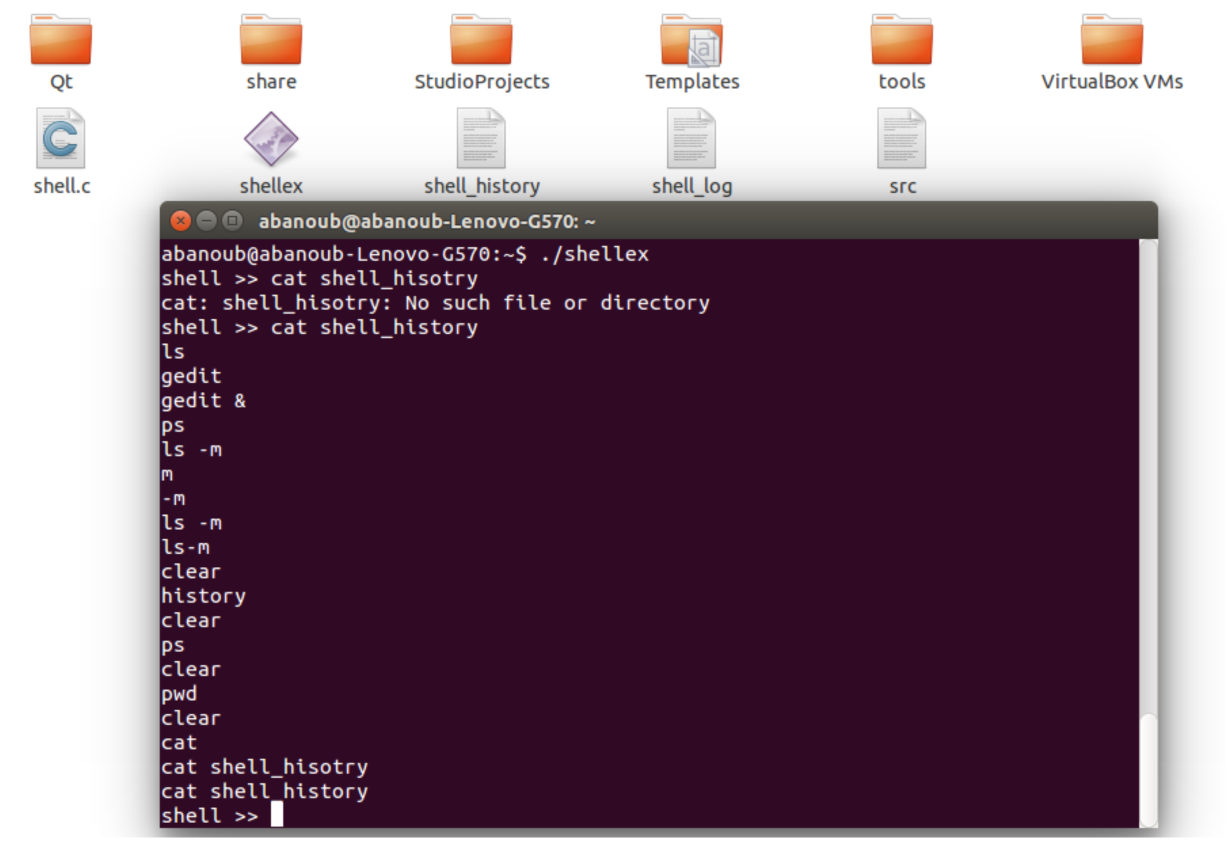
```
shell >> ps
  PID TTY          TIME CMD
23189 pts/1    00:00:00 bash
23643 pts/1    00:00:00 shellex
23683 pts/1    00:00:00 gedit <defunct>
23899 pts/1    00:00:00 ps
shell >>
```

Figure 9 ps command



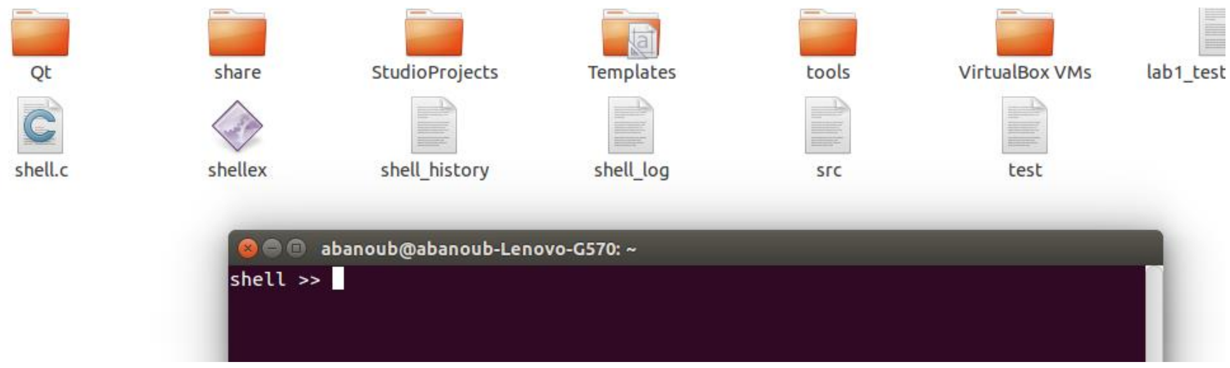
```
shell >> pwd
/home/abanoub
shell >>
```

Figure 10 pwd command



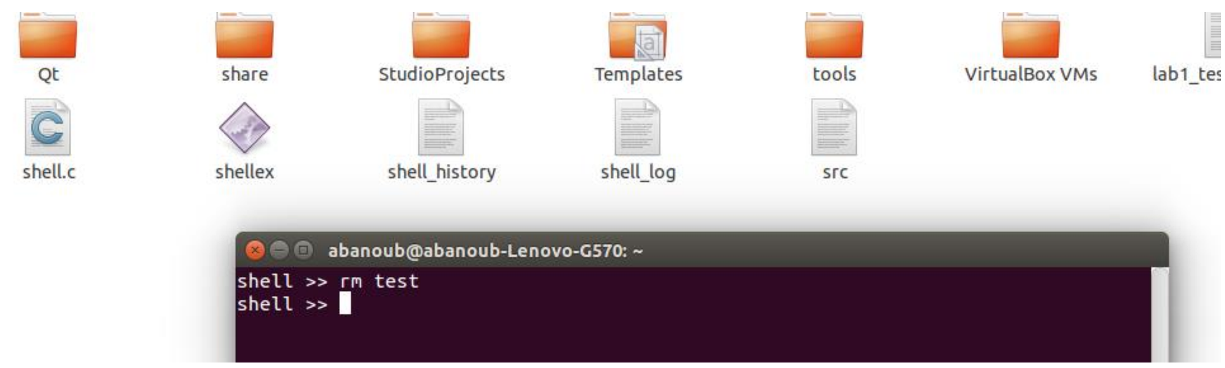
```
abanoub@abanoub-Lenovo-G570:~$ ./shellex
shell >> cat shell_hisotry
cat: shell_hisotry: No such file or directory
shell >> cat shell_history
ls
gedit
gedit &
ps
ls -m
m
-m
ls -m
ls-m
clear
history
clear
ps
clear
pwd
clear
cat
cat shell_hisotry
cat shell_history
shell >> 
```

Figure 11 cat command



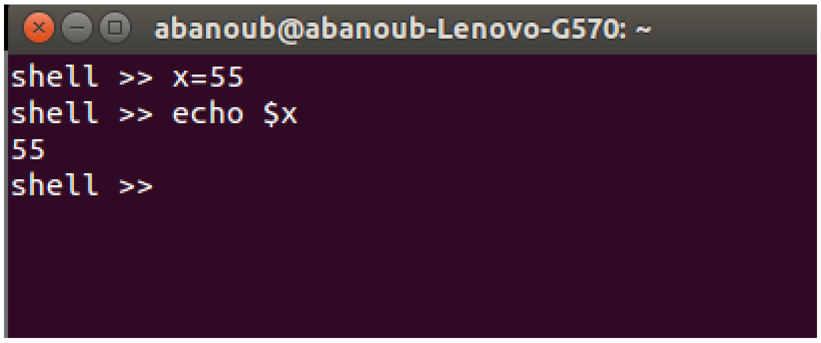
```
shell >> 
```

Figure 12 test file exists before deletion



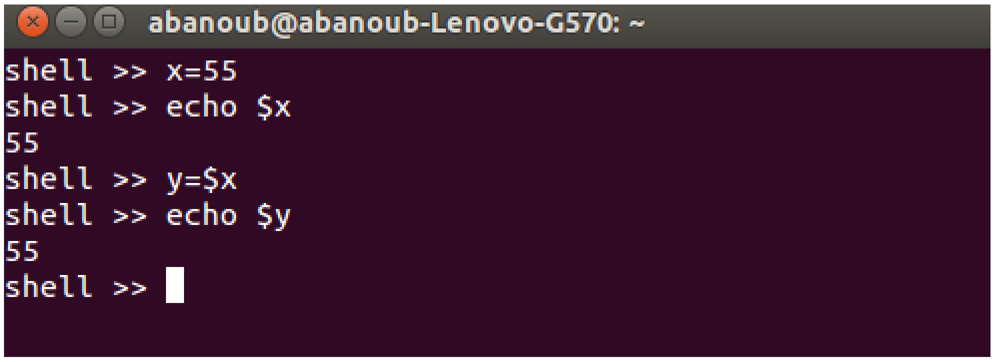
```
shell >> rm test
shell >>
```

Figure 13 deleting test file using rm command



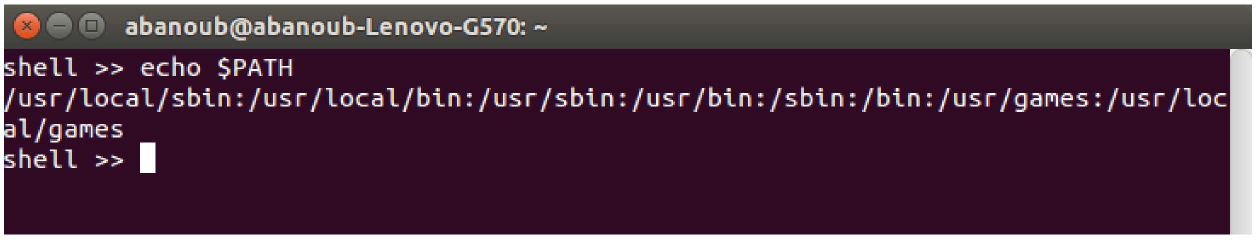
```
shell >> x=55
shell >> echo $x
55
shell >>
```

Figure 14 creating shell variables



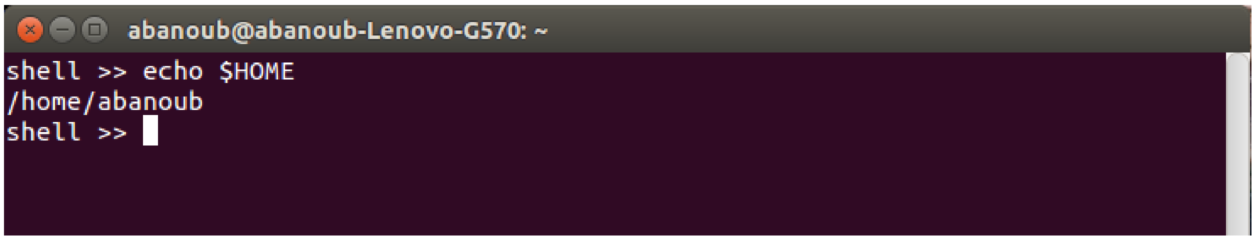
```
shell >> x=55
shell >> echo $x
55
shell >> y=$x
shell >> echo $y
55
shell >>
```

Figure 15 testing shell variables assignment



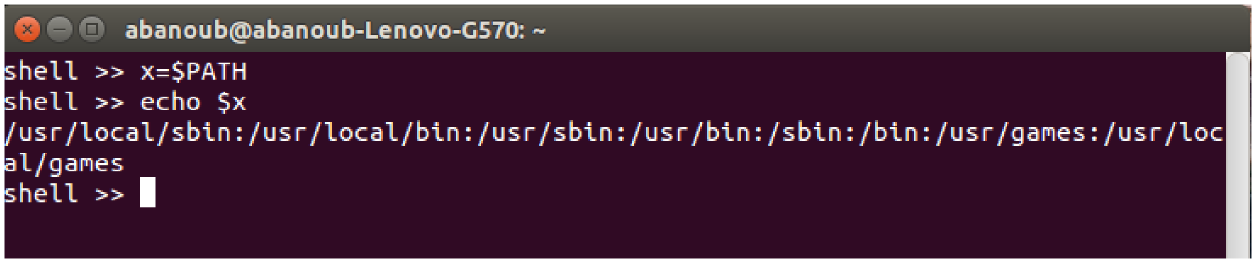
```
shell >> echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc
al/games
shell >>
```

Figure 16 echo command with system variables



```
abanoub@abanoub-Lenovo-G570: ~
shell >> echo $HOME
/home/abanoub
shell >> 
```

Figure 17 echo command with system variables



```
abanoub@abanoub-Lenovo-G570: ~
shell >> x=$PATH
shell >> echo $x
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc
al/games
shell >> 
```

Figure 18 shell and system variables assignment