

COMP 7500 — Project 1: Installing CentOS (Ubuntu Alternative)

Maha Zainab

16 January 2026

1 Overview

I set up an Ubuntu Linux system in a virtual machine and used it as my working environment for this assignment. All development and system exploration tasks were performed directly from the Linux terminal. All terminal activity was recorded using the `script` command and is submitted along with this report.

2 Task 1

I installed Ubuntu on a VirtualBox using the default storage and network configuration. After installation, I updated the system and verified the availability of the required development and system utilities.

Because the assignment was originally written for CentOS, some package names differed on Ubuntu. I installed the appropriate Ubuntu equivalents and verified each tool by invoking it from the command line. Command issues encountered during installation were resolved through repository updates and package verification. The complete command history and outputs are preserved in the submitted script logs.

3 Task 2

I examined system-level information using kernel-provided interfaces and hardware inspection tools. CPU architecture, core configuration, and cache details were obtained from `/proc/cpuinfo`. Memory usage statistics were retrieved from `/proc/meminfo`.

I used `lspci` to enumerate PCI devices recognized by the system. Network configuration details were examined using `ifconfig`, and link information such as interface status and speed was inspected using `ethtool`. I also analyzed `/proc/interrupts` to observe how hardware interrupts were distributed across devices and CPU cores. The outputs from all of these commands were captured during execution and are included in the submitted terminal recordings.

4 Task 3

I wrote a C program named `simple.c` using the Vim editor. The program initializes an array of ten numerical values, computes the square root of each element, accumulates the results, and prints the average of the computed values.

4.1 Source Code

```
#include <stdio.h>
#include <math.h>

int main(){
double numbers[10]={27,56,88,12,35,5,61,68,24,10};
double sum=0.0;
double average;
int i;
for(i=0; i<10;i++){
sum +=sqrt(numbers[i]);
}
average =sum/10;
printf("Final average of the square root numbers=%f\n", average);
return 0;
}
```

5 Compilation and Execution

I compiled the program using the GNU compiler with debugging symbols enabled and explicitly linked the math library. The program compiled successfully and executed correctly from the terminal, producing the expected numerical output. The full compilation and execution process is documented in the script logs.

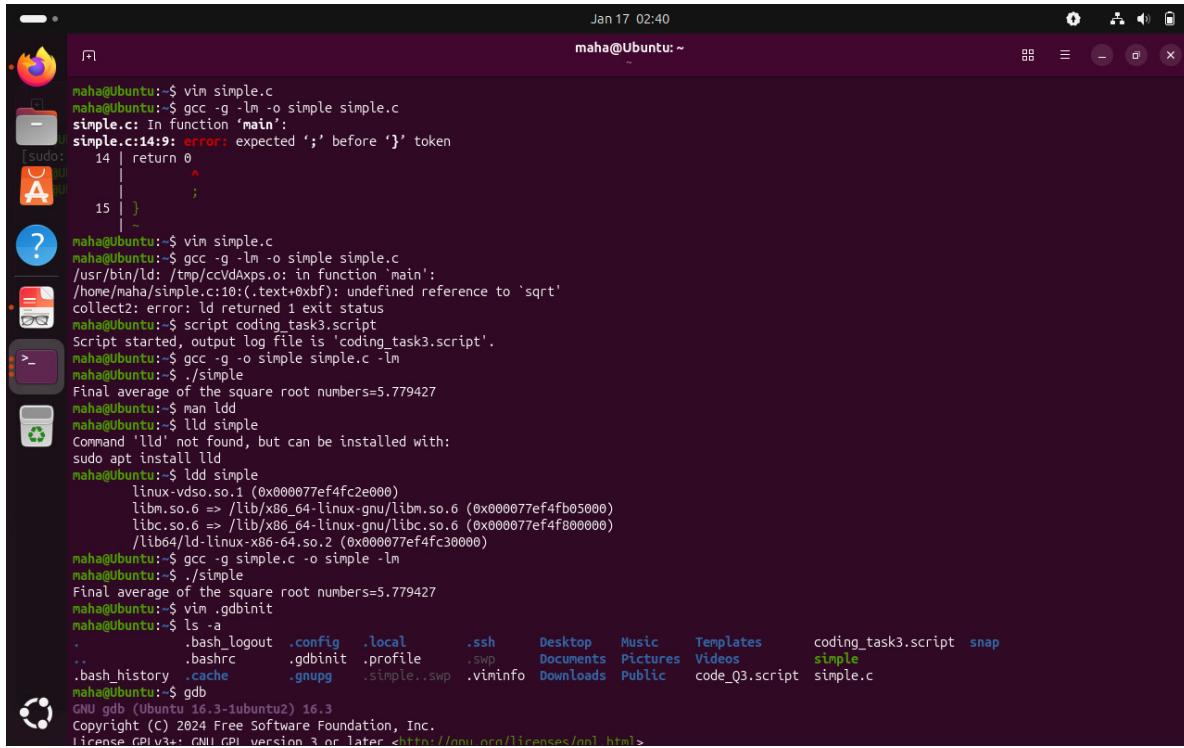
6 Library Dependency Analysis

I used the `ldd` utility to inspect the shared library dependencies of the compiled executable. The output confirmed that the program dynamically links against standard system libraries, including the C runtime and the math library.

7 Debugging with gdb

I debugged the program using the GNU debugger. I created a local `.gdbinit` file to automatically load the executable and set breakpoints at both the `main` function and the `sqrt` function. I then executed the program inside `gdb` and traced its behavior using the `run`, `next`, `step`, and `continue` commands. Screenshots and terminal transcripts documenting this process are included with the submission.

8 Screenshots



The screenshot shows a terminal window on a dark-themed desktop environment. The terminal title is "maha@Ubuntu:~". The session starts with a syntax error in simple.c:

```
maha@Ubuntu:~$ vim simple.c
maha@Ubuntu:~$ gcc -g -lm -o simple simple.c
simple.c: In function `main':
simple.c:14:9: error: expected ';' before '}' token
  14 |     return 0;
      |     ^
      |     ;
  15 | }
      | ~
```

After fixing the code, it compiles and runs:

```
maha@Ubuntu:~$ vim simple.c
maha@Ubuntu:~$ gcc -g -lm -o simple simple.c
/usr/bin/ld: /tmp/ccVdAxpso.o: in function 'main':
/home/maha/simple.c:10:(.text+0xbf): undefined reference to 'sqrt'
collect2: error: ld returned 1 exit status
maha@Ubuntu:~$ script coding_task3.script
Script started, output log file is 'coding_task3.script'.
maha@Ubuntu:~$ gcc -g -o simple simple.c -lm
maha@Ubuntu:~$ ./simple
Final average of the square root numbers=5.779427
```

It then uses ldd to check dependencies:

```
maha@Ubuntu:~$ man ldd
maha@Ubuntu:~$ ldd simple
Command 'lld' not found, but can be installed with:
sudo apt install lld
maha@Ubuntu:~$ ldd simple
linux-vdso.so.1 (0x000077ef4fc2e000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x000077ef4fb05000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000077ef4f800000)
/lib64/ld-linux-x86-64.so.2 (0x000077ef4fc30000)
```

It recompiles and runs again:

```
maha@Ubuntu:~$ gcc -g simple.c -o simple -lm
maha@Ubuntu:~$ ./simple
Final average of the square root numbers=5.779427
```

It then uses vim to edit .gdbinit:

```
maha@Ubuntu:~$ vim .gdbinit
maha@Ubuntu:~$ ls -a
.               .bash_logout  .config   .local    .ssh     Desktop  Music   Templates   coding_task3.script  snap
..              .bashrc       .gdbinit  .profile  .swp     Documents Pictures Videos   simple
.bash_history  .cache       .gnupg    .simple..swp .viminfo Downloads Public   code_Q3.script simple.c
maha@Ubuntu:~$ gdb
```

Finally, it runs the debugger:

```
GNU gdb (Ubuntu 16.3-1ubuntu2) 16.3
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
```

Figure 1: Compilation, execution, and script logging

```

maha@Ubuntu:~ Jan 17 02:40
maha@Ubuntu:~ $ gdb
GNU gdb (Ubuntu 16.3-1ubuntu2) 16.3
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type 'show copying' and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type 'show configuration' for configuration details.
Type 'show reporting' for bug reporting instructions.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type 'apropos word' to search for commands related to "word".
Breakpoint 1 at 0x1095: file simple.c, line 4.
Breakpoint 2 at 0x1090
/home/maha/.gdbinit:4: Error in sourced command file:
The program has no registers now.
(gdb) r
Starting program: /home/maha/simple

This GDB supports auto-downloading debuginfo from the following URLs:
<https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.

Breakpoint 1, main () at simple.c:4
4 int main(){
(gdb) n
5 double numbers[10]={27,56,88,12,35,5,61,68,24,10};
(gdb) n
6 double sum=0.0;
(gdb) n

```

Figure 2: Starting gdb and hitting the breakpoint at main

```

maha@Ubuntu:~ Jan 17 02:41
maha@Ubuntu:~ $ 
using most common db library /lib/x86_64-linux-gnu/libthread_db.so.1 .
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.

Breakpoint 1, main () at simple.c:4
4 int main(){
(gdb) n
5 double numbers[10]={27,56,88,12,35,5,61,68,24,10};
(gdb) n
6 double sum=0.0;
(gdb) n
9 for(i=0; i<10;i++){
(gdb) n
10 sum +=sqrt(numbers[i]);
(gdb) n
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.

Breakpoint 2, __sqrt (<=27) at ./math/w_sqrt_compat.c:34
warning: 34 ./math/w_sqrt_compat.c: No such file or directory
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/maha/simple
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.

Breakpoint 1, main () at simple.c:4
4 int main(){
(gdb) c
Continuing.
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.

Breakpoint 2, __sqrt (<=27) at ./math/w_sqrt_compat.c:34
warning: 34 ./math/w_sqrt_compat.c: No such file or directory
(gdb) n
35 in ./math/w_sqrt_compat.c
(gdb) n
38 in ./math/w_sqrt_compat.c
(gdb) q
A debugging session is active.

```

Figure 3: Stepping through the program and entering sqrt

The screenshot shows a terminal window titled "maha@Ubuntu: ~" with a dark theme. The window contains a GDB session. The user has set a breakpoint at line 27 of the file "/math/w_sqrt_compat.c". They have stepped over the breakpoint (n), which triggered a warning about a missing source file. They then continued execution (r) and quit the debugger (q). The terminal also shows the user's exit from the script and the final command prompt.

```
Jan 17 02:41
maha@Ubuntu: ~
(gdb) n
9      for(i=0; i<10;i++){
(gdb) n
10     sum +=sqrt(numbers[i]);
(gdb) n
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.
Breakpoint 2, __sqrt (<=27) at ./math/w_sqrt_compat.c:34
warning: 34 ./math/w_sqrt_compat.c: No such file or directory
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/maha/simple
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.
Breakpoint 1, main () at simple.c:4
4     int main(){
(gdb) c
Continuing.
Download failed: Invalid argument. Continuing without source file ./math./math/w_sqrt_compat.c.

Breakpoint 2, __sqrt (<=27) at ./math/w_sqrt_compat.c:34
warning: 34 ./math/w_sqrt_compat.c: No such file or directory
(gdb) n
35     in ./math/w_sqrt_compat.c
(gdb) n
38     in ./math/w_sqrt_compat.c
(gdb) q
A debugging session is active.

Inferior 1 [process 8185] will be killed.

Quit anyway? (y or n) y
maha@Ubuntu:~$ exit
exit
Script done.
maha@Ubuntu:~$
```

Figure 4: Continuing execution and exiting gdb