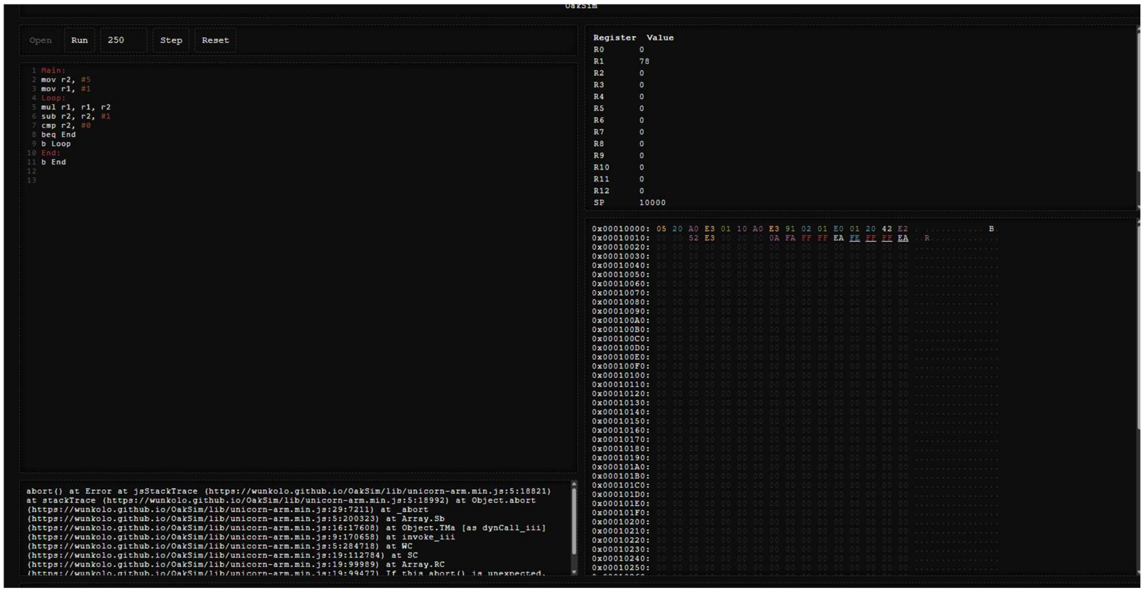
Template Week 4 – Software

Student number: 575798

Assignment 4.1: ARM assembly

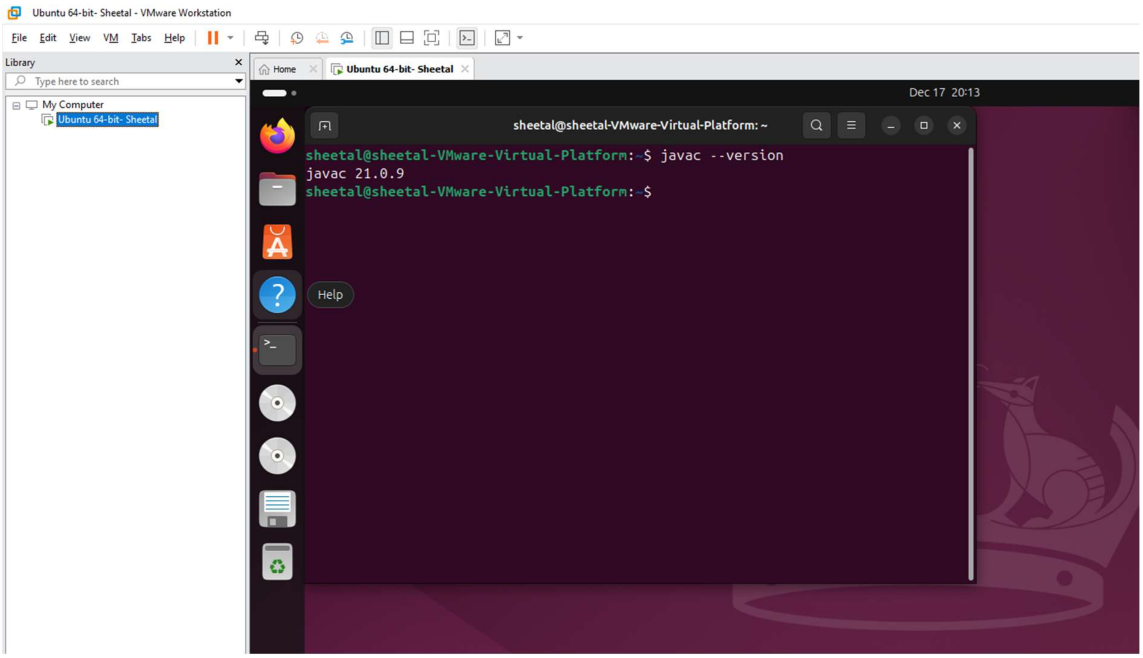
Screenshot of working assembly code of factorial calculation:



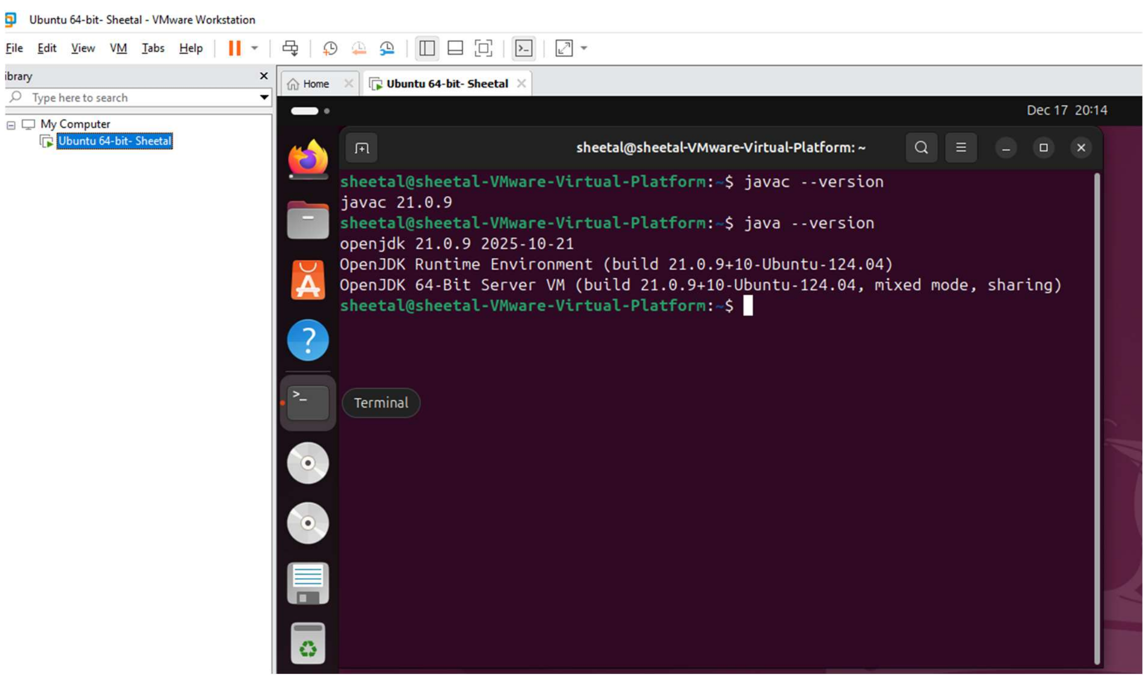
Assignment 4.2: Programming languages

Take screenshots that the following commands work:

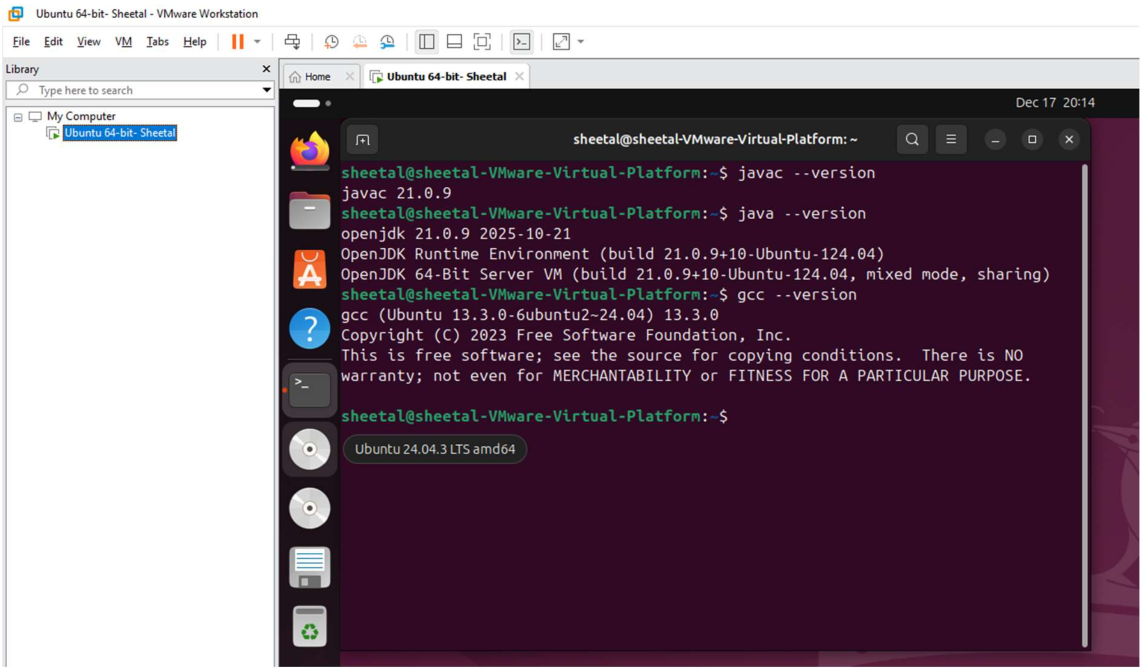
javac –version



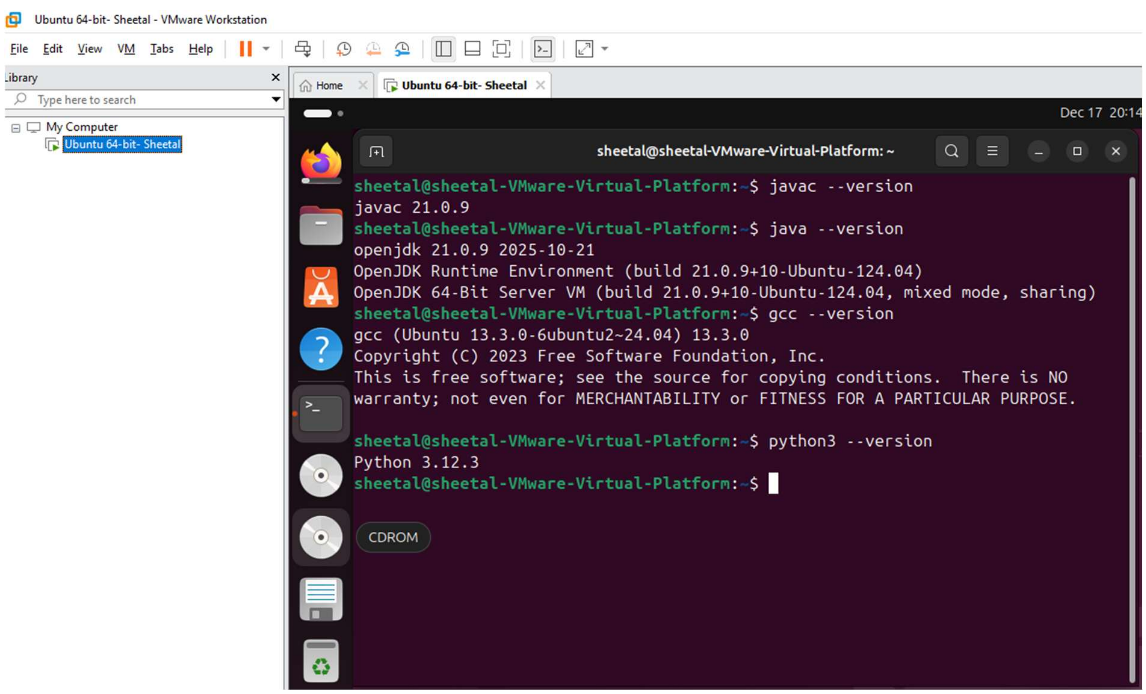
java –version



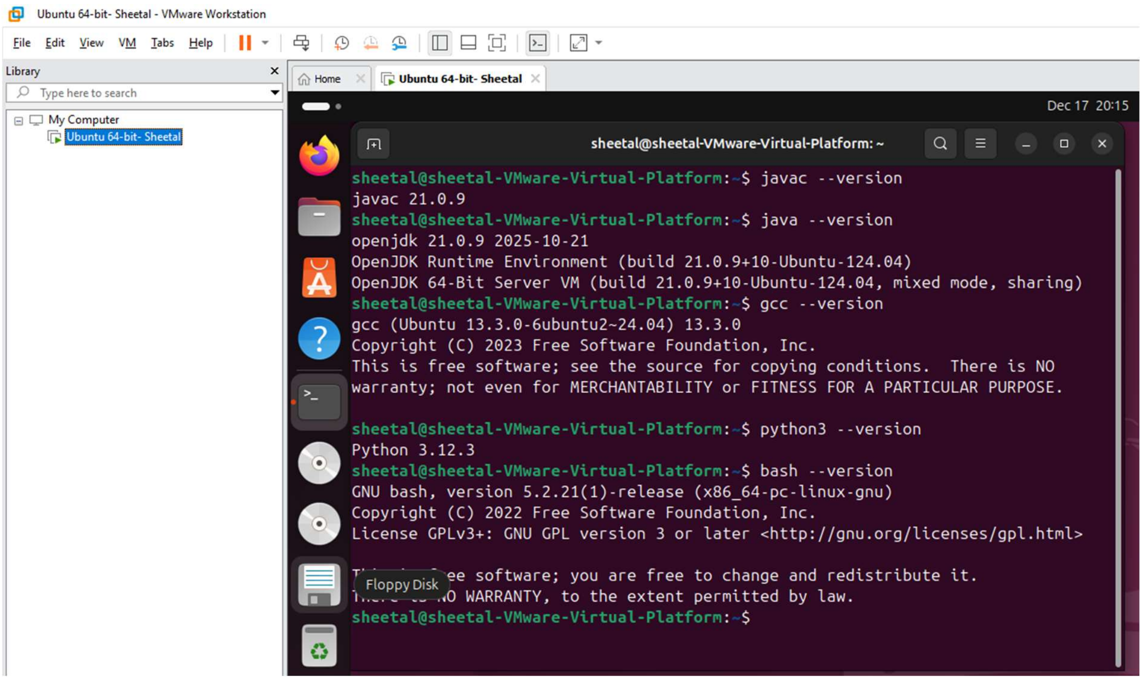
gcc –version



python3 –version



bash –version


Ubuntu 64-bit- Sheetal - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit- Sheetal
Home
Ubuntu 64-bit- Sheetal
Dec 17 20:15
sheetal@sheetal-VMware-Virtual-Platform: ~
sheetal@sheetal-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
sheetal@sheetal-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.9 2025-10-21
OpenJDK Runtime Environment (build 21.0.9+10-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 21.0.9+10-Ubuntu-124.04, mixed mode, sharing)
sheetal@sheetal-VMware-Virtual-Platform:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
sheetal@sheetal-VMware-Virtual-Platform:~$ python3 --version
Python 3.12.3
sheetal@sheetal-VMware-Virtual-Platform:~$ bash --version
GNU bash, version 5.2.21(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
Floppy Disk
ee software; you are free to change and redistribute it.
O WARRANTY, to the extent permitted by law.
sheetal@sheetal-VMware-Virtual-Platform:~$

Assignment 4.3: Compile

Which of the above files need to be compiled before you can run them?

The C source file needs to be compiled. The java source file needs to be compiled. The python file and bash script does not need to be compiled.

Which source code files are compiled into machine code and then directly executable by a processor?

The C compiler (gcc) compiles C code directly into native machine code, which is executed directly by the CPU.

Which source code files are compiled to byte code?

Java source code is compiled by java c into Java bytecode which runs on the Java Virtual Machine.

Which source code files are interpreted by an interpreter?

Python (.py) interpreted by the Python interpreter

Bash (.sh) interpreted by the Bash shell

These source code files will perform the same calculation after compilation/interpretation. Which one is expected to do the calculation the fastest?

C program is expected to be the fastest since it is compiled into native machine code. Also, because there is no virtual machine or interpreter overhead and it runs directly on the CPU.

How do I run a Java program?

Compile javac Fib.java

How do I run a Python program?

python3 fib.py

How do I run a C program?

gcc fib.c -o fib

How do I run a Bash script?

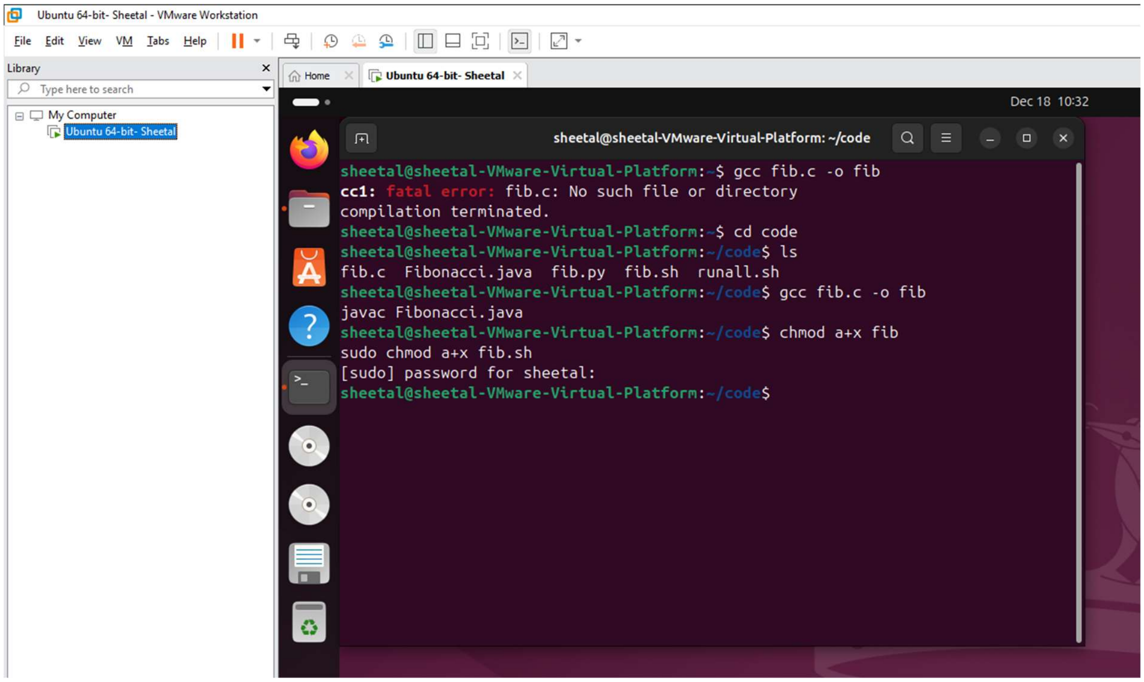
sudo chmod a+x fib.sh

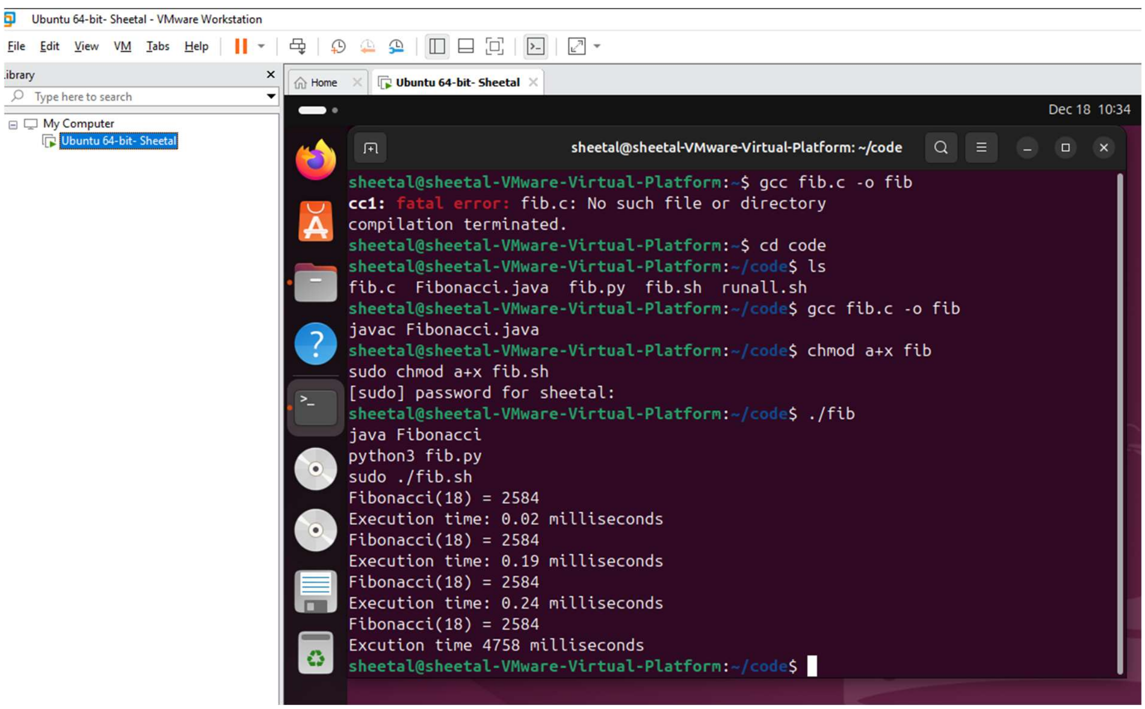
If I compile the above source code, will a new file be created? If so, which file?

For java there will be a bytecode file and for C there will be an executable file. For python and bash there will be no compiled file.

Take relevant screenshots of the following commands:

- Compile the source files where necessary
- Make them executable
- Run them
- Which (compiled) source code file performs the calculation the fastest?


Ubuntu 64-bit- Sheetal - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit- Sheetal
Home
Ubuntu 64-bit- Sheetal
Dec 18 10:32
sheetal@sheetal-VMware-Virtual-Platform: ~/code
sheetal@sheetal-VMware-Virtual-Platform:~$ gcc fib.c -o fib
cc1: fatal error: fib.c: No such file or directory
compilation terminated.
sheetal@sheetal-VMware-Virtual-Platform:~$ cd code
sheetal@sheetal-VMware-Virtual-Platform:~/code$ ls
fib.c Fibonacci.java fib.py fib.sh runall.sh
sheetal@sheetal-VMware-Virtual-Platform:~/code$ gcc fib.c -o fib
javac Fibonacci.java
sheetal@sheetal-VMware-Virtual-Platform:~/code$ chmod a+x fib
sudo chmod a+x fib.sh
[sudo] password for sheetal:
sheetal@sheetal-VMware-Virtual-Platform:~/code$

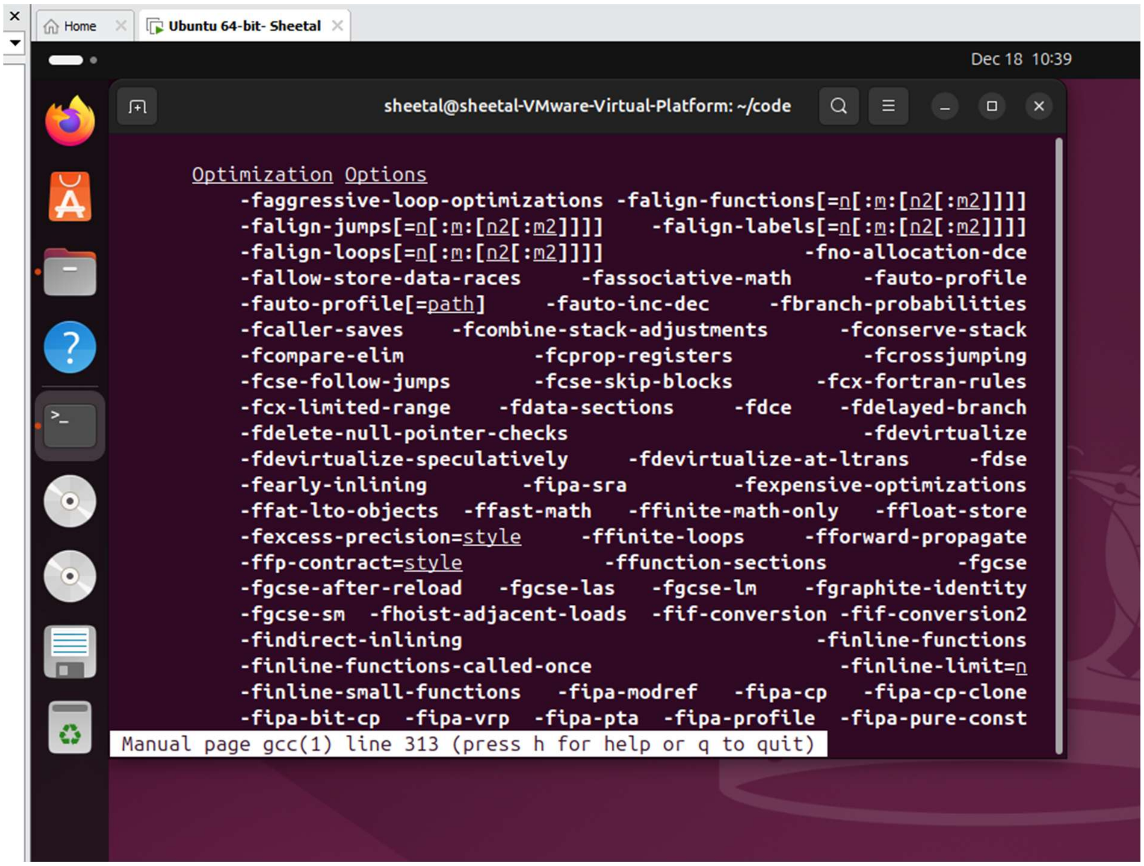

Ubuntu 64-bit- Sheetal - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit- Sheetal
Home
Ubuntu 64-bit- Sheetal
Dec 18 10:34
sheetal@sheetal-VMware-Virtual-Platform: ~/code
sheetal@sheetal-VMware-Virtual-Platform:~$ gcc fib.c -o fib
cc1: fatal error: fib.c: No such file or directory
compilation terminated.
sheetal@sheetal-VMware-Virtual-Platform:~$ cd code
sheetal@sheetal-VMware-Virtual-Platform:~/code$ ls
fib.c Fibonacci.java fib.py fib.sh runall.sh
sheetal@sheetal-VMware-Virtual-Platform:~/code$ gcc fib.c -o fib
javac Fibonacci.java
sheetal@sheetal-VMware-Virtual-Platform:~/code$ chmod a+x fib
sudo chmod a+x fib.sh
[sudo] password for sheetal:
sheetal@sheetal-VMware-Virtual-Platform:~/code$ ./fib
java Fibonacci
python3 fib.py
sudo ./fib.sh
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds
Fibonacci(18) = 2584
Execution time: 0.19 milliseconds
Fibonacci(18) = 2584
Execution time: 0.24 milliseconds
Fibonacci(18) = 2584
Excution time 4758 milliseconds
sheetal@sheetal-VMware-Virtual-Platform:~/code$

C calculates the fastest.

Assignment 4.4: Optimize

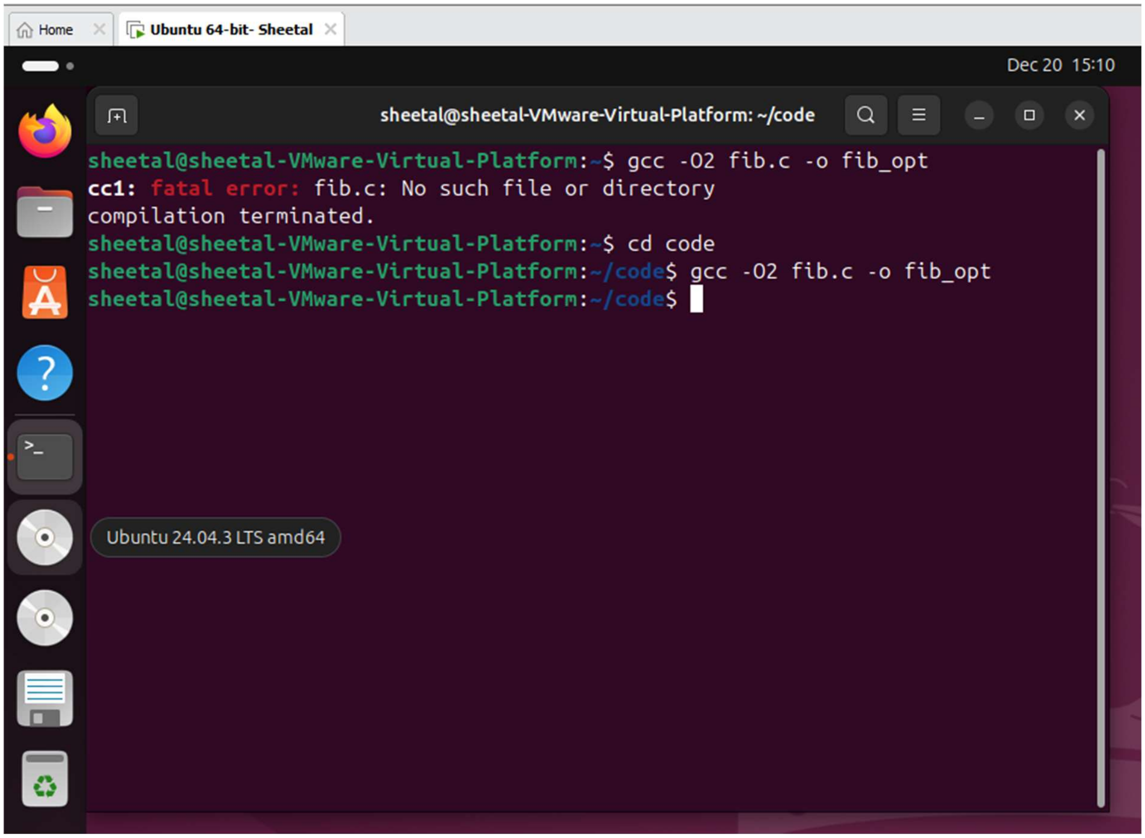
Take relevant screenshots of the following commands:

a) Figure out which parameters you need to pass to **the gcc** compiler so that the compiler performs a number of optimizations that will ensure that the compiled source code will run faster. **Tip!** The parameters are usually a letter followed by a number. Also read **page 191** of your book, but find a better optimization in the man pages. Please note that Linux is case sensitive.

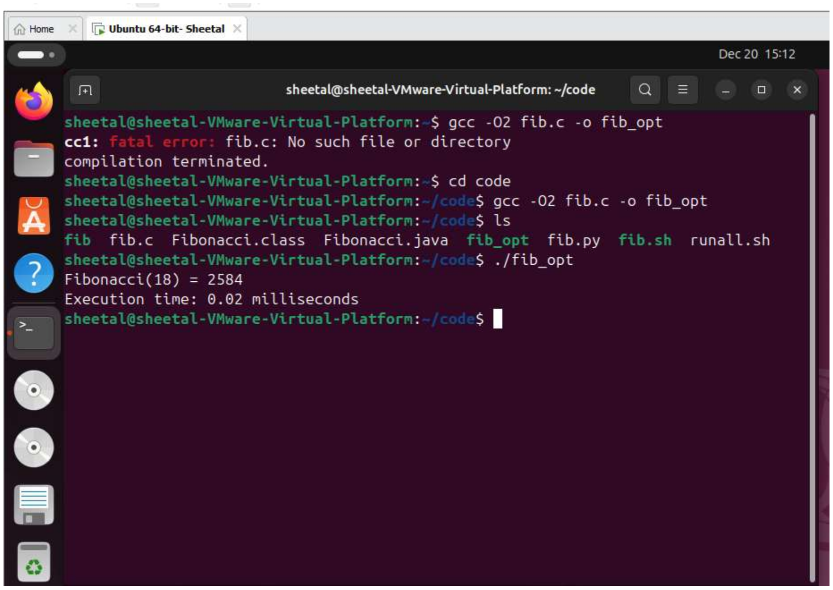


I can see that gcc supports multiple optimisation flags. The recommended optimisation level is -02 because there is a wider range or optimisations that can increase the speed.

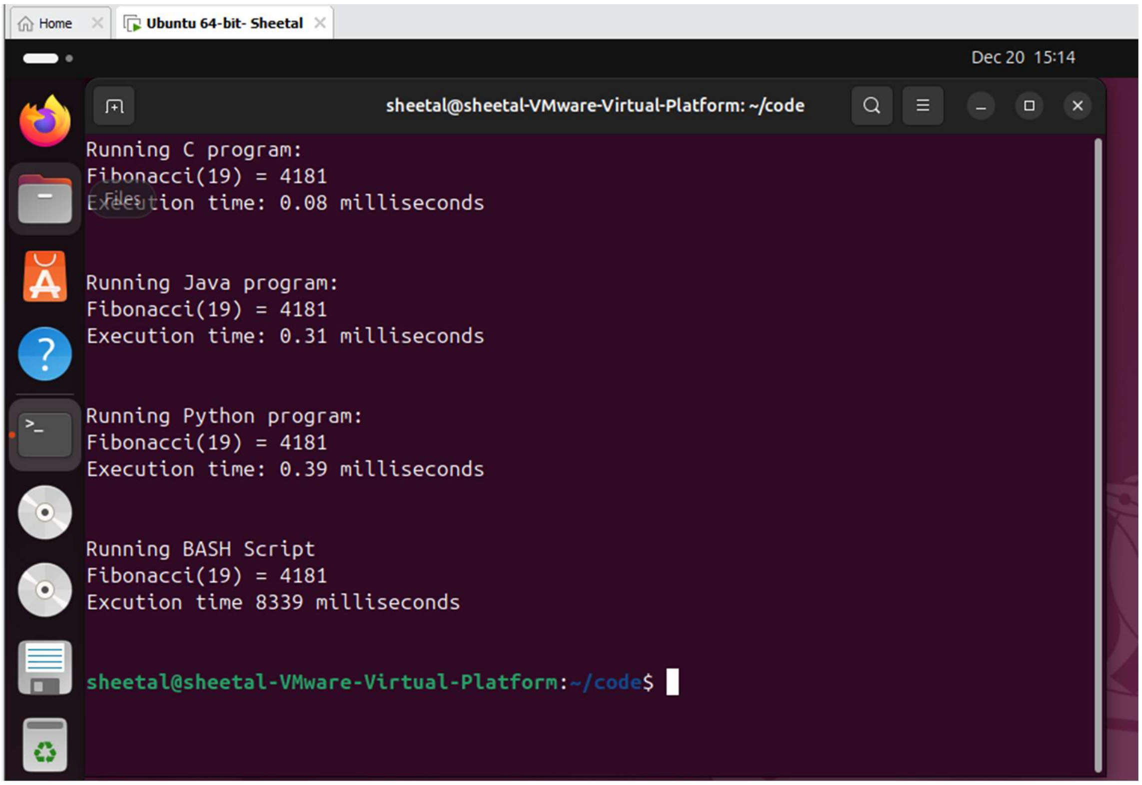
b) Compile **fib.c** again with the optimization parameters



c) Run the newly compiled program. Is it true that it now performs the calculation faster?



d) Edit the file **runall.sh**, so you can perform all four calculations in a row using this Bash script. So the (compiled/interpreted) C, Java, Python and Bash versions of Fibonacci one after the other.



As we can see C program performs the calculations the fastest.

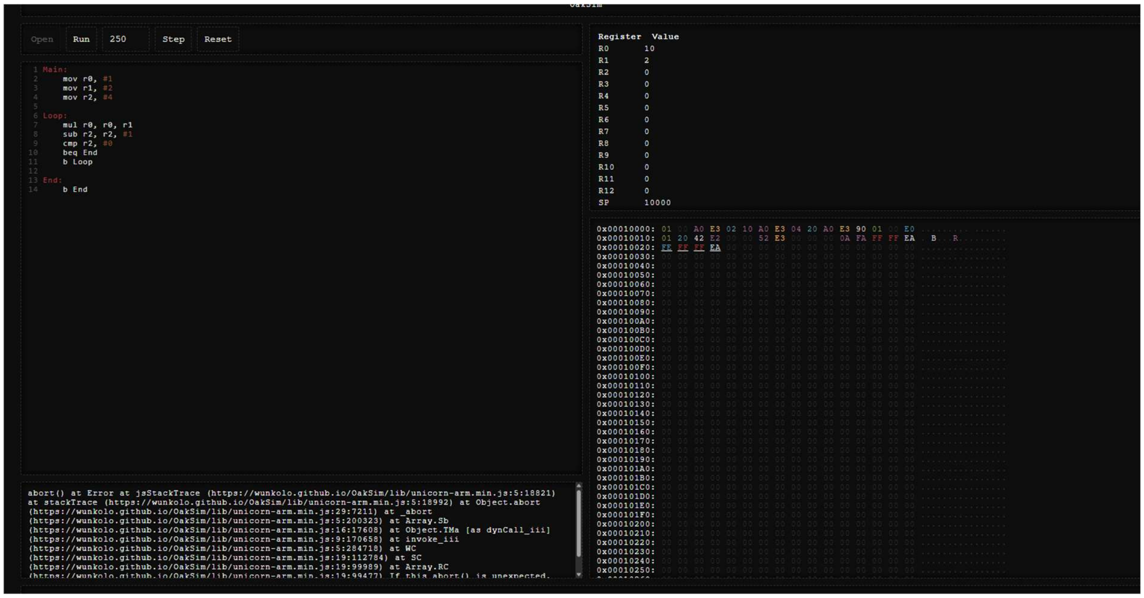
Assignment 4.5: More ARM Assembly

Like the factorial example, you can also implement the calculation of a power of 2 in assembly. For example you want to calculate $2^4 = 16$. Use iteration to calculate the result. Store the result in r0.

```
Main:

mov r1, #2

mov r2, #4


Loop:


End:
```

Complete the code. See the PowerPoint slides of week 4.

Screenshot of the completed code here.



Ready? Save this file and export it as a pdf file with the name: **week4.pdf**