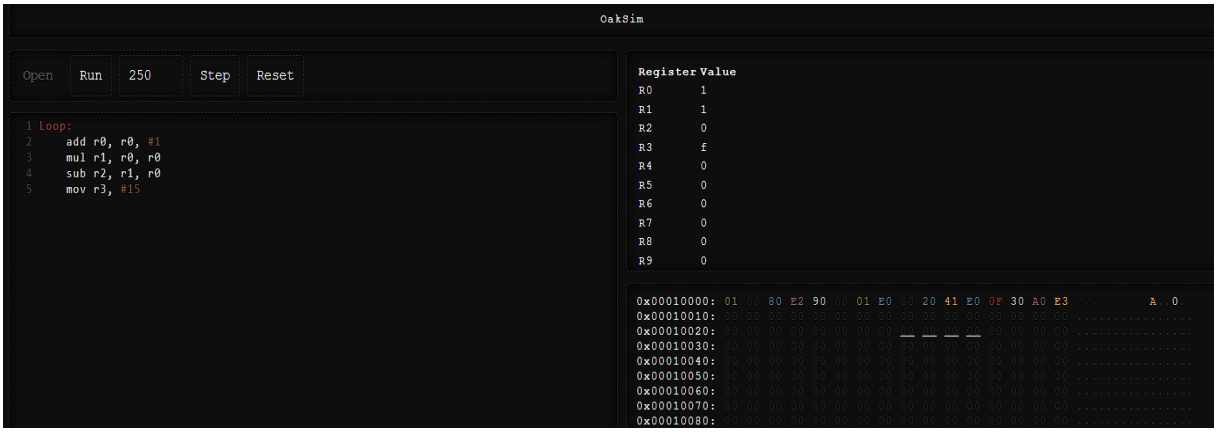
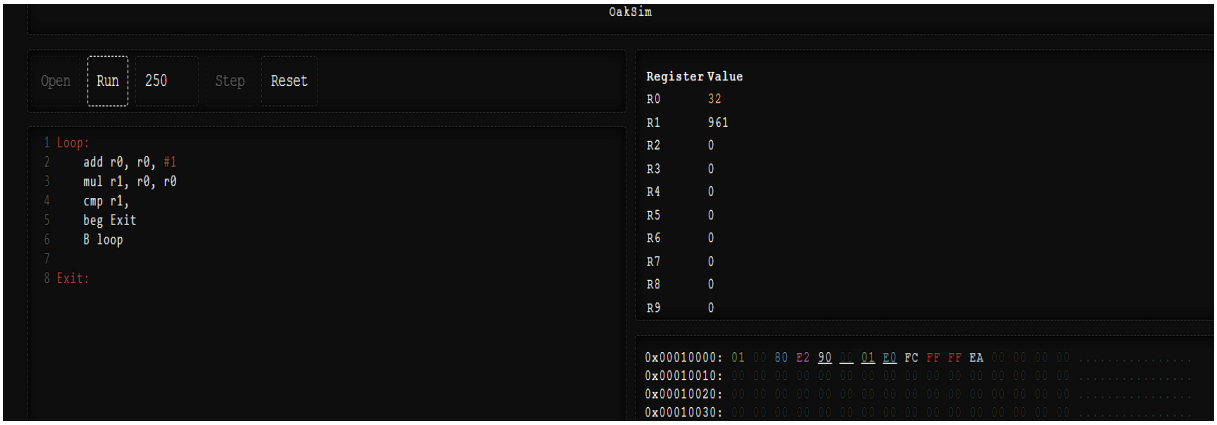
Template Week 4 – Software

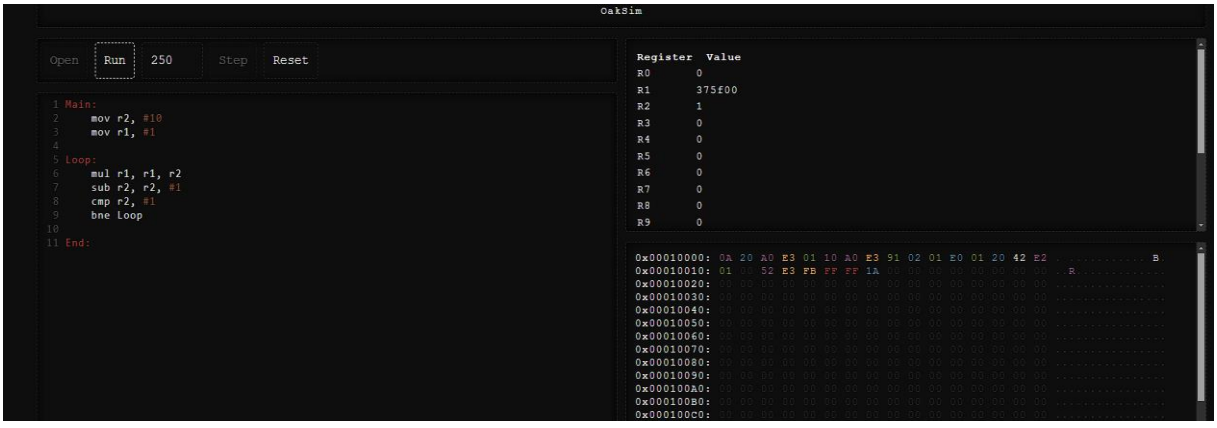
Student number: 569508

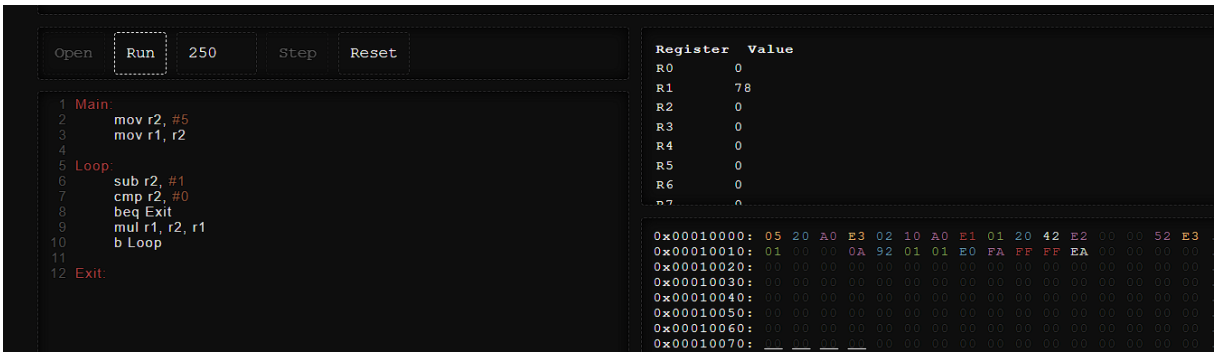
Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:







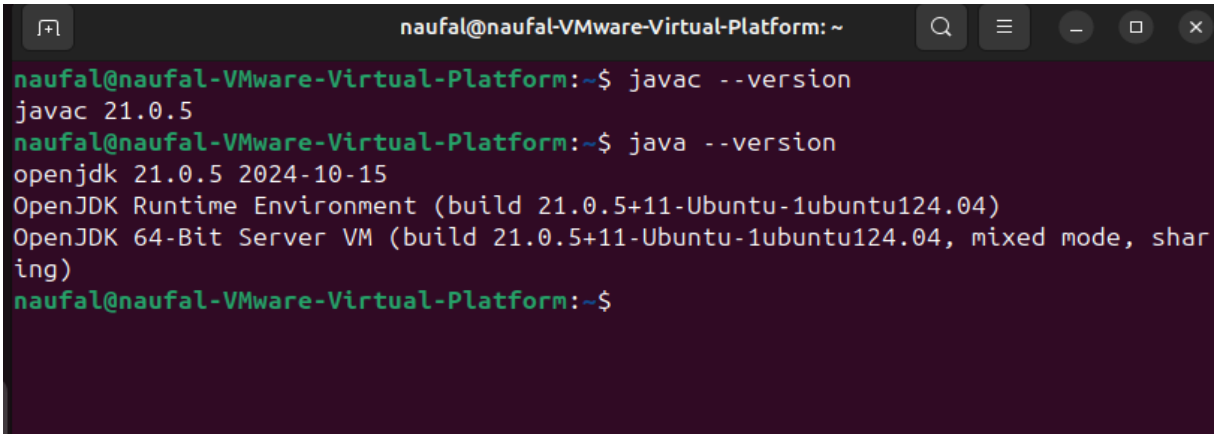


Assignment 4.2: Programming languages

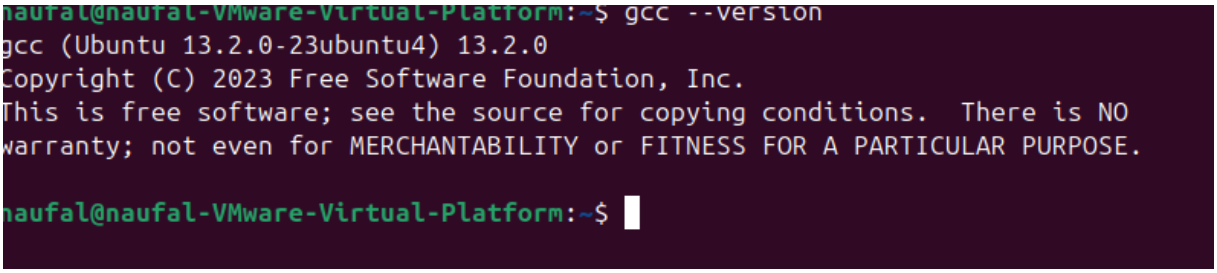
Take screenshots that the following commands work:

javac –version

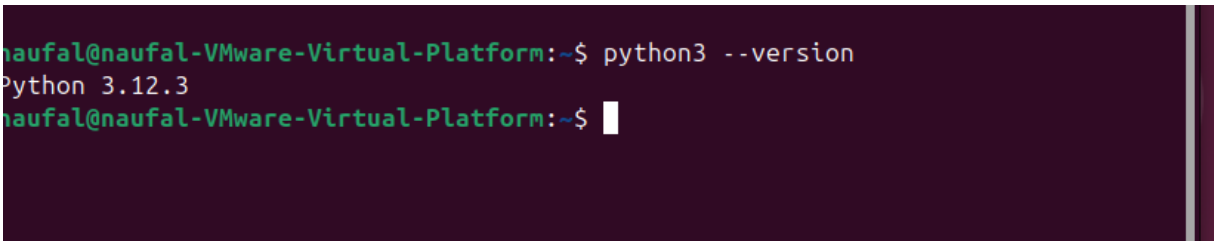
java –version



gcc –version



python3 –version



bash --version

Assignment 4.3: Compile

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

fib.c and Fibonacci.java need to be compiled.

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

fib.c

Which source code files are compiled to byte code?

fibonacci.java

Which source code files are interpreted by an interpreter?

fib.py and fib.sh

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

fib.c

How do I run a Java program?

javac Fibonacci.java then java Fibonacci

How do I run a Python program?

python3 fib.py

How do I run a C program?

gcc fib.c -o fib then ./fib

How do I run a Bash script?

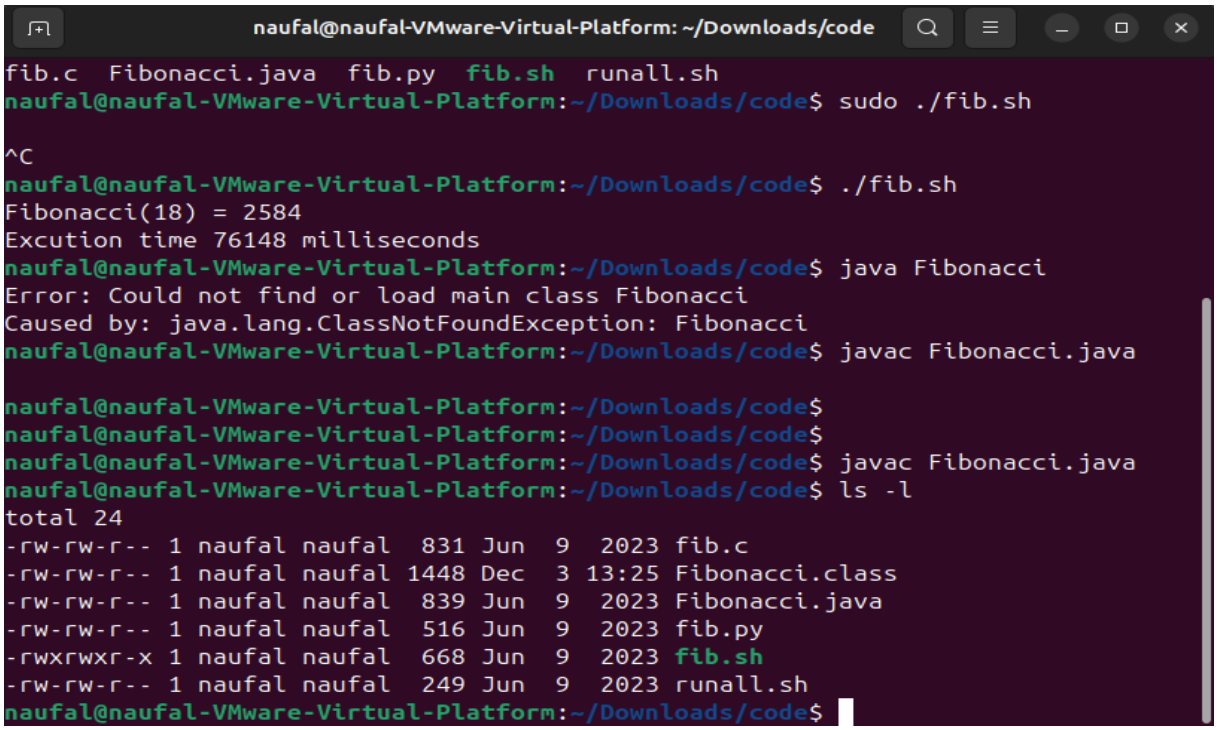
bash fib.sh

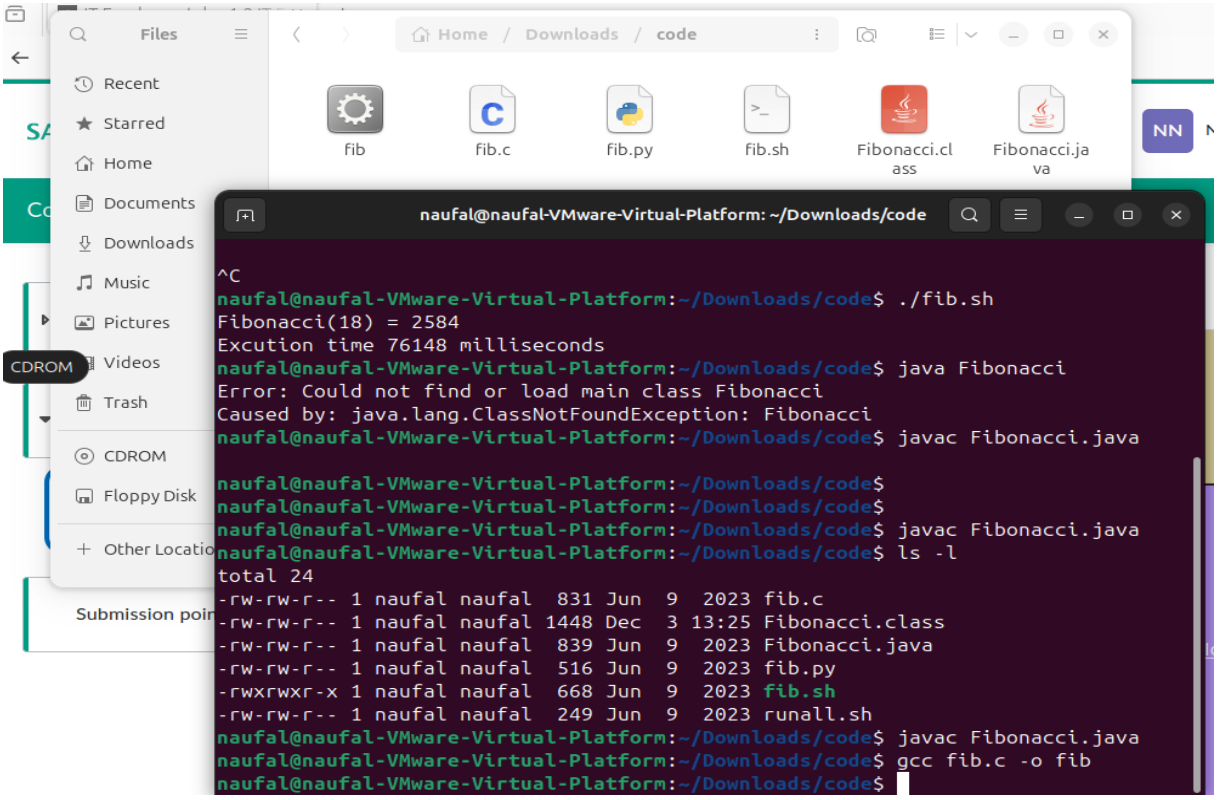
If I compile the above source code, will a new file be created? If so, which file?

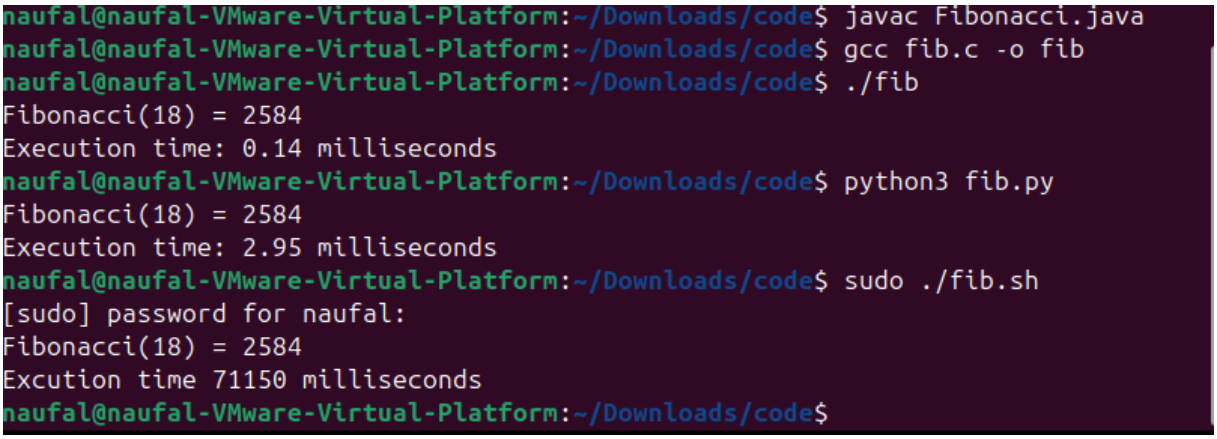
Fib, fib.exe, Fibonacci.class fib.py and fib.sh

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?


naufal@naufal-VMware-Virtual-Platform: ~/Downloads/code
fib.c Fibonacci.java fib.py fib.sh runall.sh
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ sudo ./fib.sh
^C
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ ./fib.sh
Fibonacci(18) = 2584
Excution time 76148 milliseconds
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ java Fibonacci
Error: Could not find or load main class Fibonacci
Caused by: java.lang.ClassNotFoundException: Fibonacci
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ ls -l
total 24
-rw-rw-r-- 1 naufal naufal 831 Jun 9 2023 fib.c
-rw-rw-r-- 1 naufal naufal 1448 Dec 3 13:25 Fibonacci.class
-rw-rw-r-- 1 naufal naufal 839 Jun 9 2023 Fibonacci.java
-rw-rw-r-- 1 naufal naufal 516 Jun 9 2023 fib.py
-rwxrwxr-x 1 naufal naufal 668 Jun 9 2023 fib.sh
-rw-rw-r-- 1 naufal naufal 249 Jun 9 2023 runall.sh
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$

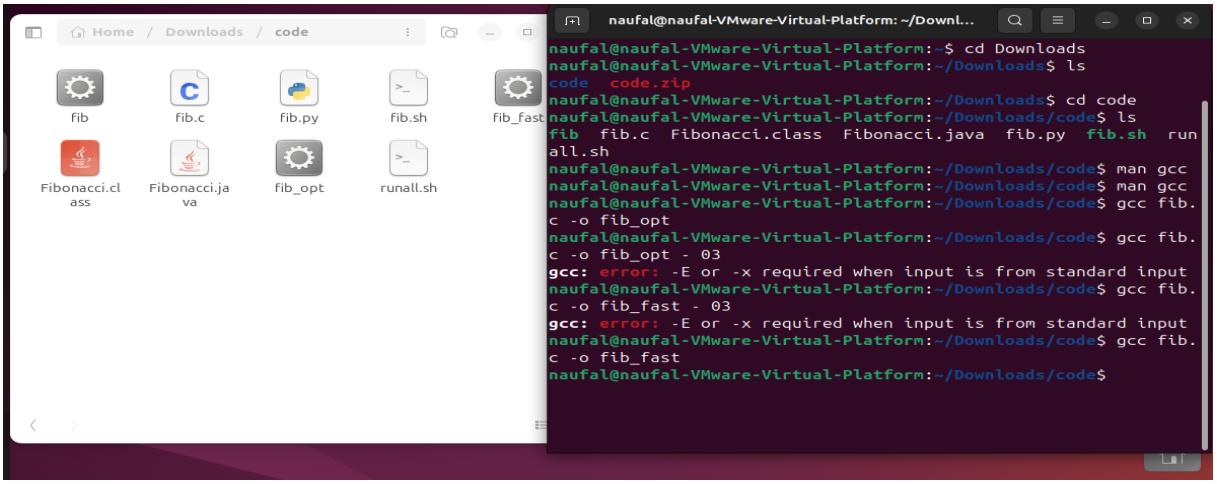

Files
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
CDROM
Floppy Disk
Other Locatio
Home / Downloads / code
fib
fib.c
fib.py
fib.sh
Fibonacci.class
Fibonacci.java
CDROM
Submission poir
naufal@naufal-VMware-Virtual-Platform: ~/Downloads/code
^C
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ ./fib.sh
Fibonacci(18) = 2584
Excution time 76148 milliseconds
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ java Fibonacci
Error: Could not find or load main class Fibonacci
Caused by: java.lang.ClassNotFoundException: Fibonacci
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ ls -l
total 24
-rw-rw-r-- 1 naufal naufal 831 Jun 9 2023 fib.c
-rw-rw-r-- 1 naufal naufal 1448 Dec 3 13:25 Fibonacci.class
-rw-rw-r-- 1 naufal naufal 839 Jun 9 2023 Fibonacci.java
-rw-rw-r-- 1 naufal naufal 516 Jun 9 2023 fib.py
-rwxrwxr-x 1 naufal naufal 668 Jun 9 2023 fib.sh
-rw-rw-r-- 1 naufal naufal 249 Jun 9 2023 runall.sh
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ gcc fib.c -o fib
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$


naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ gcc fib.c -o fib
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ ./fib
Fibonacci(18) = 2584
Execution time: 0.14 milliseconds
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 2.95 milliseconds
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$ sudo ./fib.sh
[sudo] password for naufal:
Fibonacci(18) = 2584
Excution time 71150 milliseconds
naufal@naufal-VMware-Virtual-Platform:~/Downloads/code$

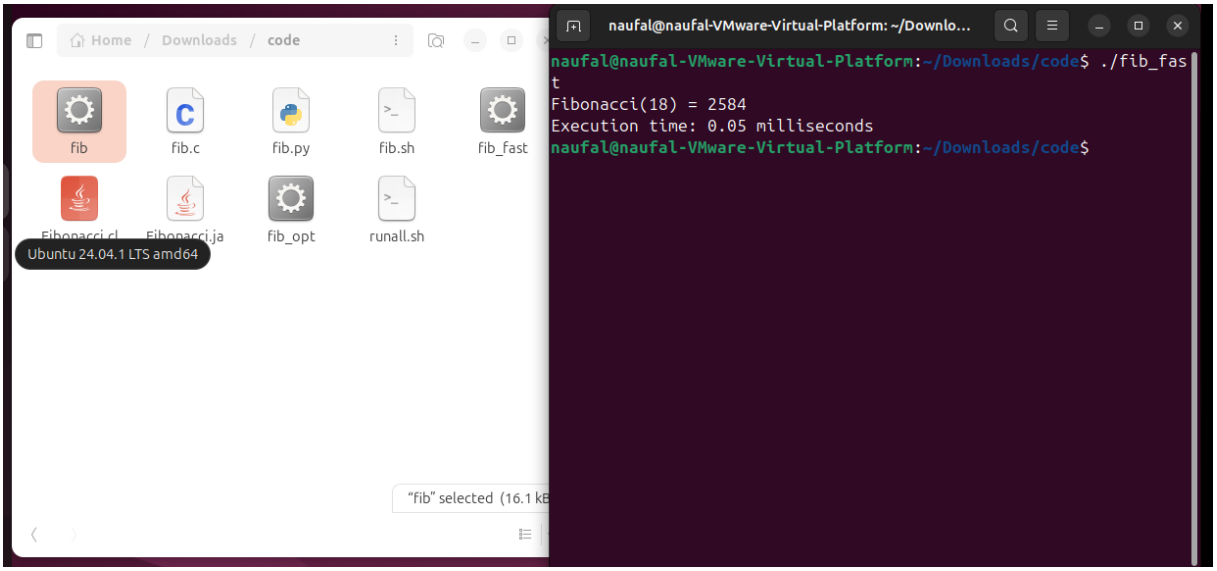
Assignment 4.4: Optimize

Take relevant screenshots of the following commands:

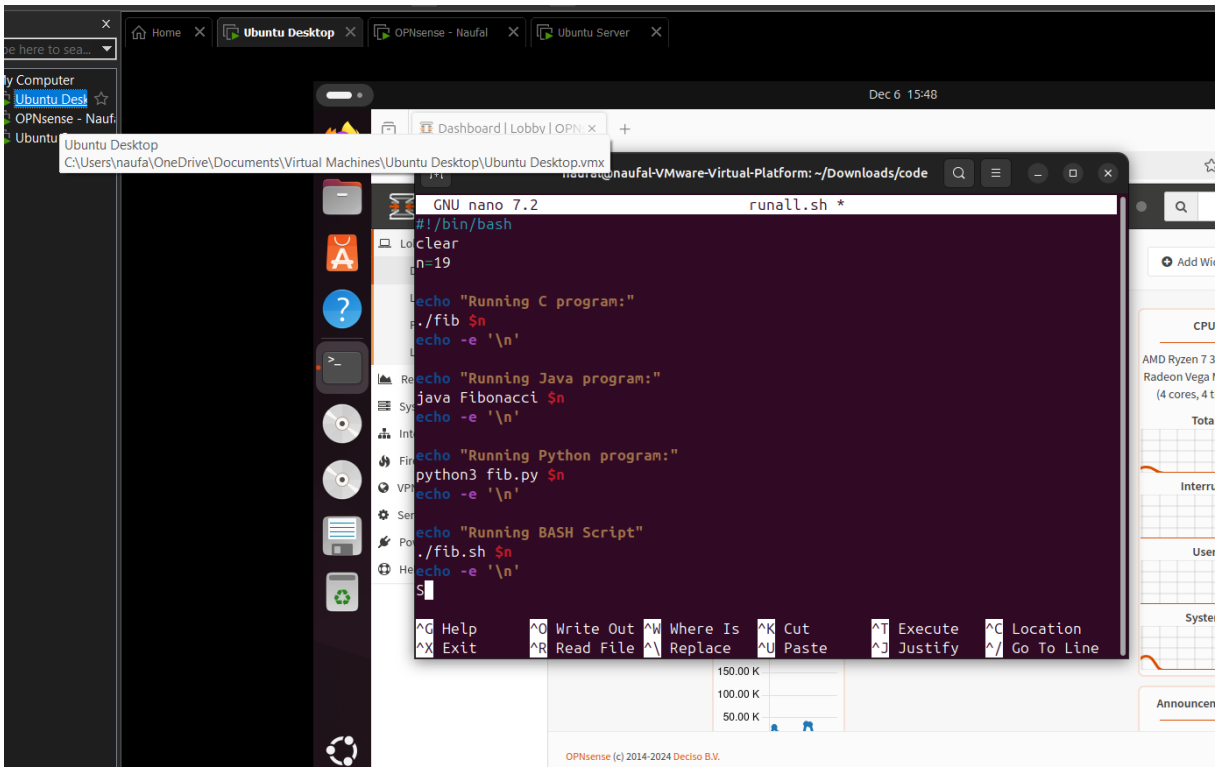
a) Figure out which parameters you need to pass to **the gcc** compiler so that the compiler performs a number of optimizations that will ensure that the compiled source code will run faster. **Tip!** The parameters are usually a letter followed by a number. Also read **page 191** of your book, but find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



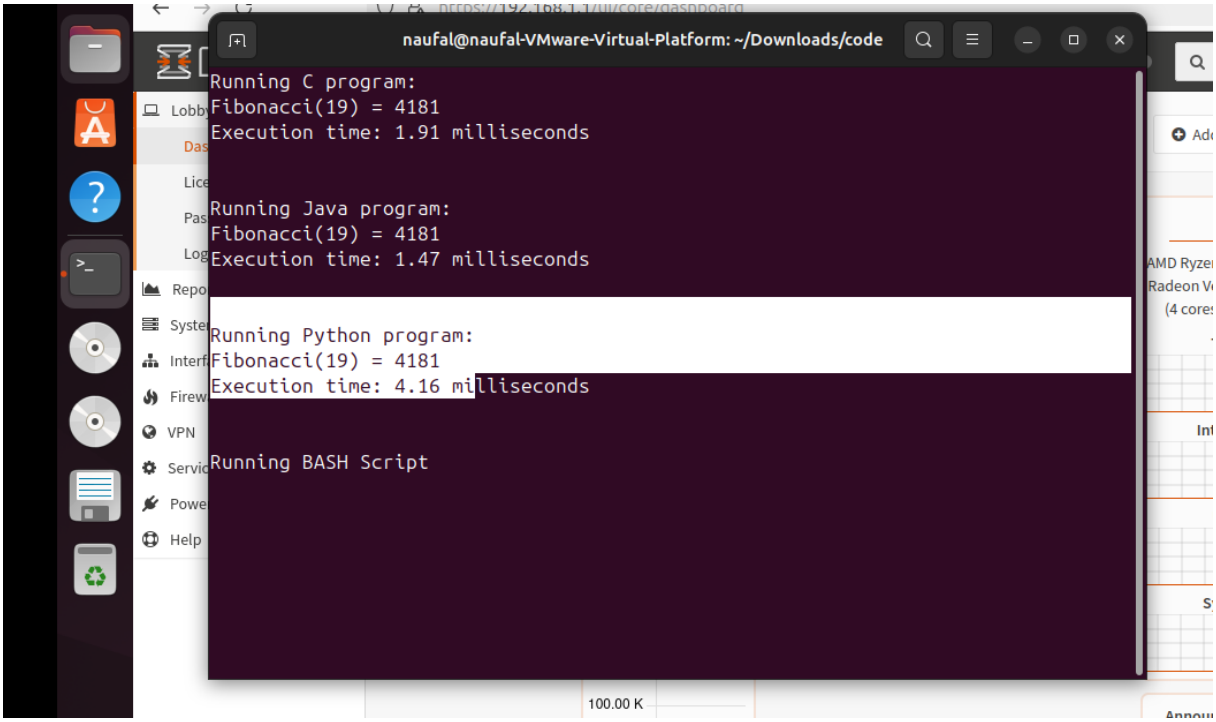
b) Compile **fib.c** again with the optimization parameters



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.



Bonus point assignment – week 4

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.

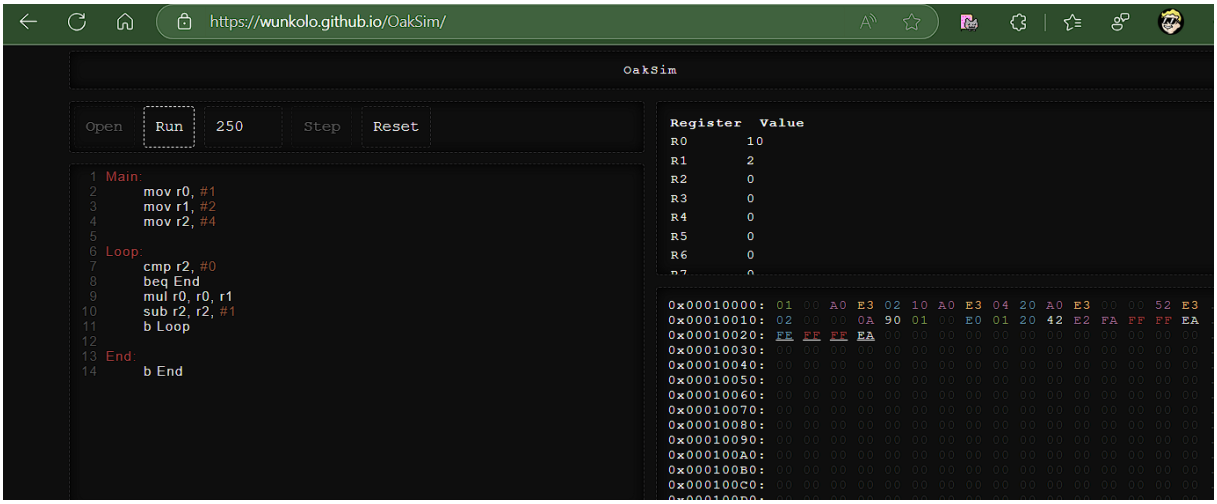
```
Main:
mov r1, #2
mov r2, #4

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**