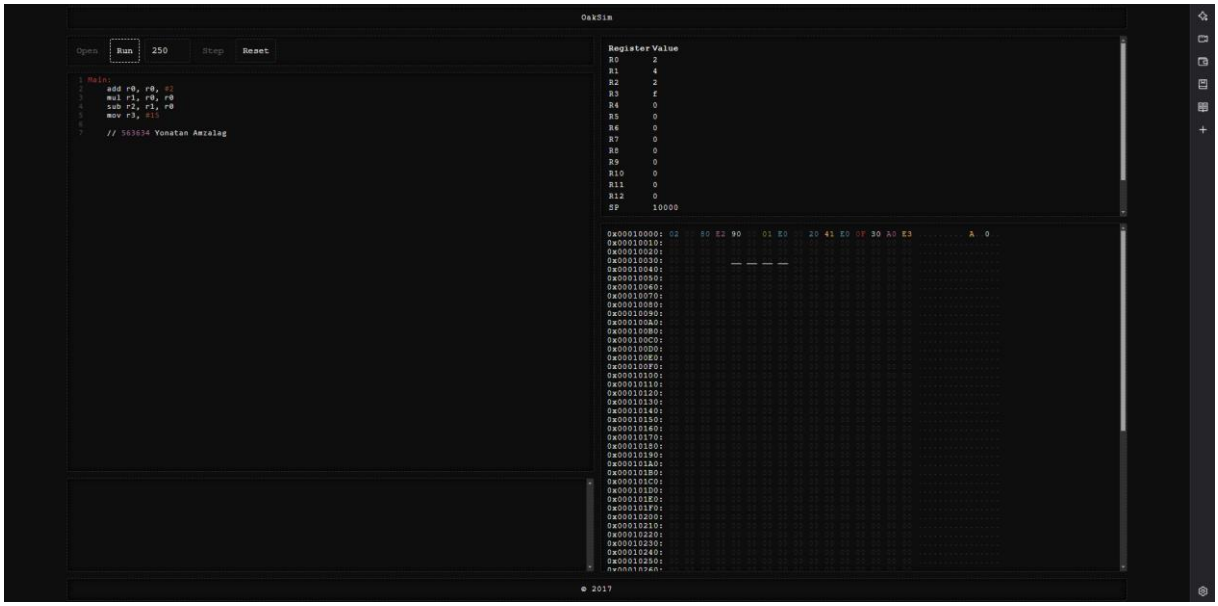
Template Week 4 – Software

Student number: 563634

Assignment 4.1: ARM assembly

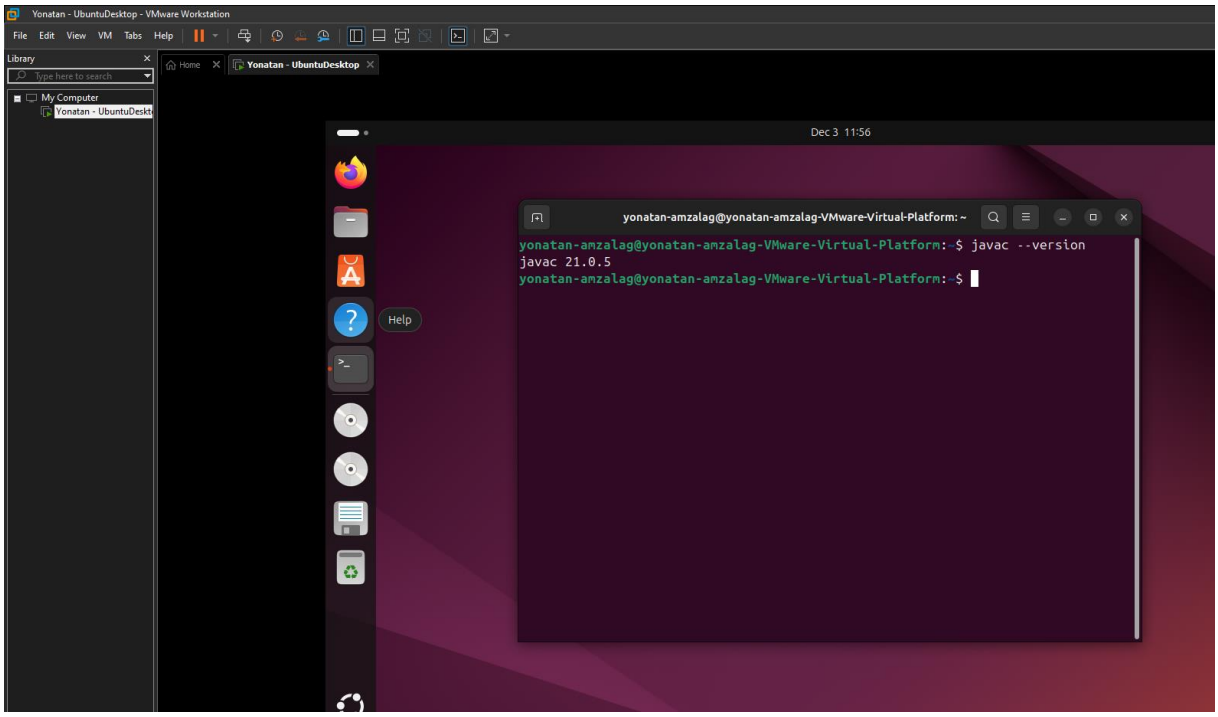
Screenshot of working assembly code of factorial calculation:



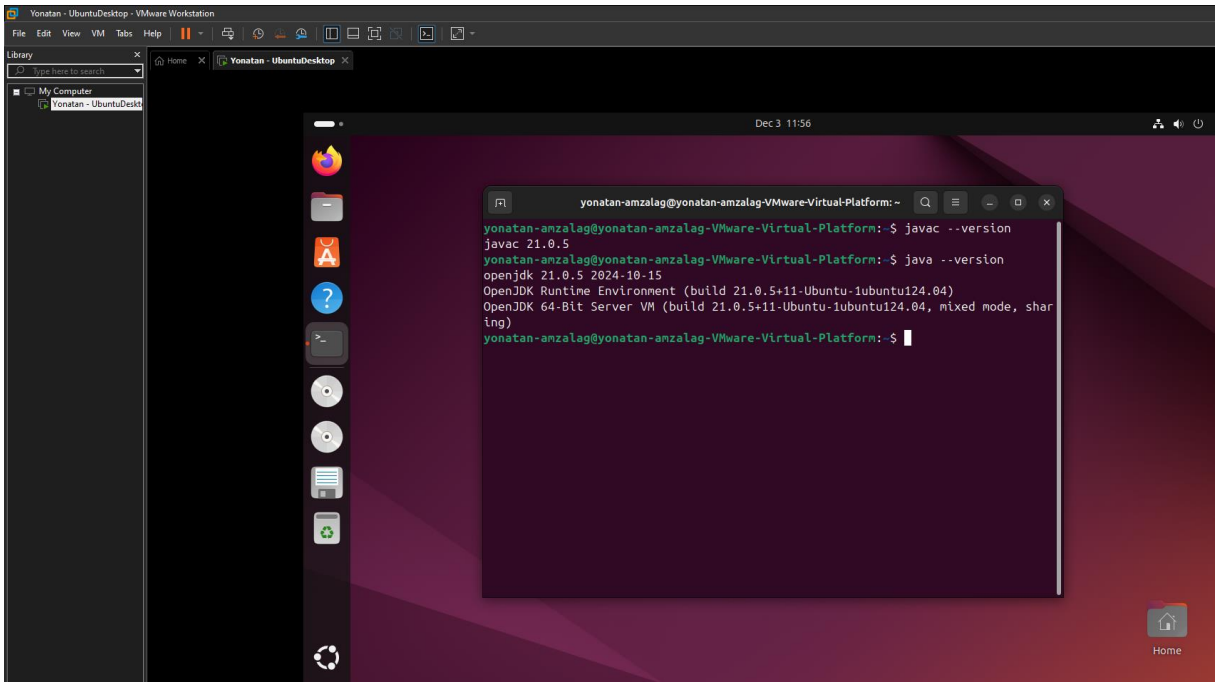
Assignment 4.2: Programming languages

Take screenshots that the following commands work:

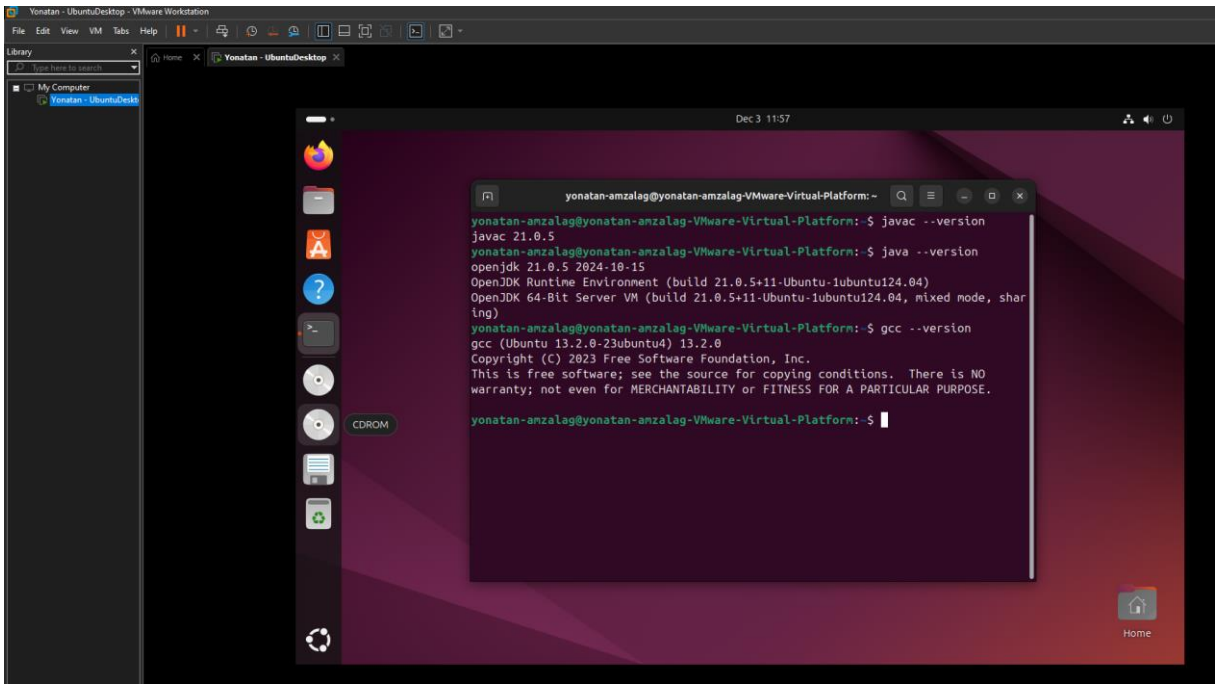
javac –version



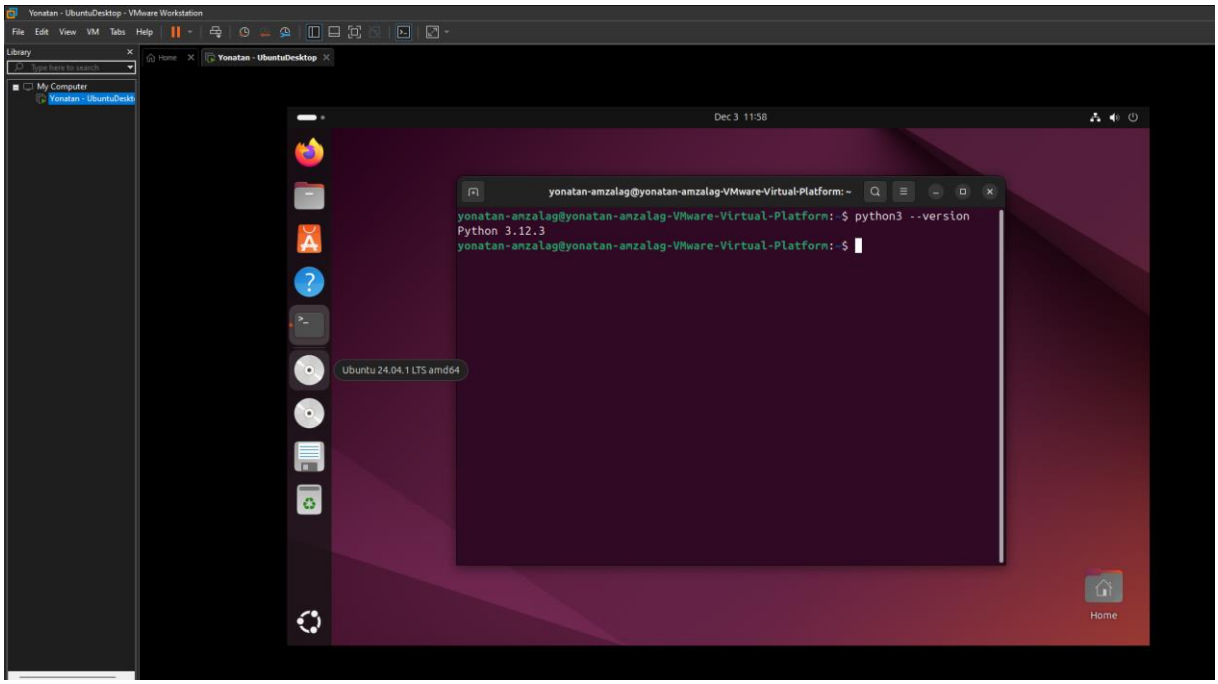
java –version



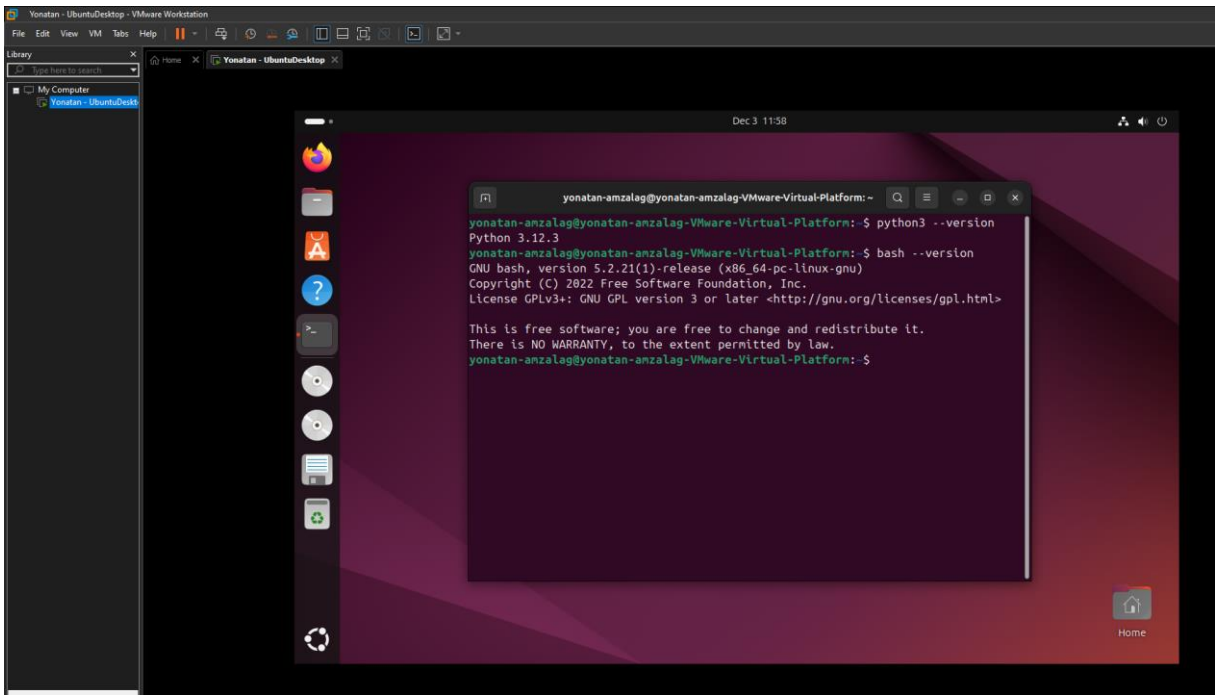
gcc –version



python3 –version



bash --version



Assignment 4.3: Compile

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

Fibonacci.java and fib.c

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

The fib.c source code and using a compiler like gcc.

Which source code files are compiled to byte code?

The Fibonacci.java source code, using the compiler javac.

Which source code files are interpreted by an interpreter?

Fib.py interpreted by python3 interpreter, and fib.sh interpreted by Bash shell.

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

Fib.c, it´s a compilled C code that is executed instantly by the processor.

How do I run a Java program?

We need to compile the program first, and then run the program using the Java interpreter (javac)

How do I run a Python program?

We use the python3 interpreter.

How do I run a C program?

We can use a compiler like gcc to compile it, and then execute it.

How do I run a Bash script?

Run the script.

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

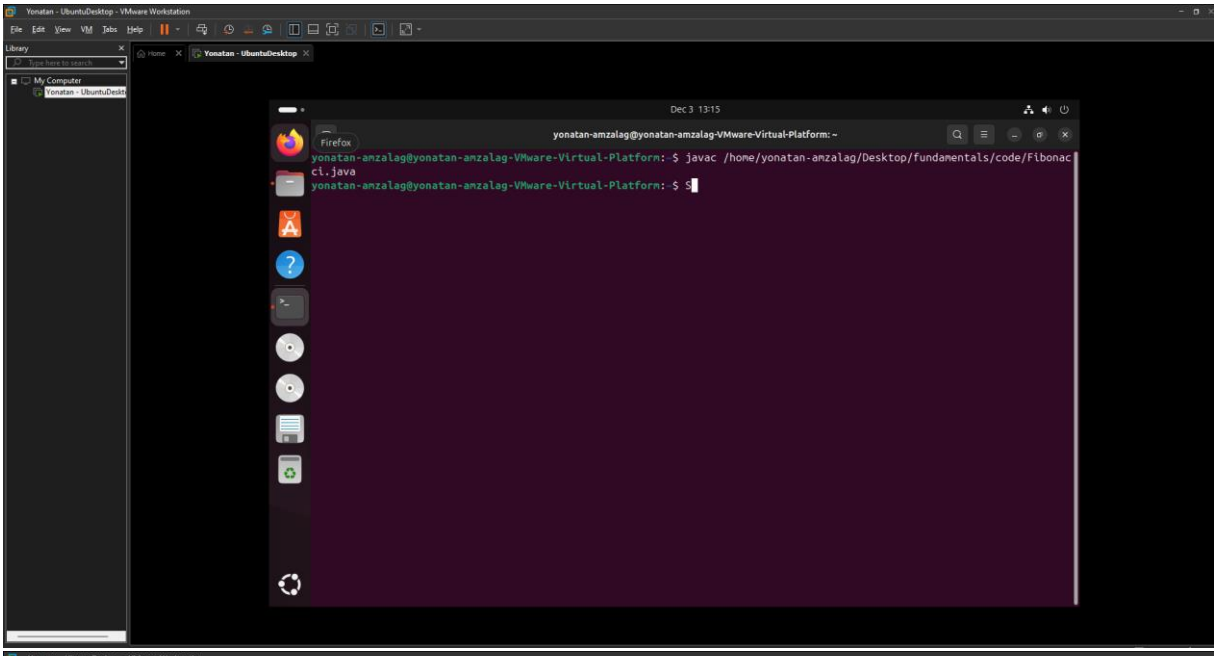
When we run the Fibonacci.java file, a Fibonacci.class file will be created.

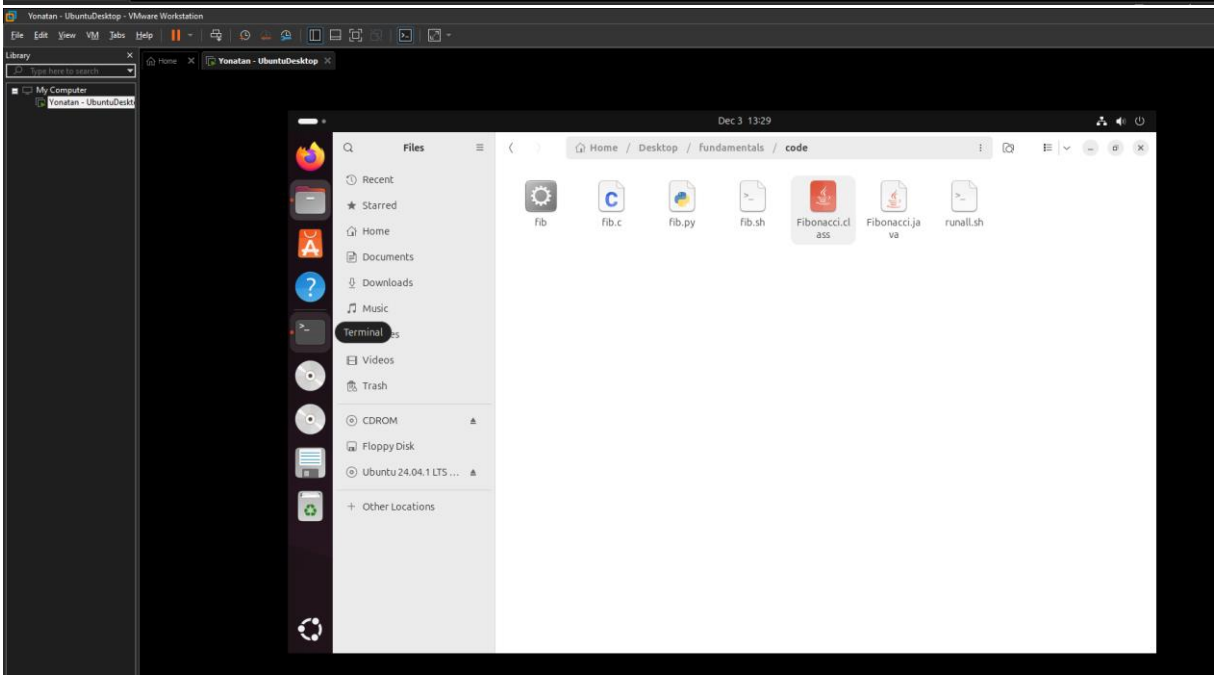
When we run the fib.c file, an executable file will be created.

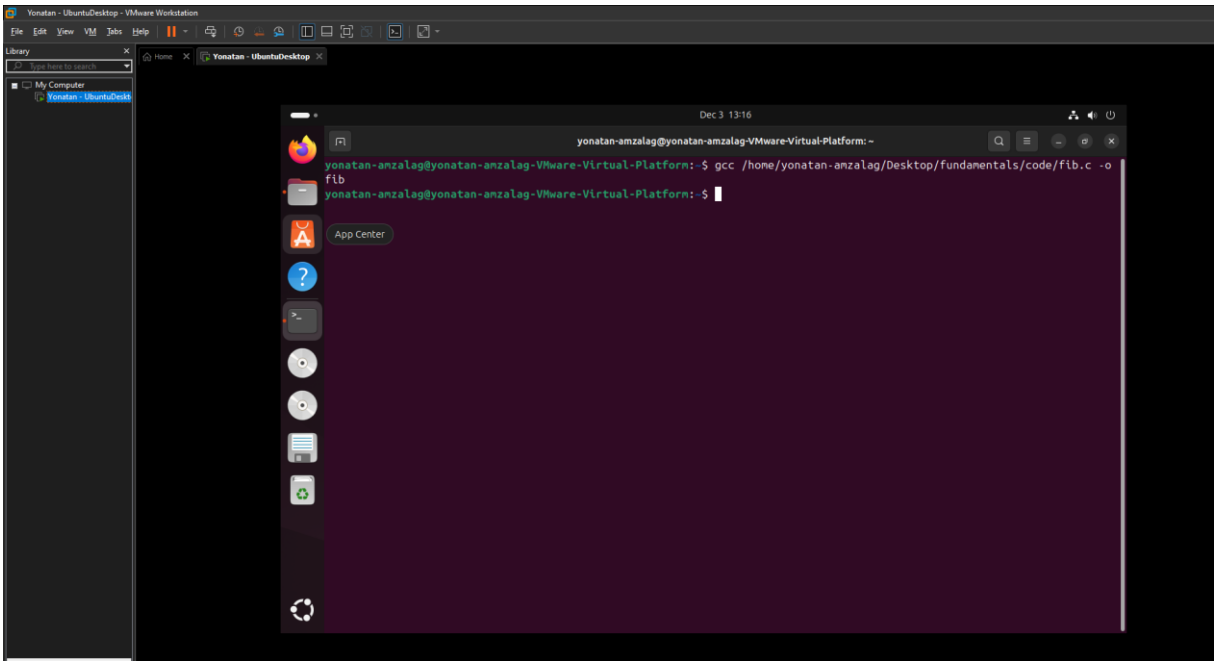
When we use python and bash, no new file will be created.

Take relevant screenshots of the following commands:

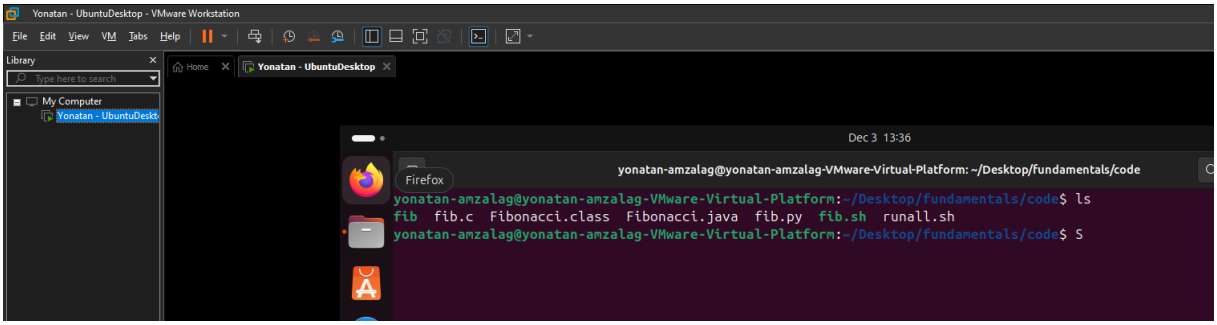
- Compile the source files where necessary



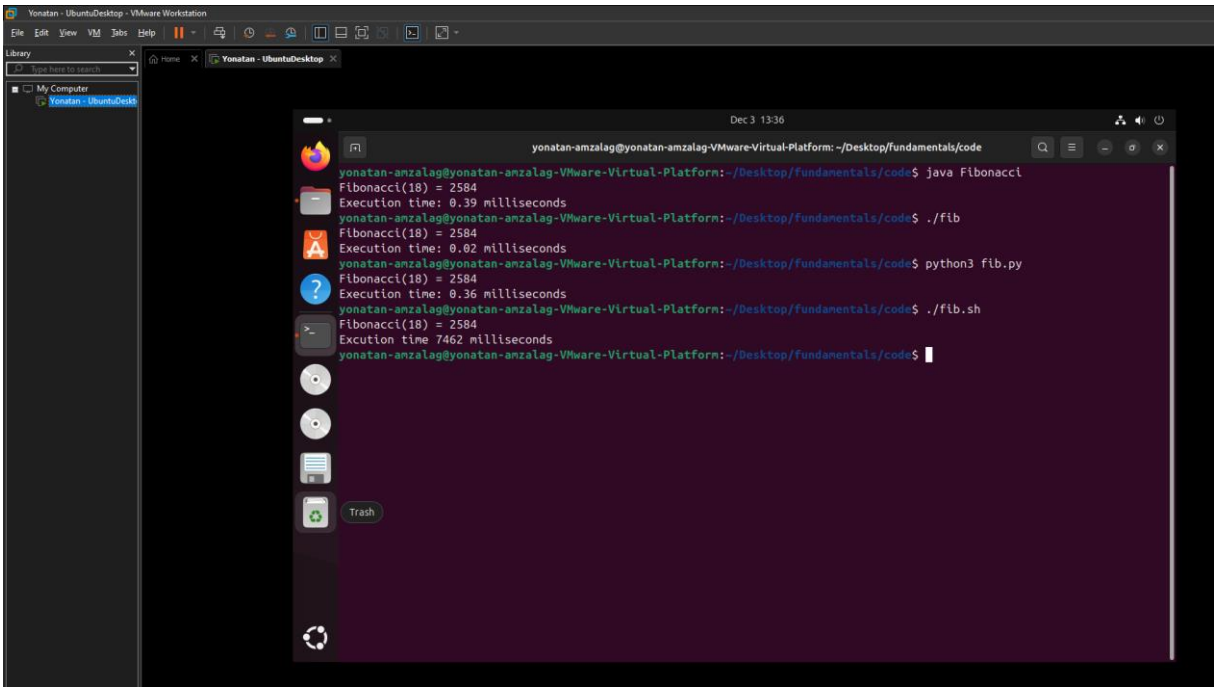




- Make them executable



- Run them

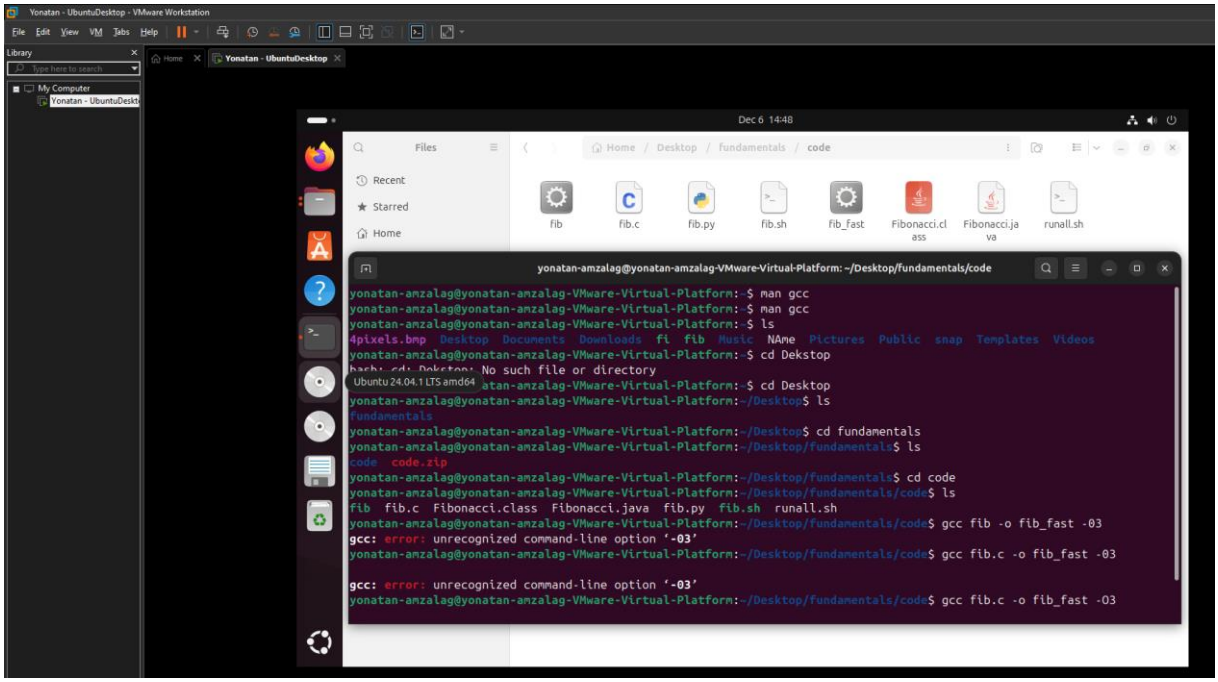


- Which (compiled) source code file performs the calculation the fastest?
  The fastest is C, because it´s a compiled machine code.

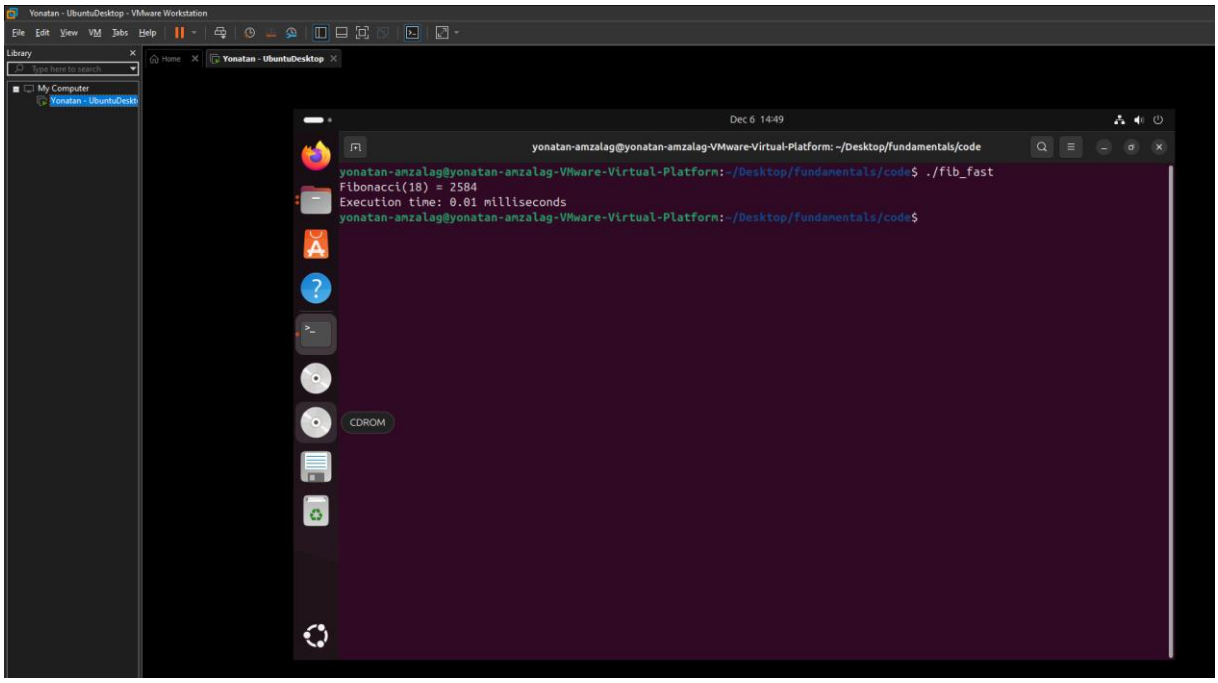
Assignment 4.4: Optimize

Take relevant screenshots of the following commands:

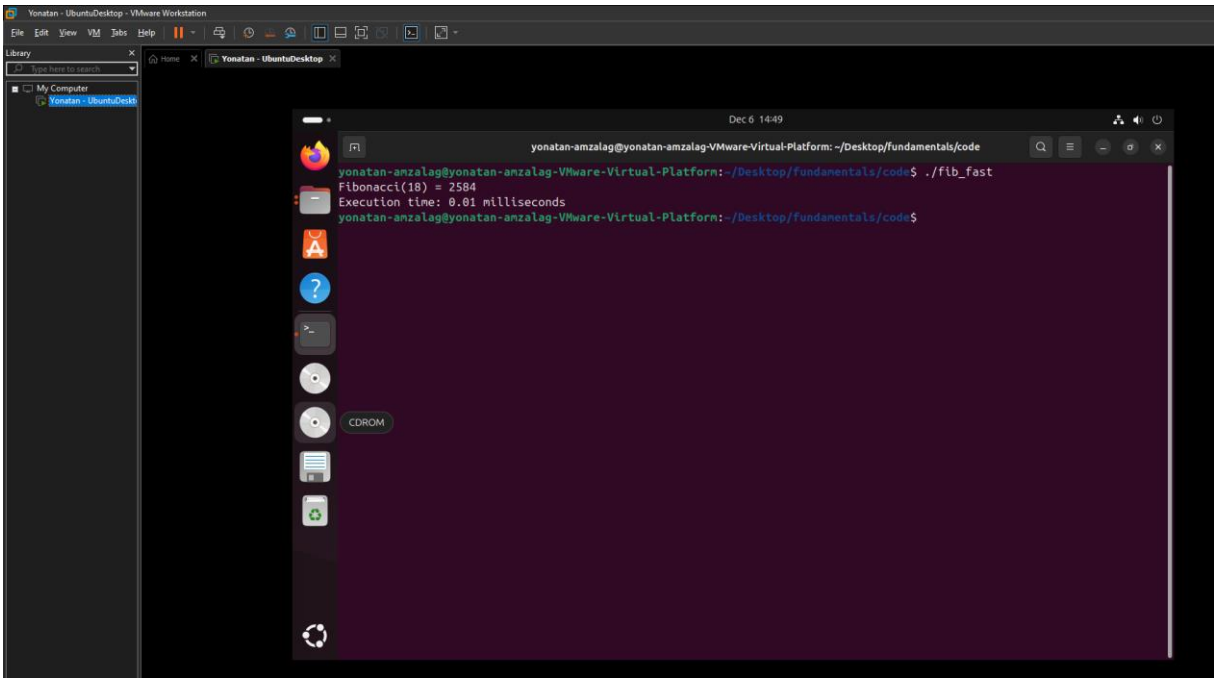
a) Figure out which parameters you need to pass to **the gcc** compiler so that the compiler performs a number of optimizations that will ensure that the compiled source code will run faster. **Tip!** The parameters are usually a letter followed by a number. Also read **page 191** of your book, but find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



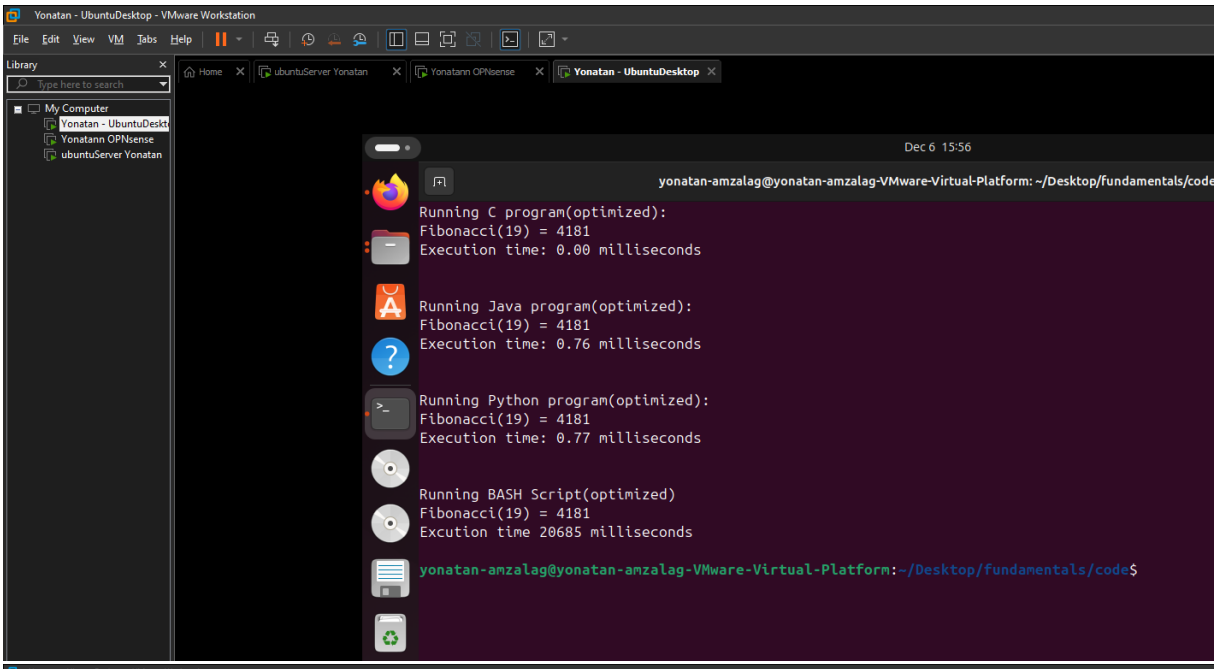
b) Compile **fib.c** again with the optimization parameters

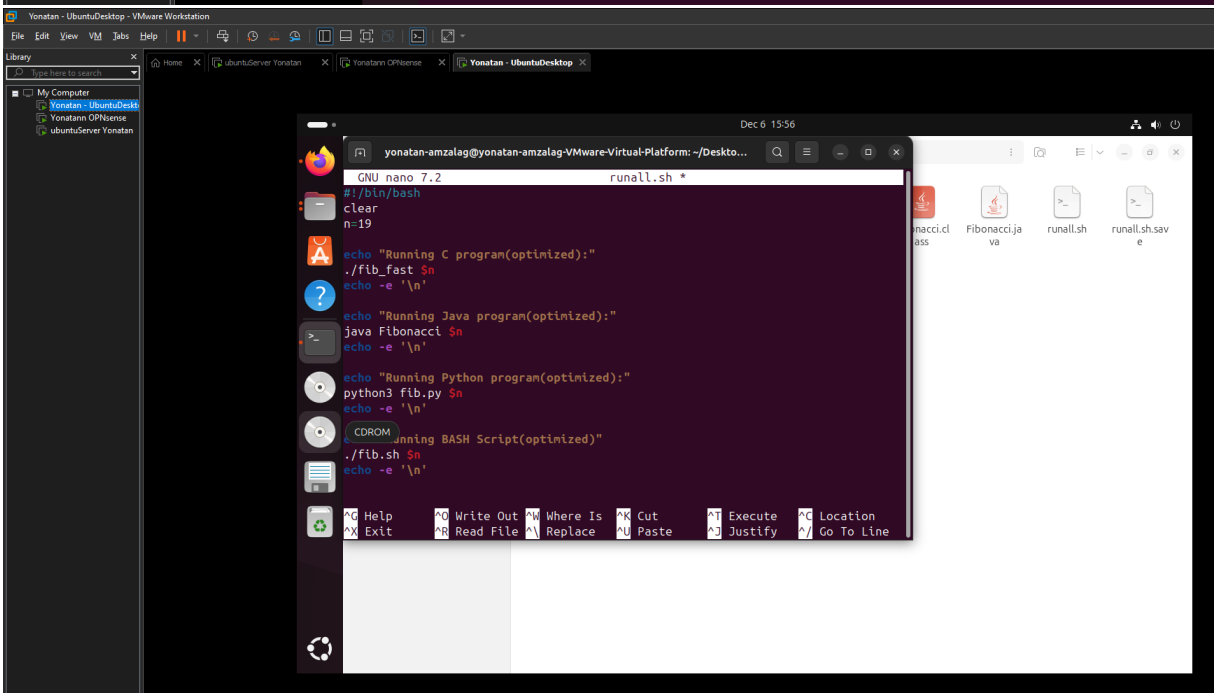


c) Run the newly compiled program. Is it true that it now performs the calculation faster?



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.

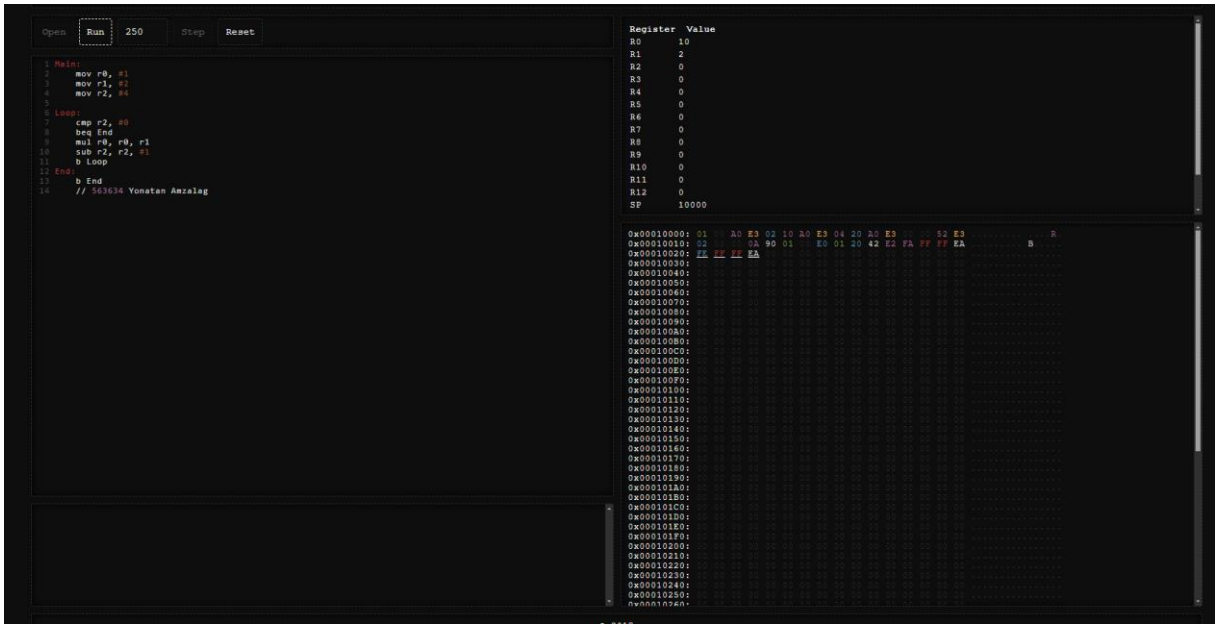
```
Main:
mov r1, #2
mov r2, #4
```

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**