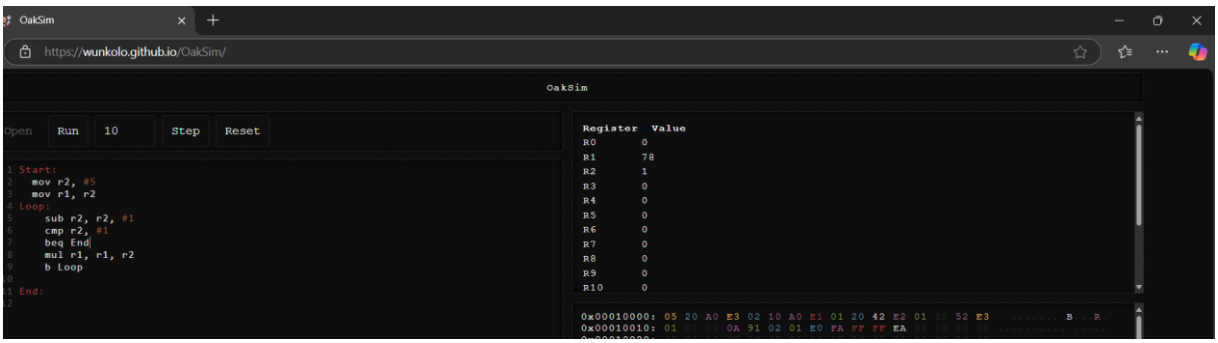
Template Week 4 – Software

Student number: 568209 eilyad

Used chatcpt only for better English writing and getting some information about assignment 4.3

Assignment 4.1: ARM assembly

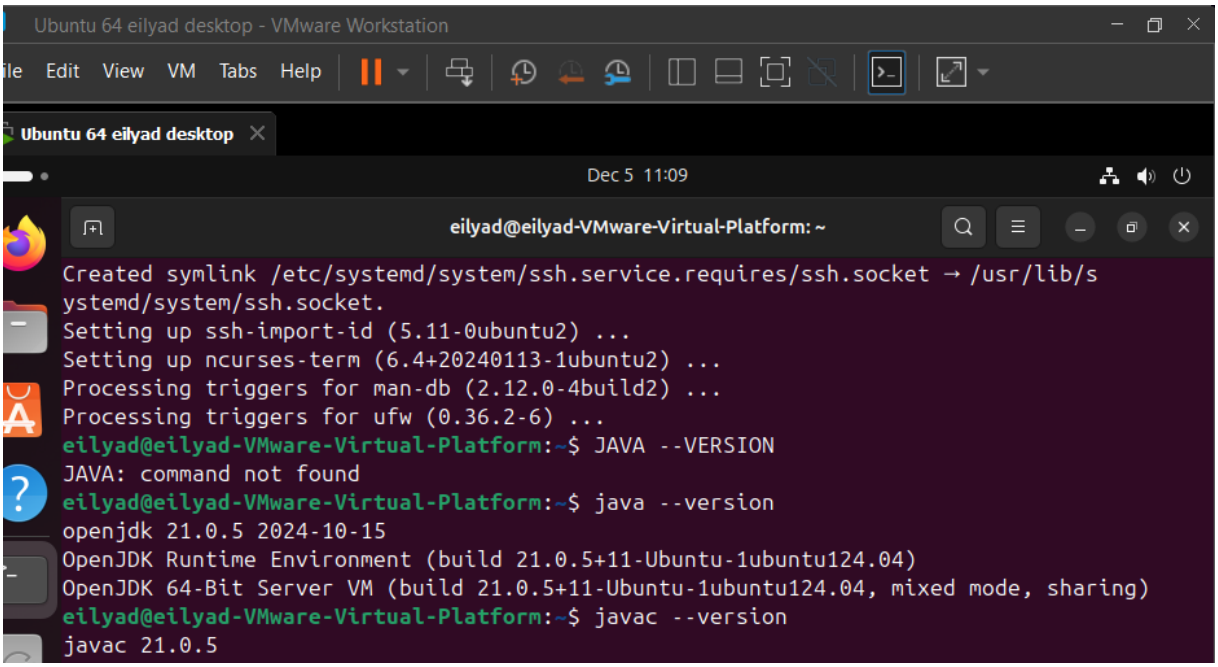
Screenshot of working assembly code of factorial calculation:



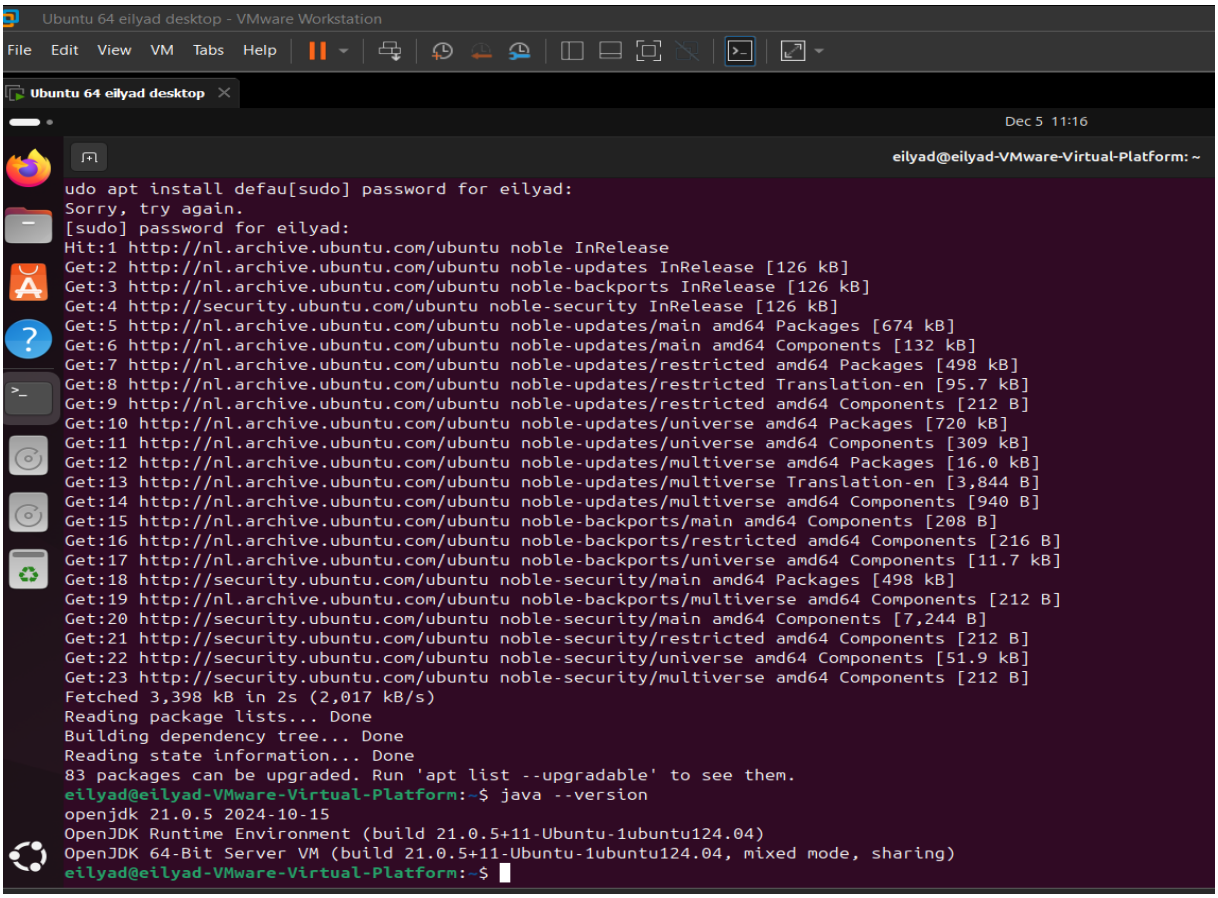
Assignment 4.2: Programming languages

Take screenshots that the following commands work:

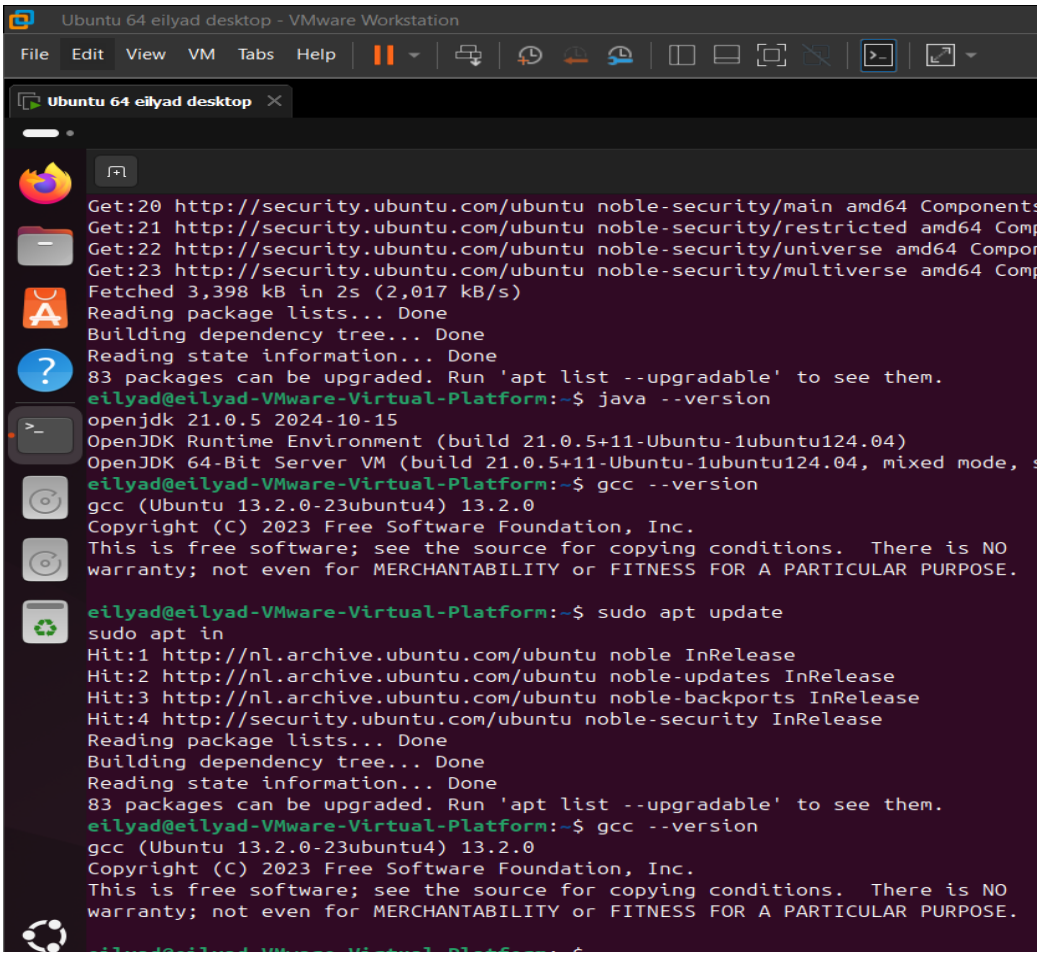
javac –version



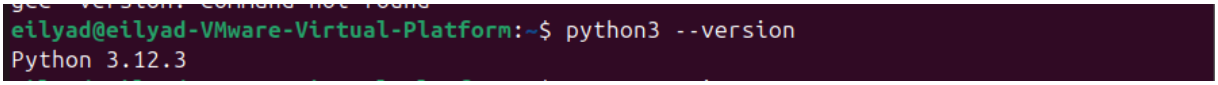
java –version



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 (compiled to bytecode).

Fib.c (compiled to machine code).

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

`fib.c` is compiled into machine code using `gcc`.

Which source code files are compiled to byte code?

`Fibonacci.java` is compiled into bytecode (`Fibonacci.Class`) by the javac command.

Which source code files are interpreted by an interpreter?

Python programs (e.g., `fib.py`) are interpreted by the Python interpreter (`python3`).Bash scripts (e.g., `fib.sh`) are 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 (compiled to machine code) is expected to perform calculations the fastest because it runs directly on the hardware without any intermediate layer like the JVM or an interpreter

How do I run a Java program?

Compile: javac Fibonacci.java

Run: java Fibonacci

How do I run a Python program?

Run the Python script directly using: python3 fib.py

How do I run a C program?

Compile: gcc -o fib Fib.c

Run: ./fib

How do I run a Bash script?

Make the script executable: chmod +x fib.sh

Run: ./fib.sh

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

`Fibonacci.java`

Compiled into: `Fibonacci.class` (bytecode file for the JVM).

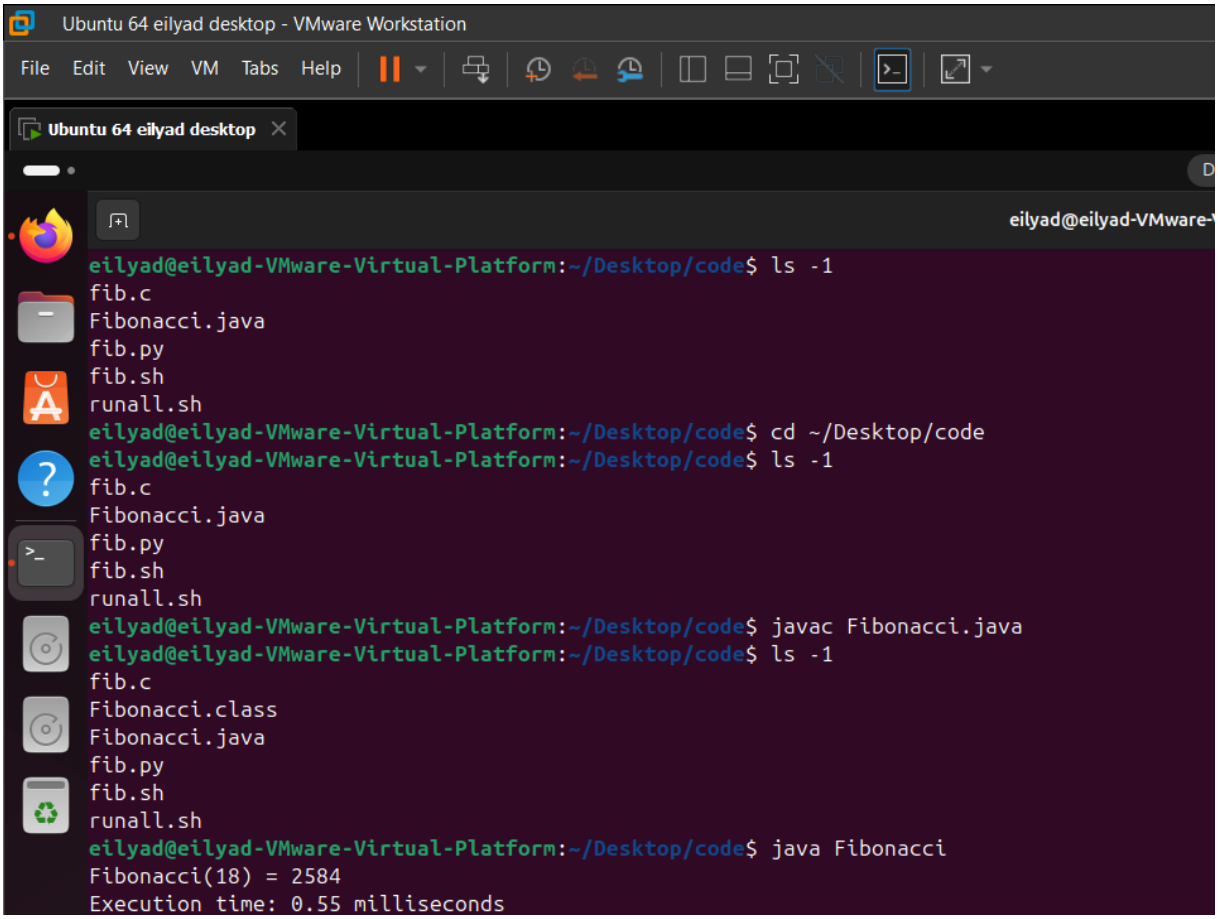
`Fib.c`
Compiled into: `fib` (machine code executable).

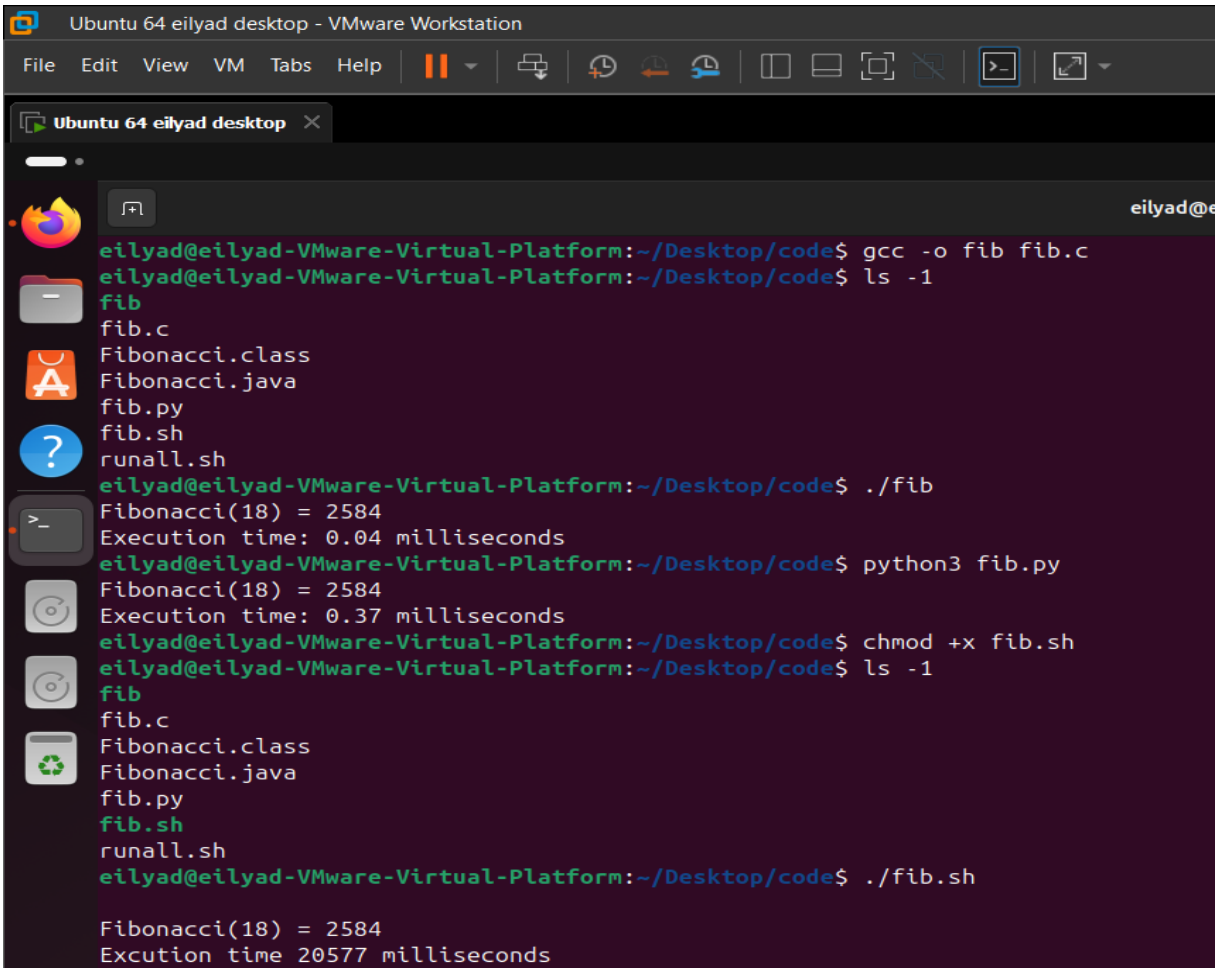
Python and Bash scripts

No new files are created since they are interpreted directly without compilation.

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?

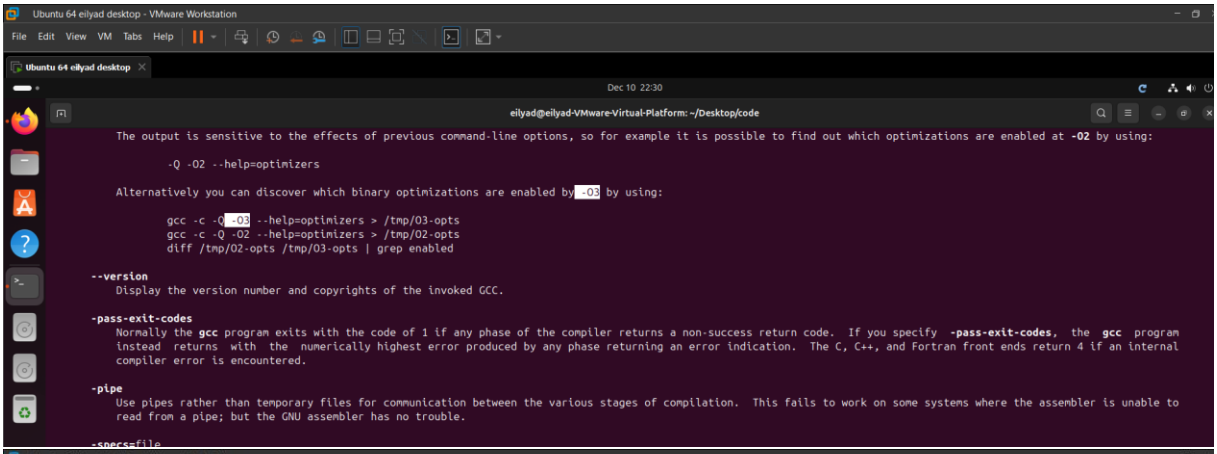


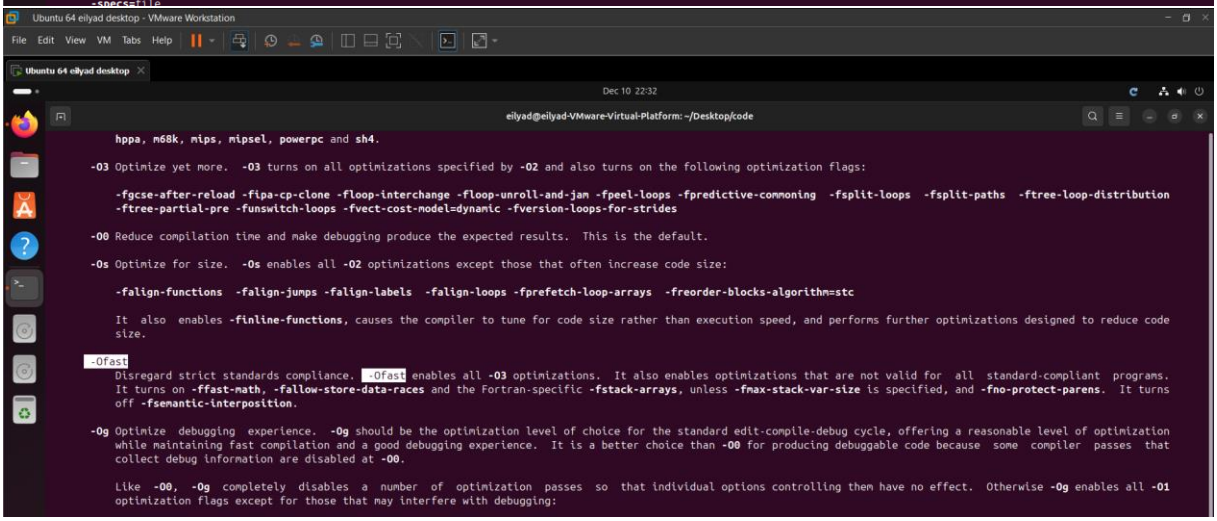

Ubuntu 64 eilyad desktop - VMware Workstation
File Edit View VM Tabs Help
Ubuntu 64 eilyad desktop
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ gcc -o fib fib.c
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ ls -1
fib
fib.c
Fibonacci.class
Fibonacci.java
fib.py
fib.sh
runall.sh
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ ./fib
Fibonacci(18) = 2584
Execution time: 0.04 milliseconds
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.37 milliseconds
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ chmod +x fib.sh
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ ls -1
fib
fib.c
Fibonacci.class
Fibonacci.java
fib.py
fib.sh
runall.sh
eilyad@eilyad-VMware-Virtual-Platform:~/Desktop/code$ ./fib.sh
Fibonacci(18) = 2584
Excution time 20577 milliseconds

Assignment 4.4: Optimize

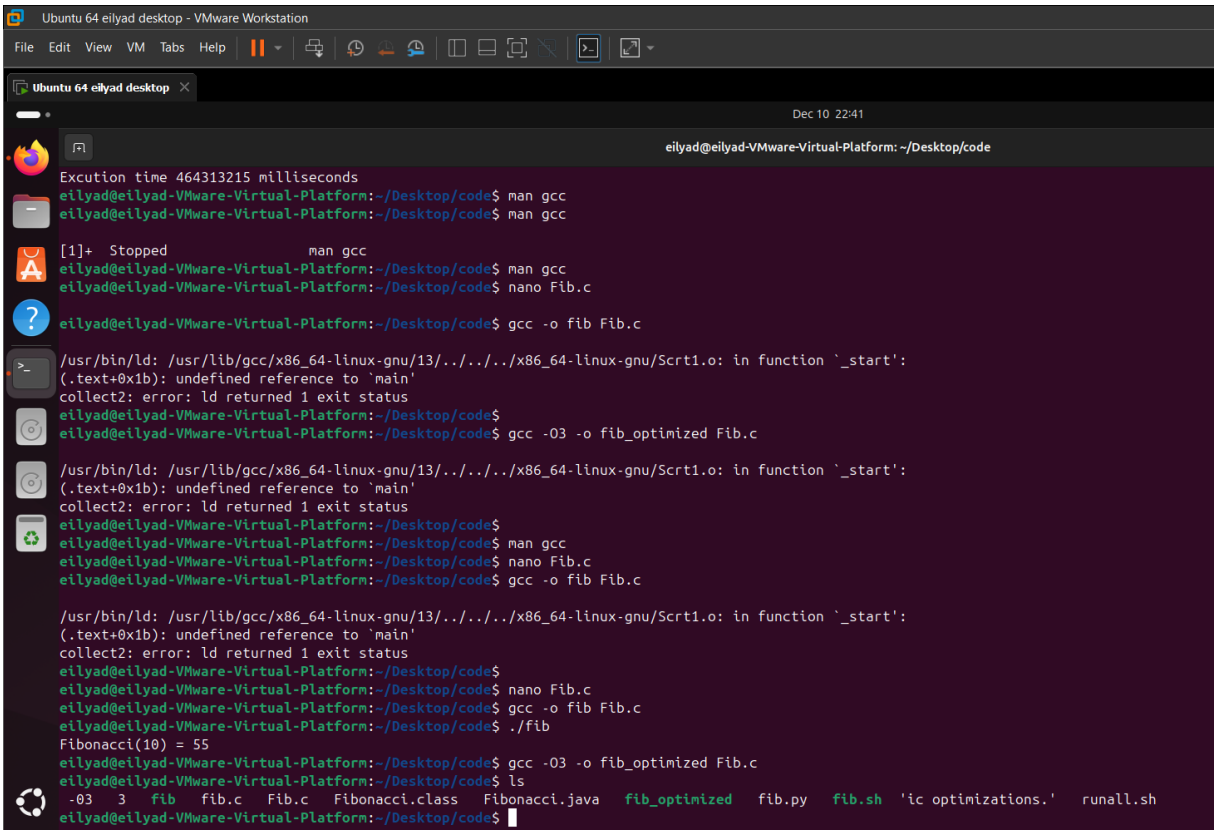
Take relevant screenshots of the following commands:

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.

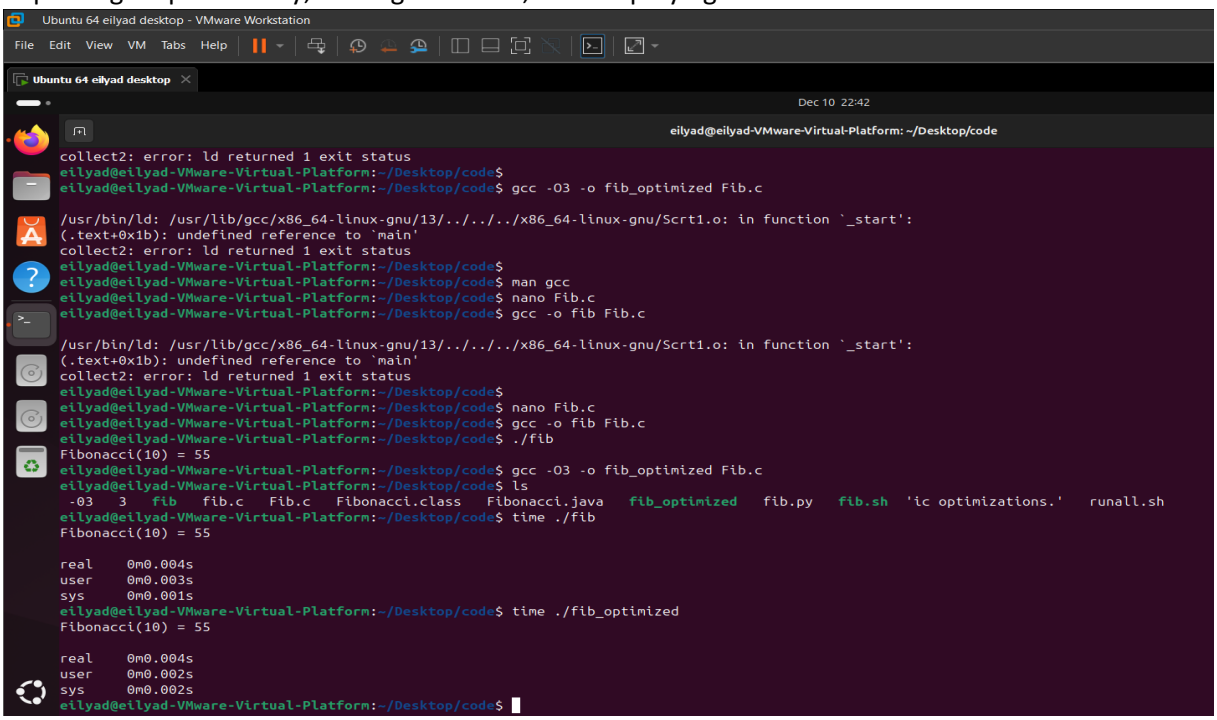




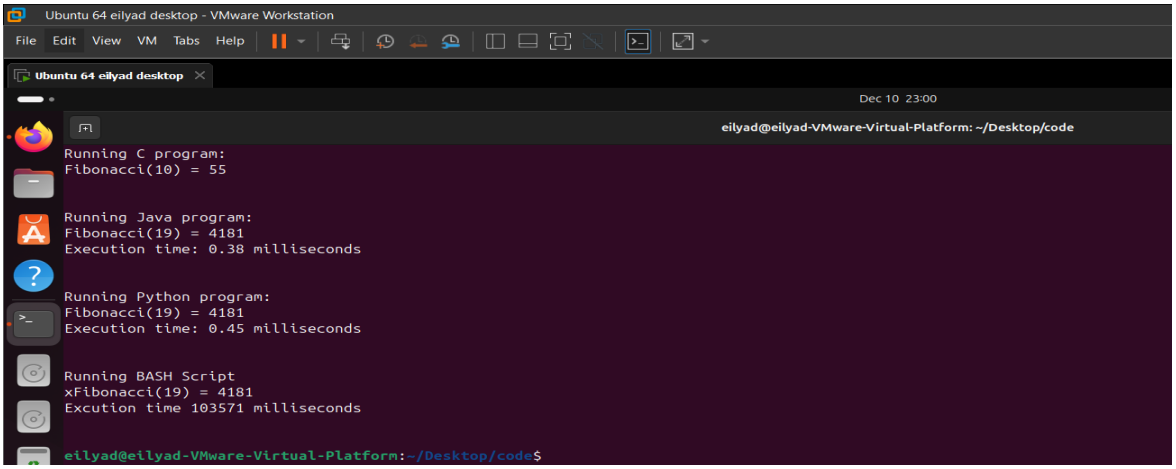
b) Compile **fib.c** again with the optimization parameters



c) Run the newly compiled program. Is it true that it now performs the calculation faster?
The optimized version uses advanced compiler optimizations, reducing execution time by improving loop efficiency, inlining functions, and simplifying calculations.



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.
C: Runs directly on the hardware as compiled machine code → fastest.
Java: Runs on the Java Virtual Machine (JVM), which adds some overhead.
Python: Interpreted language → Slower than Java or C.
Bash: Designed for scripting, not computation → Slowest.
C is the fastest because the code is compiled into native machine code and runs directly on the CPU.
Optimizations like `-O3` make C even faster by improving how loops and functions are handled.



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.

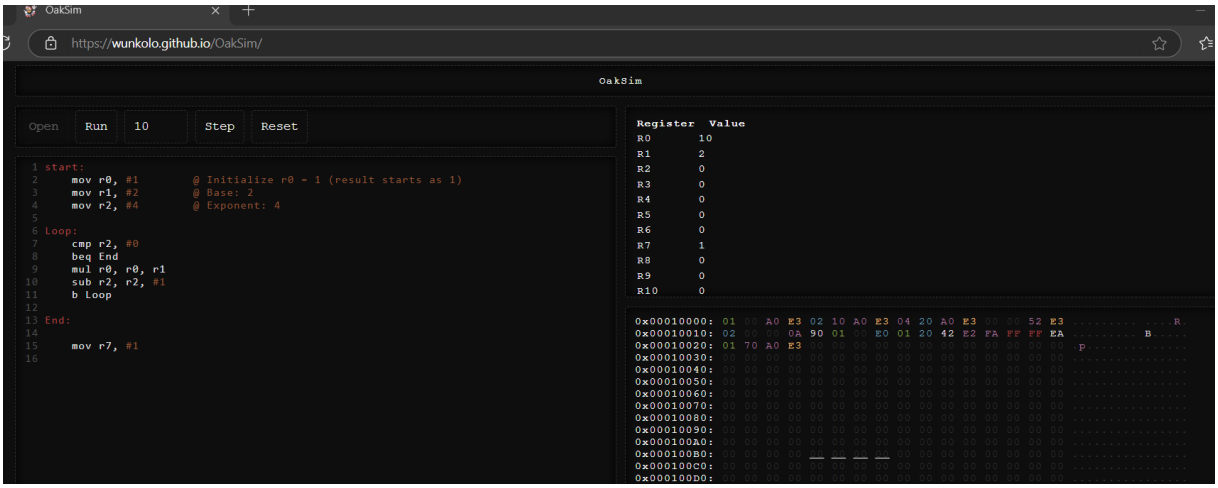
```
Main:
mov r1, #2
mov r2, #4

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**

(I used ChatGPT to get informations extra and write it in a better english)