

Template Week 4 – Software

Student number: 587707

Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:

The screenshot shows the OakSim ARM simulator interface. The left pane displays the following assembly code:

```
1 Main:
2     mov r2, #5
3     subs r3, r2, #1
4     ble End
5 Loop:
6     mul r2, r3, r2
7     subs r3, r3, #1
8     bne Loop
9 End:
10    mov r1, r2
11    mov r2, #0
```

The right pane shows the Register File and Memory Dump.

Register	Value
R0	0
R1	78
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10	0

The Memory Dump shows the following values:

Address	Value
0x00010000	05 20 A0 E3 01 30 52 E2 02 00
0x00010010	01 30 53 E2 FC FF FF 1A 02 00
0x00010020	00 00 00 00 00 00 00 00 00 00
0x00010030	00 00 00 00 00 00 00 00 00 00
0x00010040	00 00 00 00 00 00 00 00 00 00
0x00010050	00 00 00 00 00 00 00 00 00 00
0x00010060	00 00 00 00 00 00 00 00 00 00
0x00010070	00 00 00 00 00 00 00 00 00 00
0x00010080	00 00 00 00 00 00 00 00 00 00
0x00010090	00 00 00 00 00 00 00 00 00 00
0x000100A0	00 00 00 00 00 00 00 00 00 00
0x000100B0	00 00 00 00 00 00 00 00 00 00
0x000100C0	00 00 00 00 00 00 00 00 00 00
0x000100D0	00 00 00 00 00 00 00 00 00 00
0x000100E0	00 00 00 00 00 00 00 00 00 00
0x000100F0	00 00 00 00 00 00 00 00 00 00
0x00010100	00 00 00 00 00 00 00 00 00 00
0x00010110	00 00 00 00 00 00 00 00 00 00
0x00010120	00 00 00 00 00 00 00 00 00 00

A text box is overlaid on the bottom left of the simulator window, containing the following text:

Angela Visser
587707
26-12-2025

Assignment 4.2: Programming languages

Take screenshots that the following commands work:

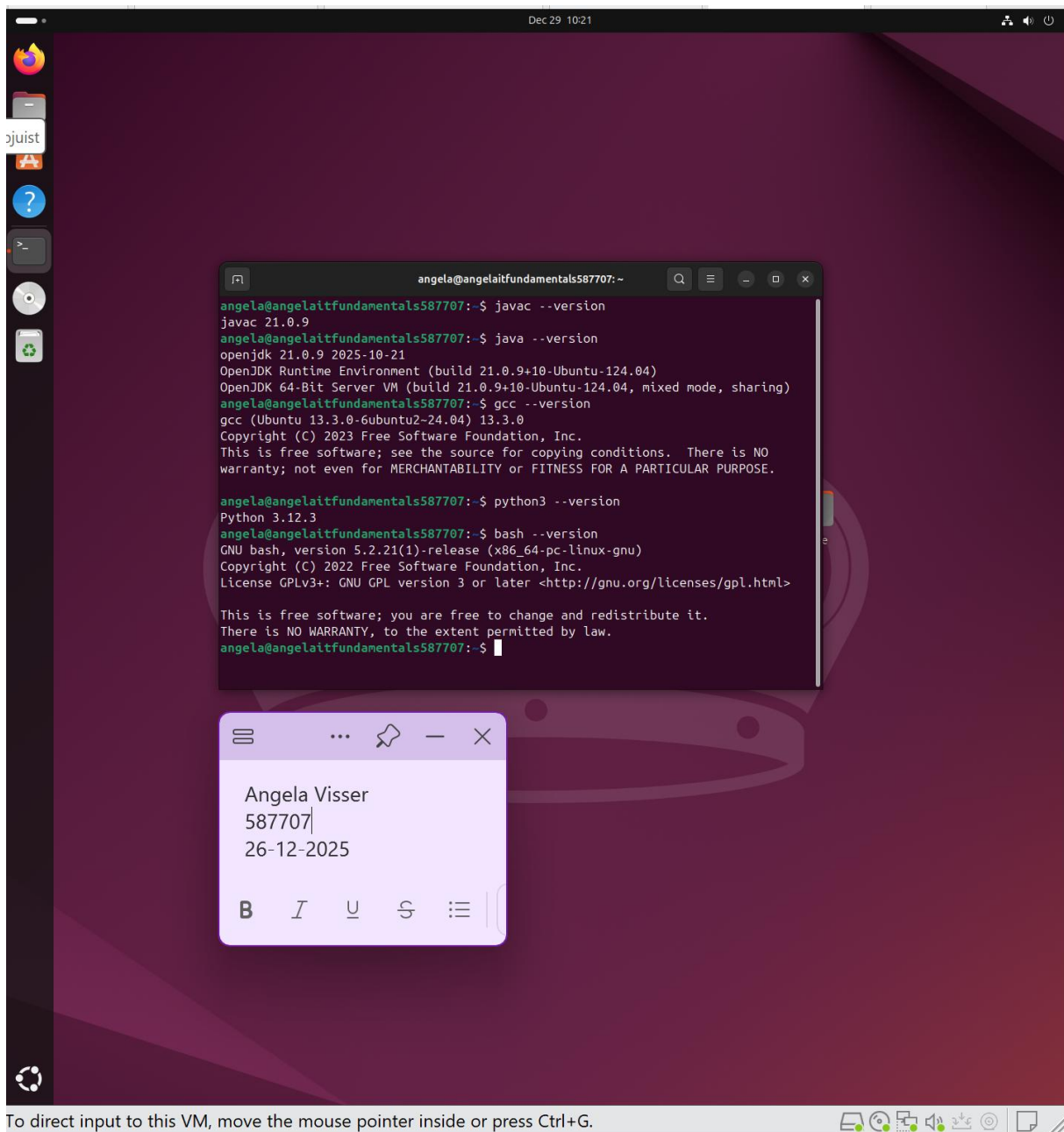
`javac --version`

`java --version`

`gcc --version`

`python3 --version`

`bash --version`



```
angela@angelaifundamentals587707: ~  
angela@angelaifundamentals587707:~$ javac --version  
javac 21.0.9  
angela@angelaifundamentals587707:~$ 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)  
angela@angelaifundamentals587707:~$ 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.  
angela@angelaifundamentals587707:~$ python3 --version  
Python 3.12.3  
angela@angelaifundamentals587707:~$ 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>  
This is free software; you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
angela@angelaifundamentals587707:~$
```

Angela Visser
587707
26-12-2025

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Assignment 4.3: Compile

O output

D disassemble

Fibonacci.java -java source code

fib.c -c source code

fib.py -python3 source code

fib.sh -bash script source code

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

Fibonacci.java en fib.c

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

Fib.c, c is a language that is compiled directly into machine code.
java is compiled into bytecode and runs on the java virtual machine.
the other two are interpreted languages.

Which source code files are compiled to byte code?

Fibonacci.Java, see explanation above.

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 is expected to run the fastest. This is because c is compiled directly into machine code.

How do I run a Java program?

First, Java has to be compiled into bytecode, which then is runs within the Java virtual machine.

How do I run a Python program?

A python program is read and executed by the interpreter. And the interpreter's machine code runs on the CPU.

How do I run a C program?

Because c is a compiled language, the source code is converted into machine instructions that can be directly executed by a processor.

How do I run a Bash script?

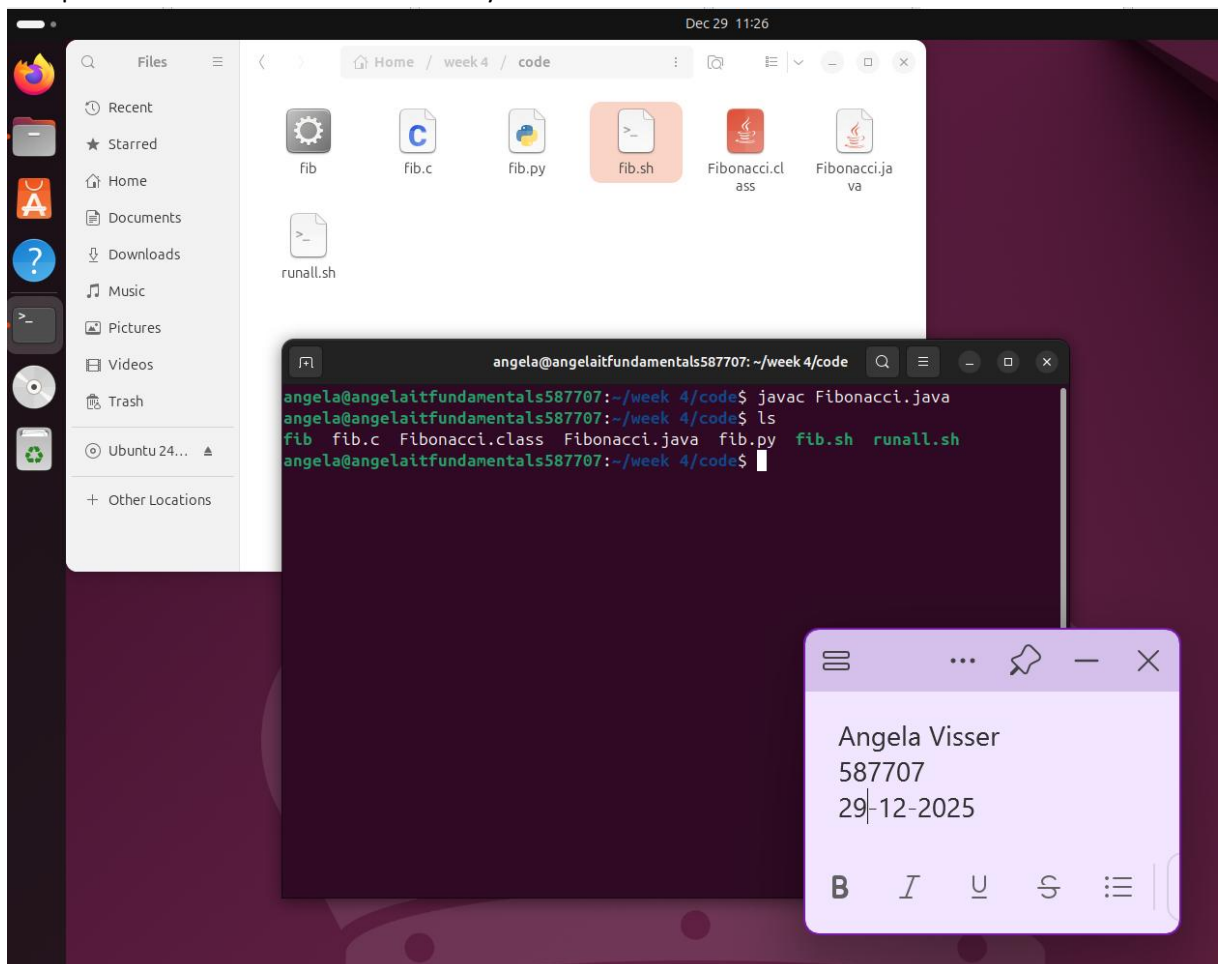
A bash script needs to be interpreted by an interpreter (Bash shell). Do note that we are trying to execute it on Ubuntu, so it needs to be made executable by changing the rights using the command: `[sudo chmod a+x fib.sh]`.

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

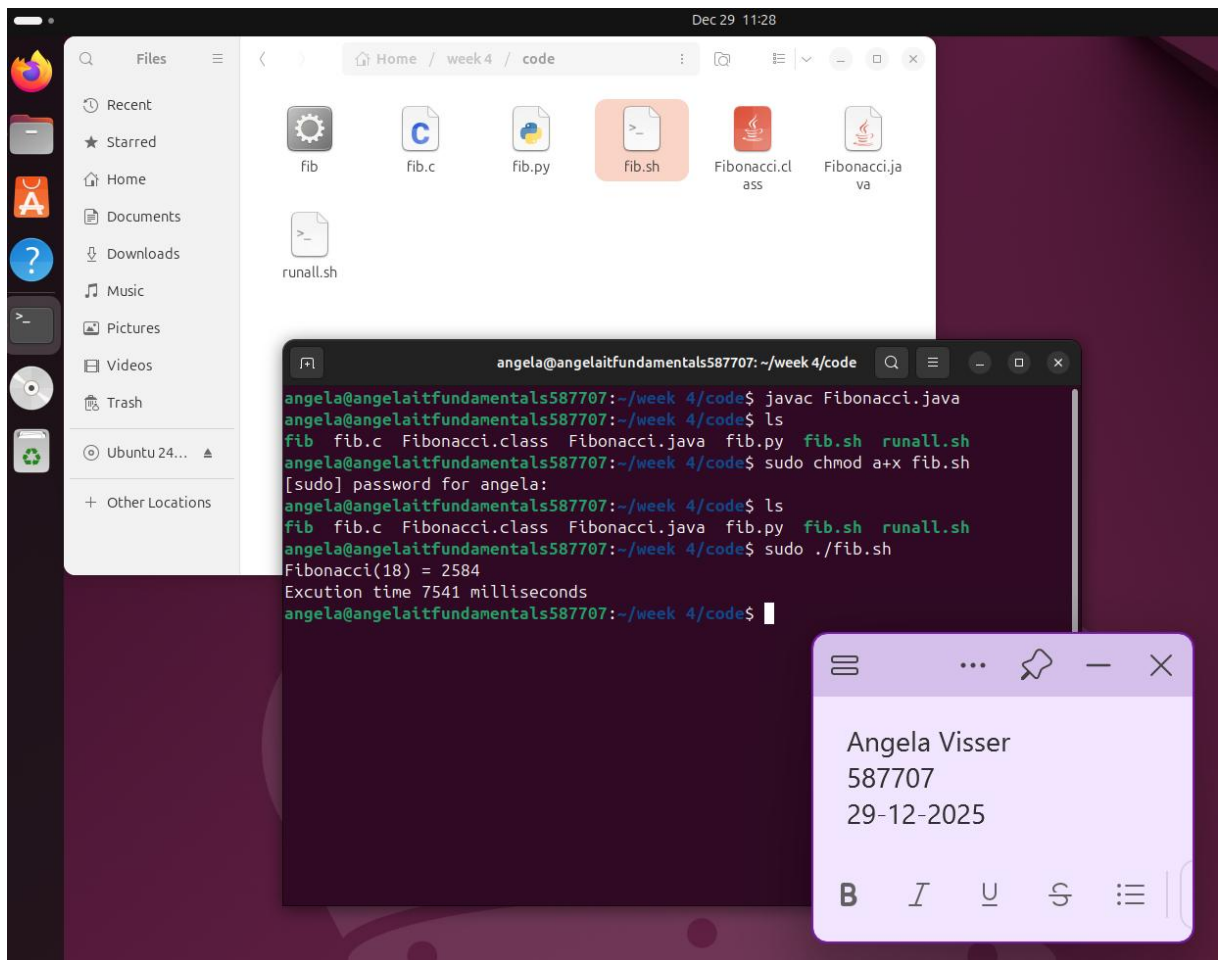
The compiled languages will add a file after compilation. The interpreted languages will not.
Fibonacci.java → Fibonacci.class
fib.c → fib.

Take relevant screenshots of the following commands:

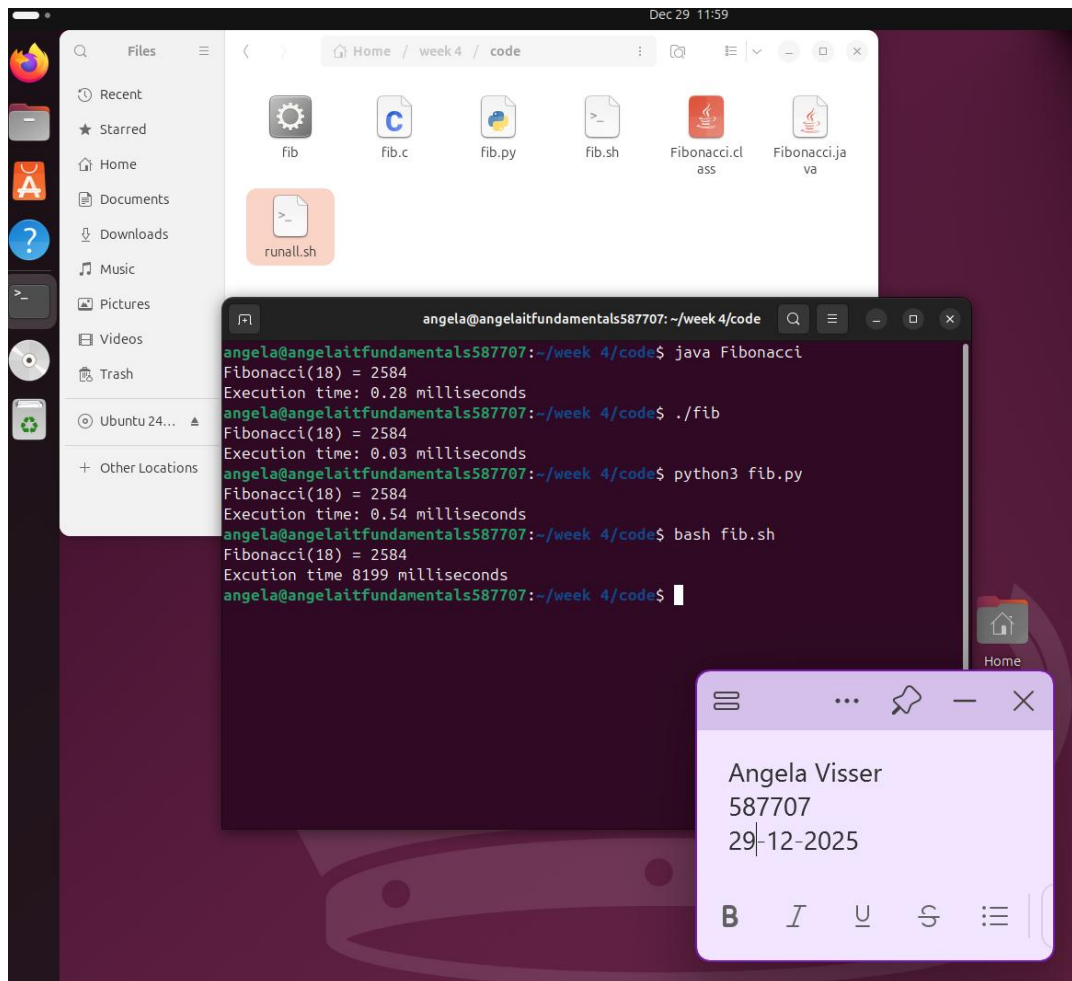
- Compile the source files where necessary



- Make them executable



- Run them



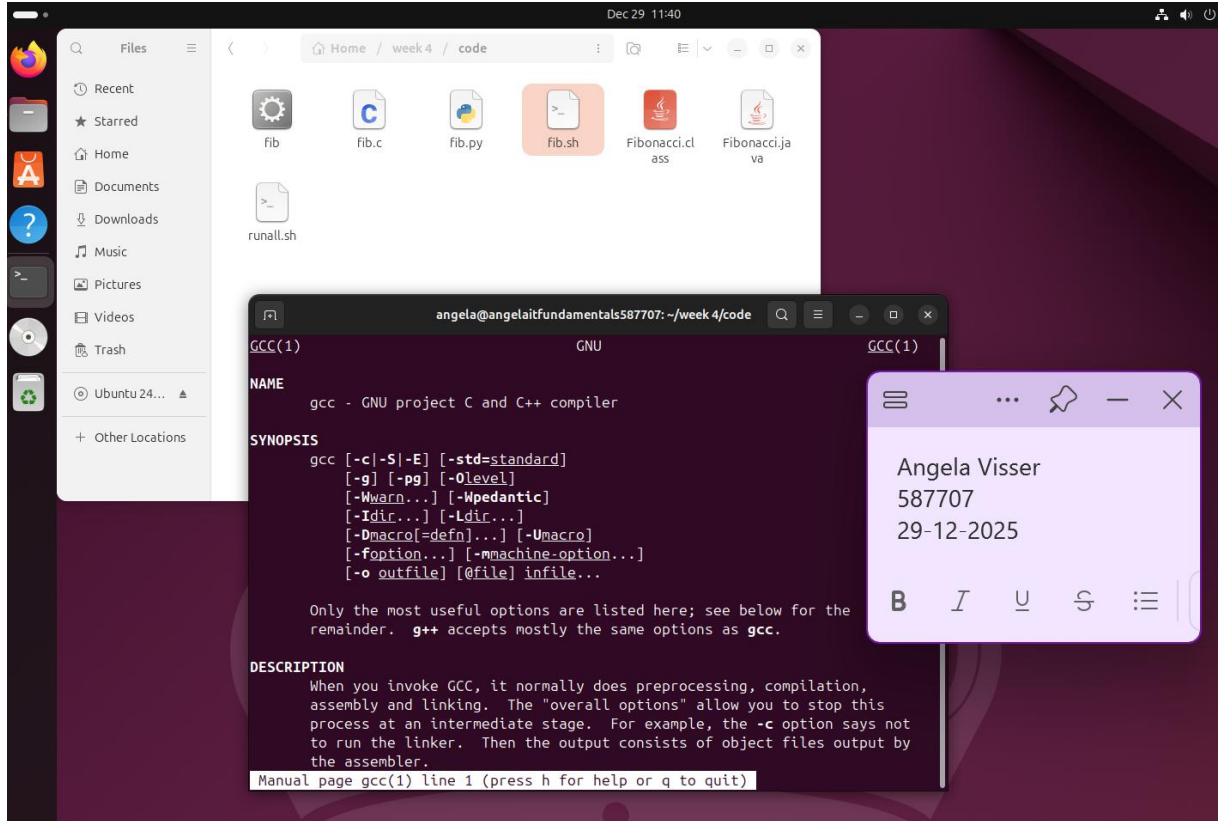
- Which (compiled) source code file performs the calculation the fastest?

The c source code performs the fastest.

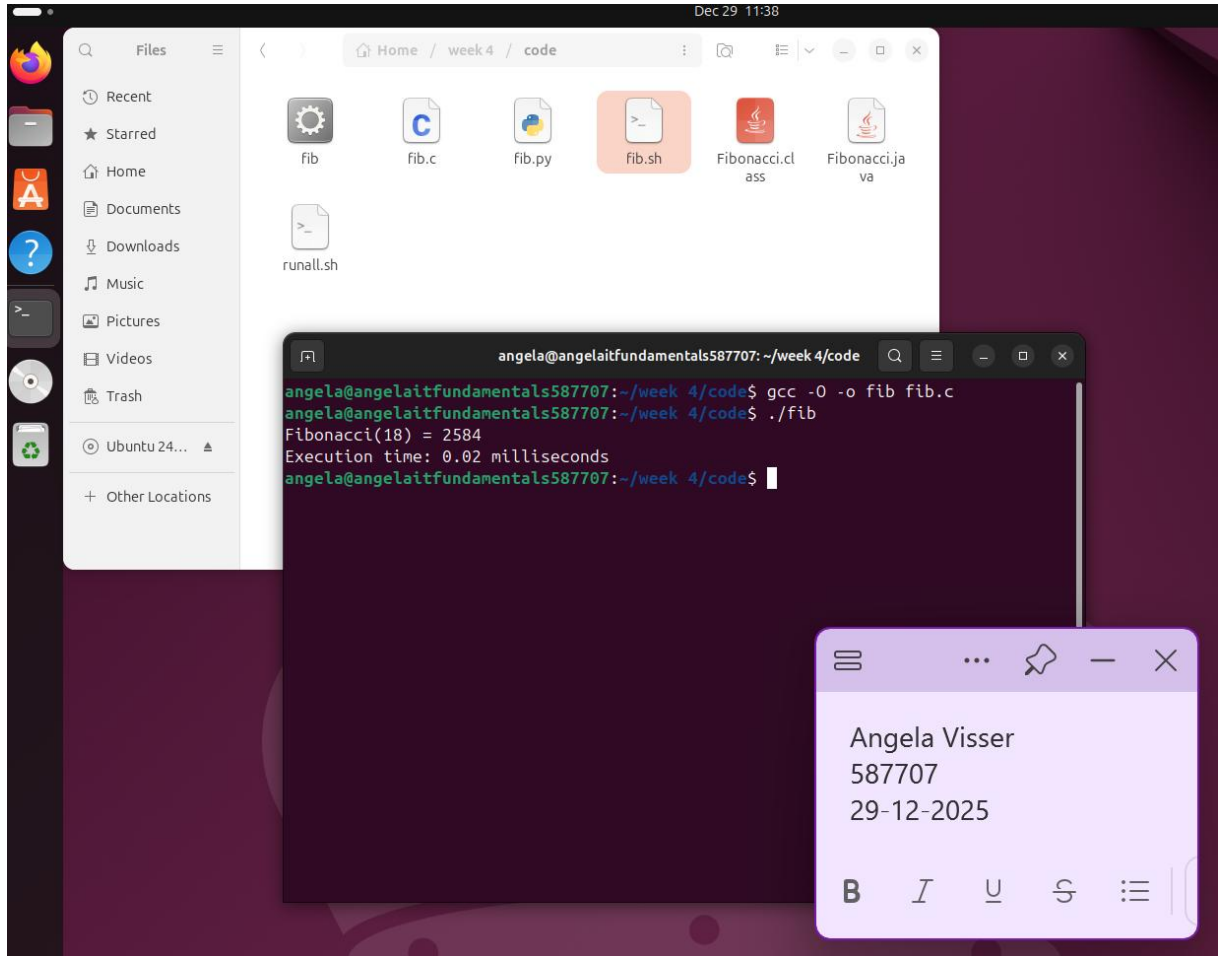
Assignment 4.4: Optimize

Take relevant screenshots of the following commands:

- 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.



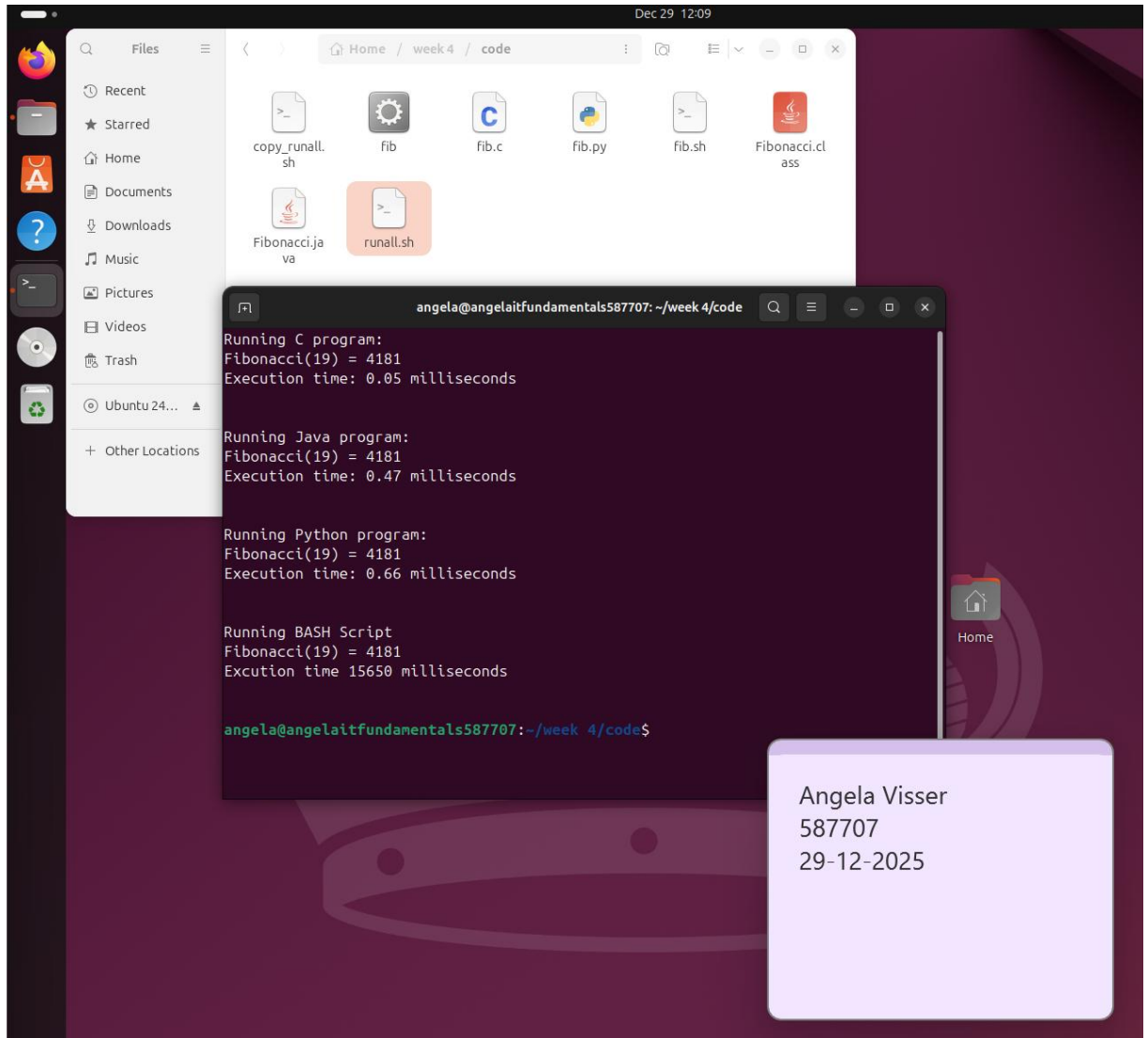
b) Compile **fib.c** again with the optimization parameters



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

Yes

- 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.



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
```

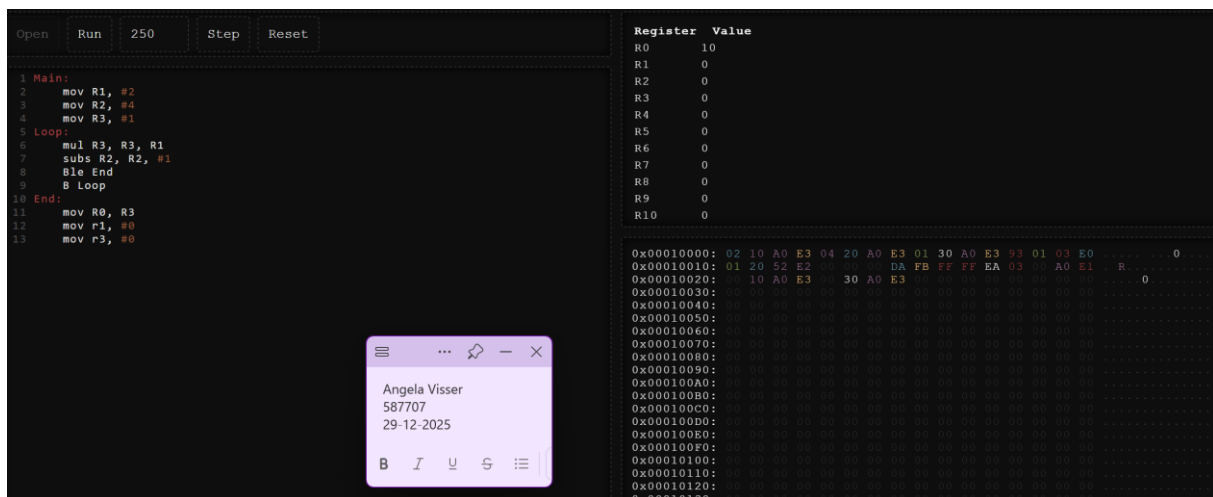
```
mov r2, #4
```

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](#)