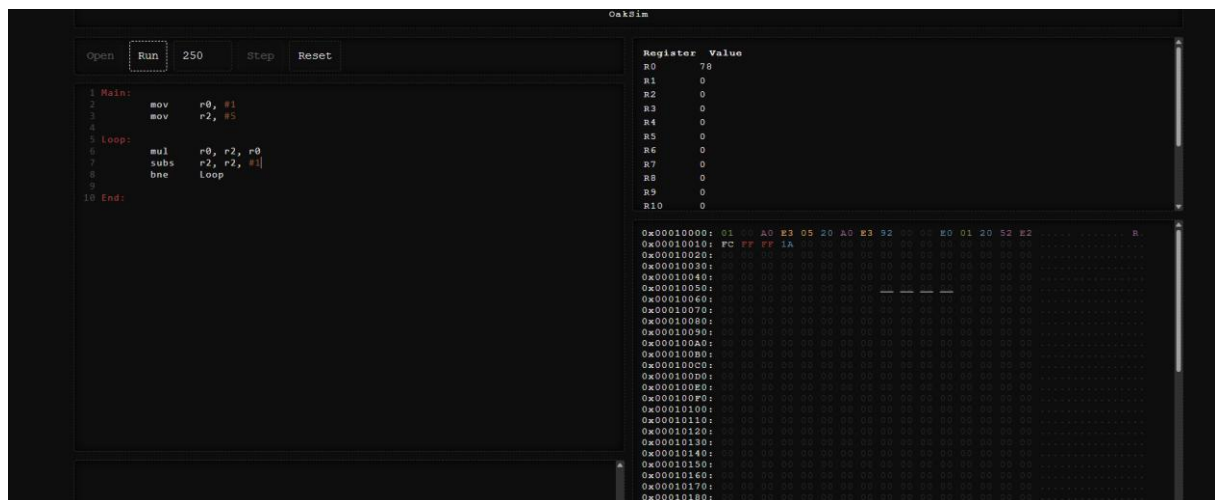


Template Week 4 – Software

Student number:

Assignment 4.1: ARM assembly

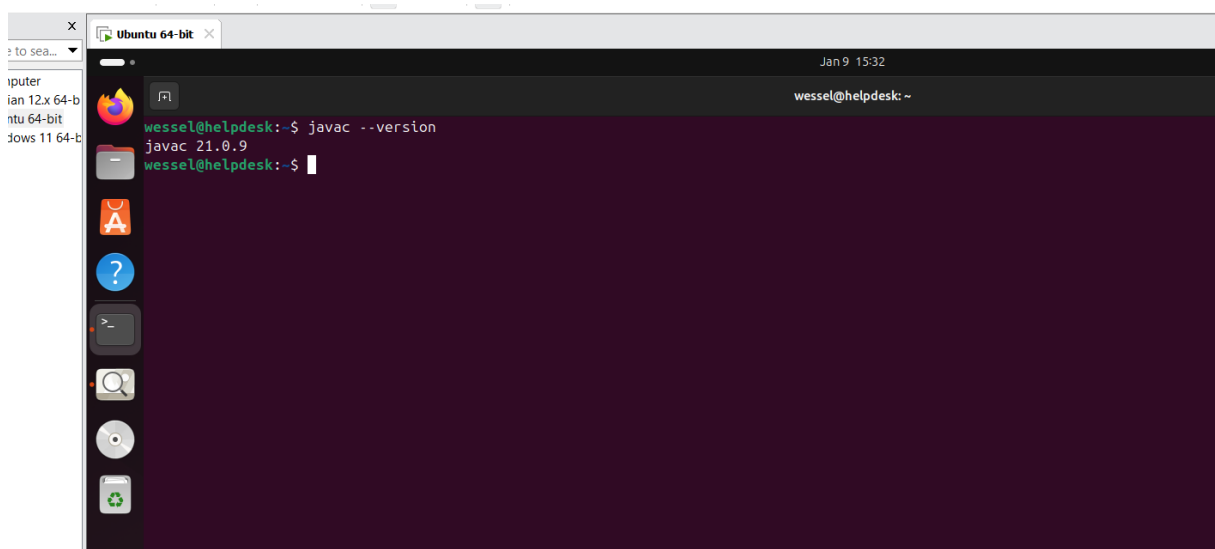
Screenshot of working assembly code of factorial calculation:



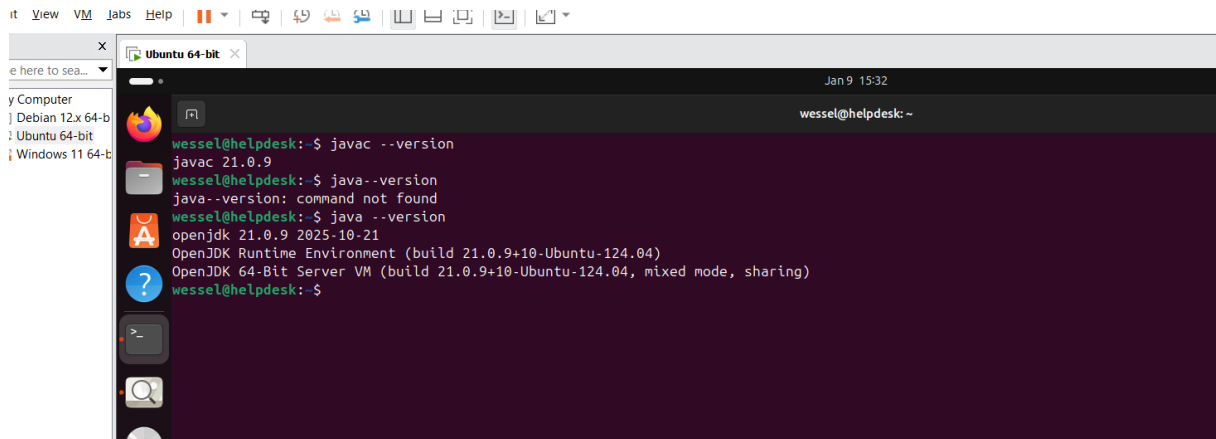
Assignment 4.2: Programming languages

Take screenshots that the following commands work:

javac --version



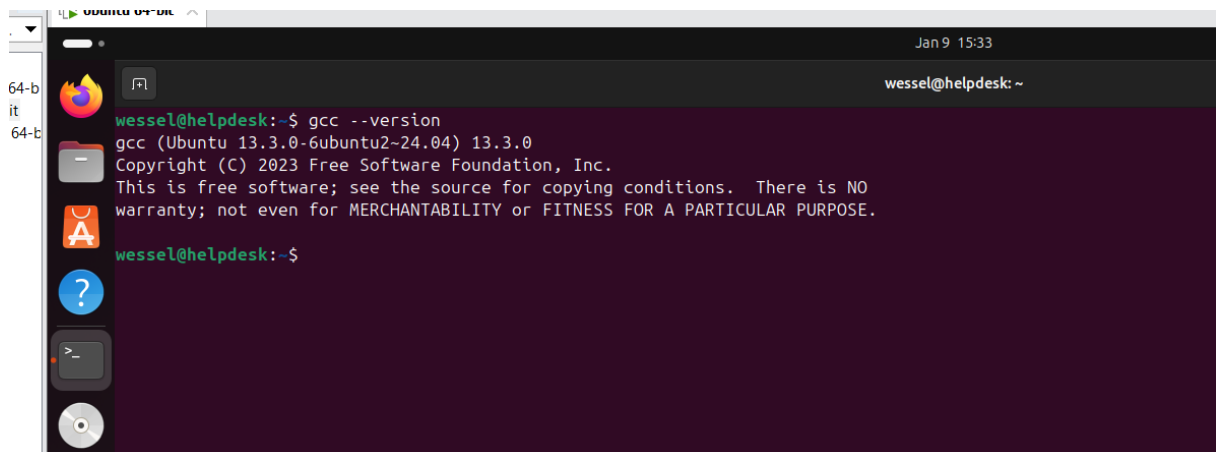
java --version



The screenshot shows a terminal window titled 'Ubuntu 64-bit' with a dark purple background. The user 'wessel@helpdesk' is at the prompt. The following commands and their outputs are shown:

```
wessel@helpdesk:~$ javac --version
javac 21.0.9
wessel@helpdesk:~$ java --version
java --version: command not found
wessel@helpdesk:~$ 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)
wessel@helpdesk:~$
```

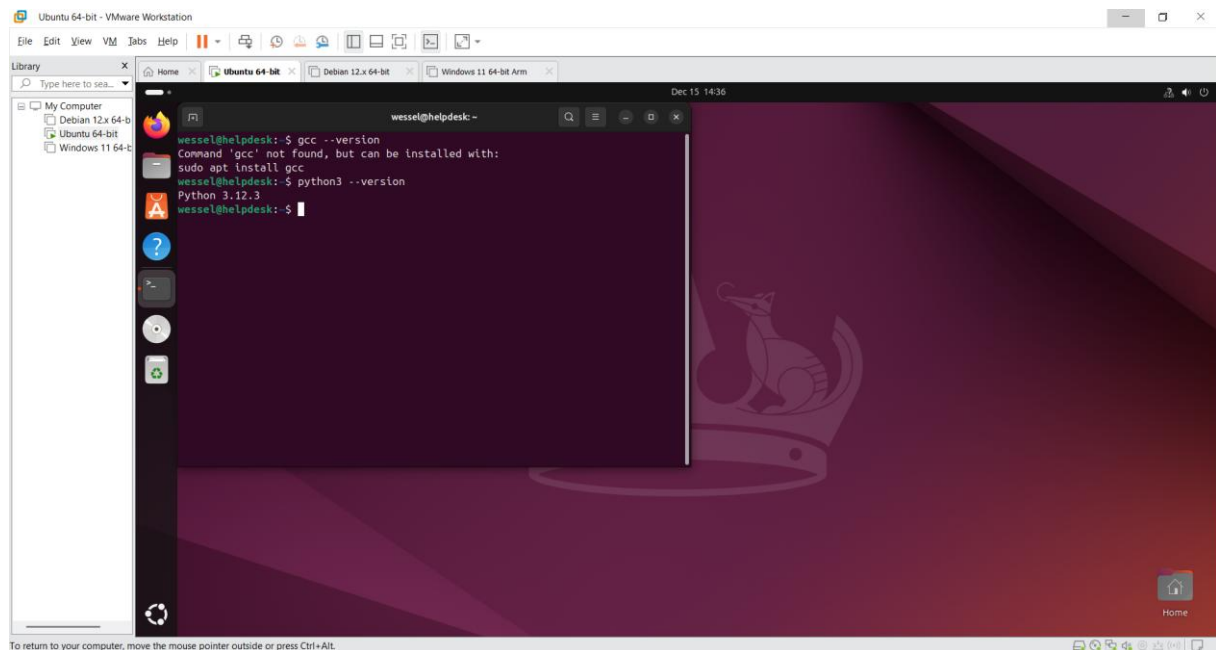
gcc --version



The screenshot shows a terminal window titled 'Ubuntu 64-bit' with a dark purple background. The user 'wessel@helpdesk' is at the prompt. The following commands and their outputs are shown:

```
wessel@helpdesk:~$ 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.
wessel@helpdesk:~$
```

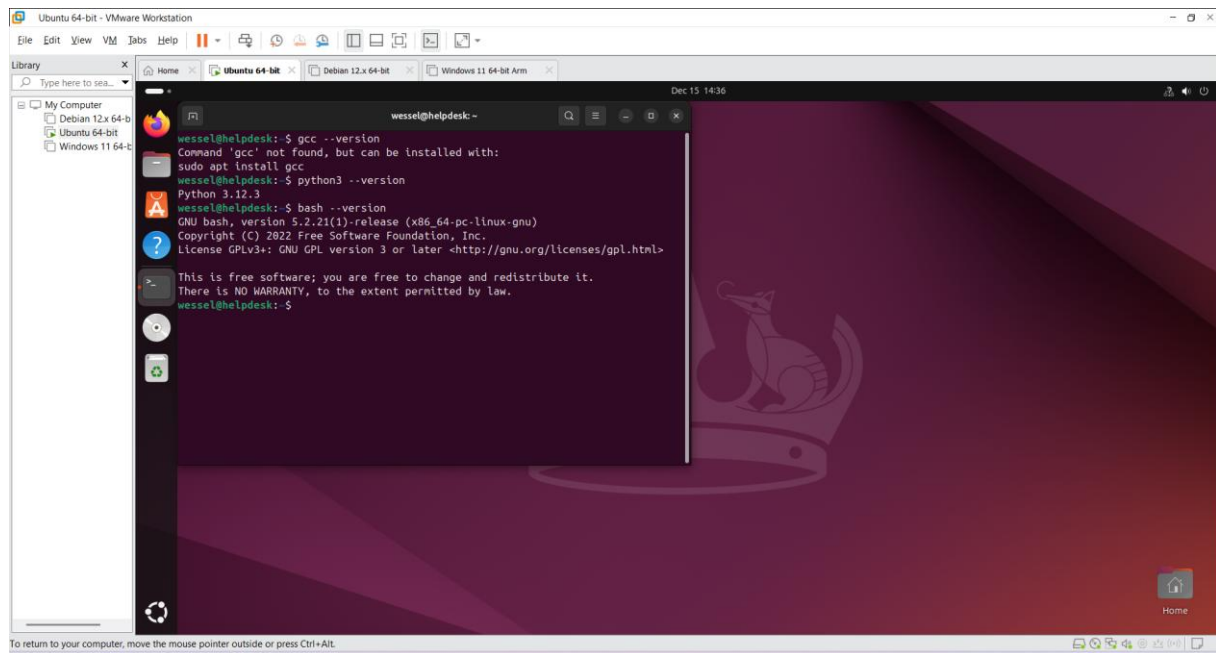
python3 --version



The screenshot shows a terminal window titled 'Ubuntu 64-bit' with a dark purple background. The user 'wessel@helpdesk' is at the prompt. The following commands and their outputs are shown:

```
wessel@helpdesk:~$ gcc --version
Command 'gcc' not found, but can be installed with:
sudo apt install gcc
wessel@helpdesk:~$ python3 --version
Python 3.12.3
wessel@helpdesk:~$
```

bash --version



Assignment 4.3: Compile

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

Java en C

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

C C++, Assembly

Which source code files are compiled to byte code?

Java, C#

Which source code files are interpreted by an interpreter?

Py, Sh, Bash

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

C, omdat hij gelijk dan word gecompileerd naar machinecode

How do I run a Java program?

Om te compileren doe je `javac hello.java`

En dan uitvoeren doe je `hello`

How do I run a Python program?

Je kan hem direct uitvoeren met

`Python3 hello.py`

How do I run a C program?

Om te compileren doe je `gcc hello.c -o hello`

En dan uitvoeren doe je `./hello`

How do I run a Bash script?

Om hem uitvoerbaar te maken doe je `chmod +x hello.sh`

En dan uitvoeren doe je `./hello.sh`

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

Ja `runall.sh`

Take relevant screenshots of the following commands:

- Compile the source files where necessary
- Make them executable
- Run them

```
C:\Saxion\virtual machines\Ubuntu 64-bit\Ubuntu 64-bit.vmx 24.04) 13.5.0
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

wessel@helpdesk:~$ 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.

wessel@helpdesk:~$ cd code
wessel@helpdesk:~/code$ ls
fib.c  Fibonacci.java  fib.py  fib.sh  runall.sh
wessel@helpdesk:~/code$
```

Fib.sh

```
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.

wessel@helpdesk:~$ cd code
wessel@helpdesk:~/code$ ls
fib.c  Fibonacci.java  fib.py  fib.sh  runall.sh
wessel@helpdesk:~/code$ sudo ./fib.sh
Fibonacci(18) = 2584
Execution time 3600 milliseconds
wessel@helpdesk:~/code$
```

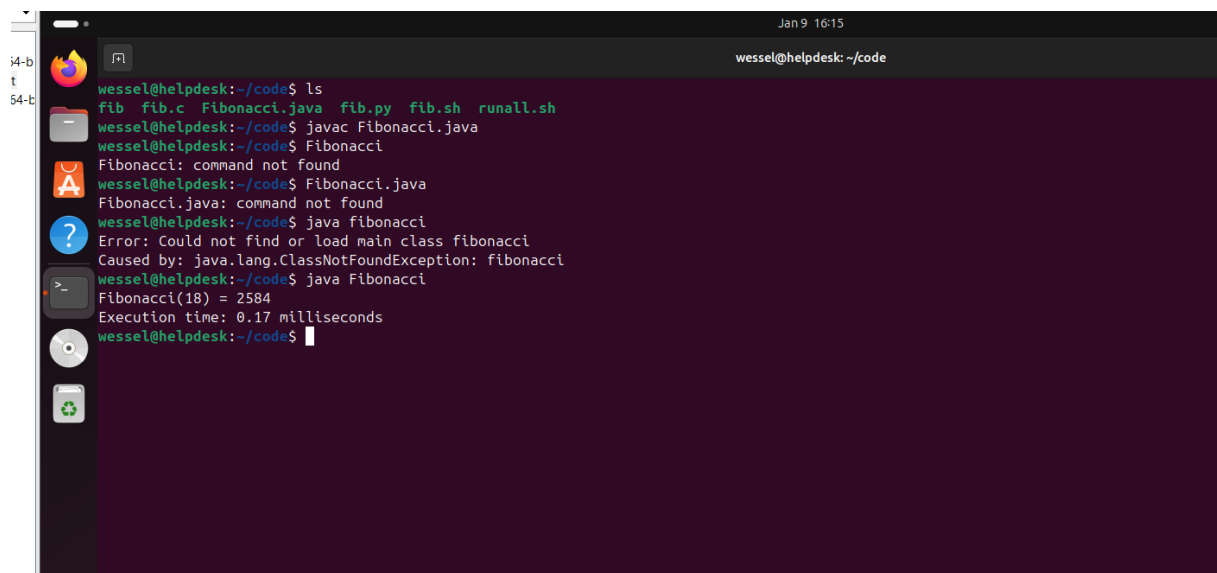
Python3

```
Fibonacci(18) = 2584
Execution time 3600 milliseconds
wessel@helpdesk:~/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.20 milliseconds
wessel@helpdesk:~/code$
```

C

```
Excution time 3600 milliseconds
wessel@helpdesk:~/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.20 milliseconds
wessel@helpdesk:~/code$ gcc fib.c -o fib
wessel@helpdesk:~/code$ ./fib.c
./fib.c: line 5: syntax error near unexpected token '('
./fib.c: line 5: `int fibonacci(int n) {'
wessel@helpdesk:~/code$ ./fib
Fibonacci(18) = 2584
Execution time: 0.05 milliseconds
wessel@helpdesk:~/code$
```

Java



The screenshot shows a terminal window titled 'wessel@helpdesk: ~/code' with a timestamp of 'Jan 9 16:15'. The terminal output is as follows:

```
wessel@helpdesk:~/code$ ls
fib fib.c Fibonacci.java fib.py fib.sh runall.sh
wessel@helpdesk:~/code$ javac Fibonacci.java
wessel@helpdesk:~/code$ Fibonacci
Fibonacci: command not found
wessel@helpdesk:~/code$ Fibonacci.java
Fibonacci.java: command not found
wessel@helpdesk:~/code$ java fibonacci
Error: Could not find or load main class fibonacci
Caused by: java.lang.ClassNotFoundException: fibonacci
wessel@helpdesk:~/code$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.17 milliseconds
wessel@helpdesk:~/code$
```

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

C omdat het direct naar machinecode word vertaald

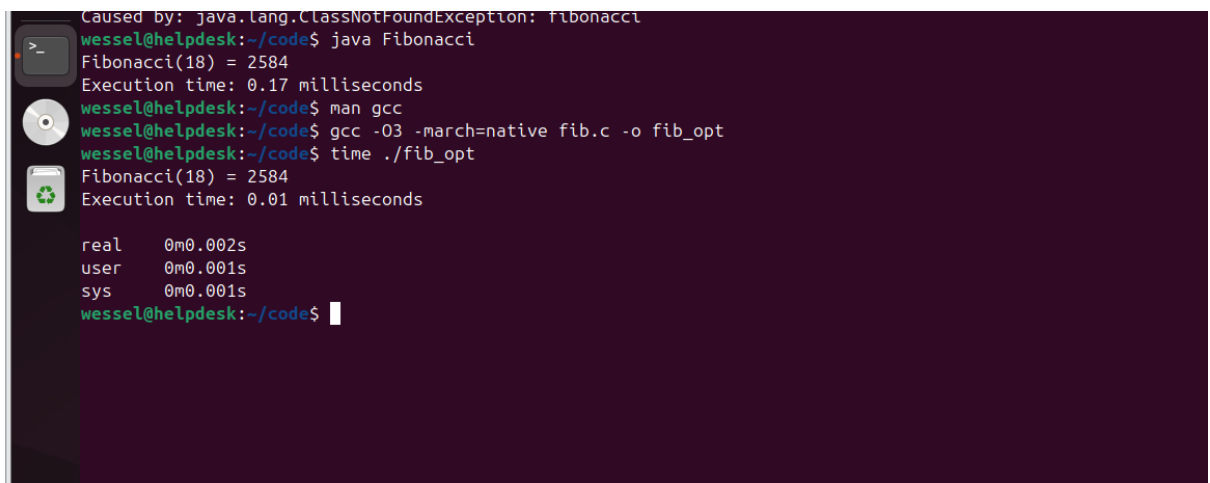
Assignment 4.4: Optimize

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.

O3 – march=native

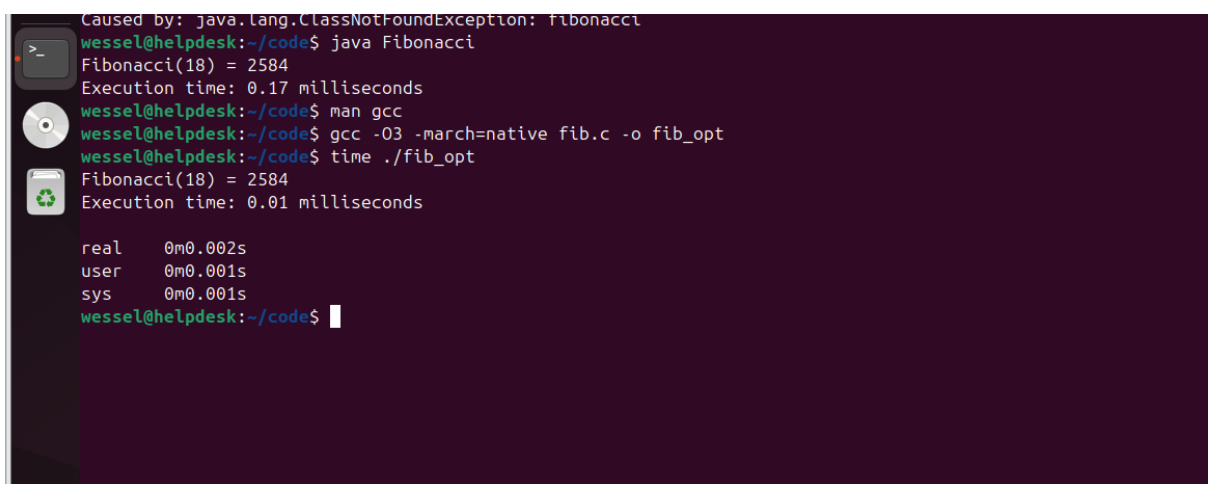
- b) Compile **fib.c** again with the optimization parameters



```
Caused by: java.lang.ClassNotFoundException: Fibonacci
wessel@helpdesk:~/code$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.17 milliseconds
wessel@helpdesk:~/code$ man gcc
wessel@helpdesk:~/code$ gcc -O3 -march=native fib.c -o fib_opt
wessel@helpdesk:~/code$ time ./fib_opt
Fibonacci(18) = 2584
Execution time: 0.01 milliseconds

real    0m0.002s
user    0m0.001s
sys      0m0.001s
wessel@helpdesk:~/code$
```

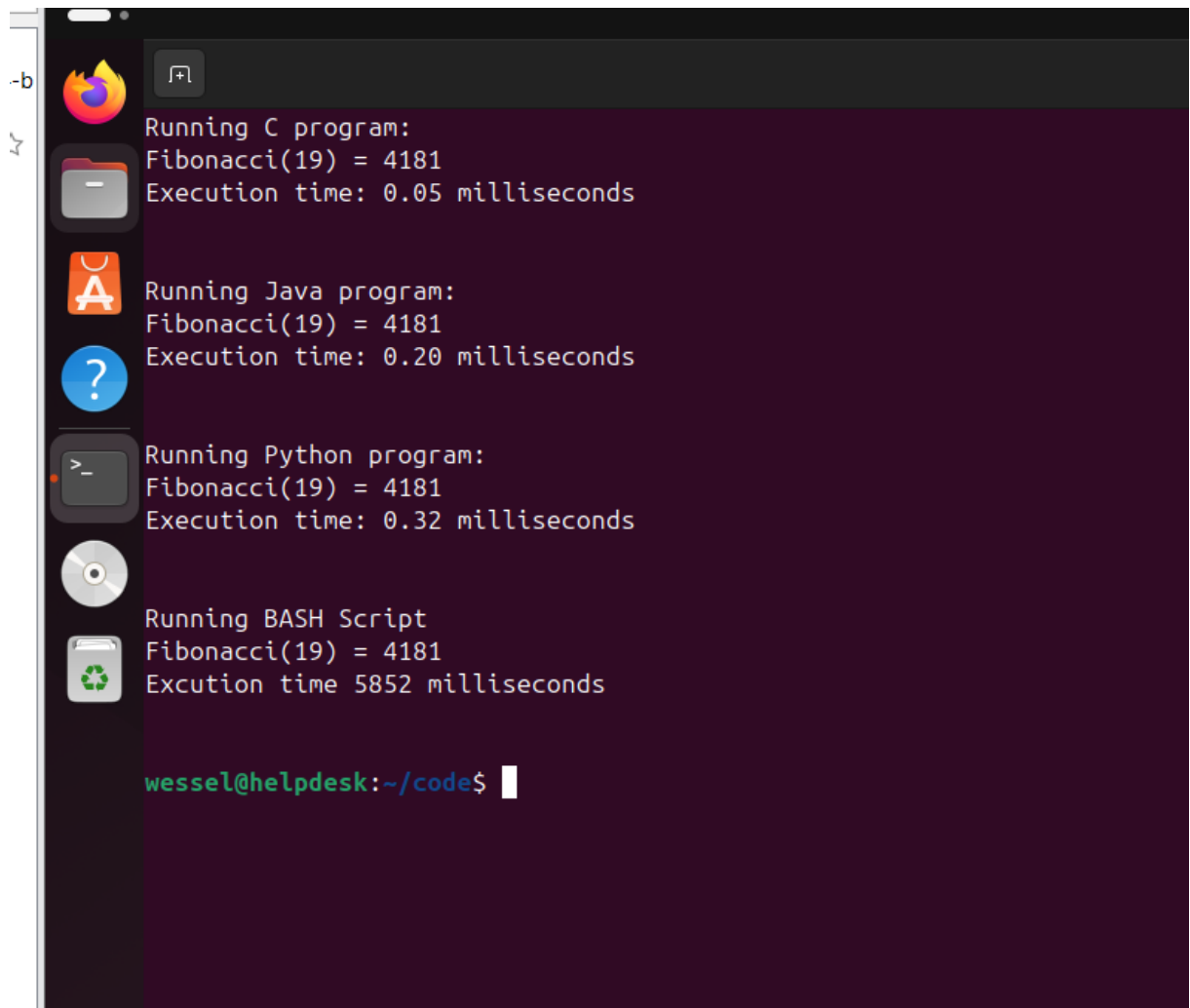
- c) Run the newly compiled program. Is it true that it now performs the calculation faster?



```
Caused by: java.lang.ClassNotFoundException: Fibonacci
wessel@helpdesk:~/code$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.17 milliseconds
wessel@helpdesk:~/code$ man gcc
wessel@helpdesk:~/code$ gcc -O3 -march=native fib.c -o fib_opt
wessel@helpdesk:~/code$ time ./fib_opt
Fibonacci(18) = 2584
Execution time: 0.01 milliseconds

real    0m0.002s
user    0m0.001s
sys      0m0.001s
wessel@helpdesk:~/code$
```

- 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 r0, #1
mov r1, #2
mov r2, #4
```

Loop:

```
cmp r2, #0
beq End
mul r0, r0, r1
```



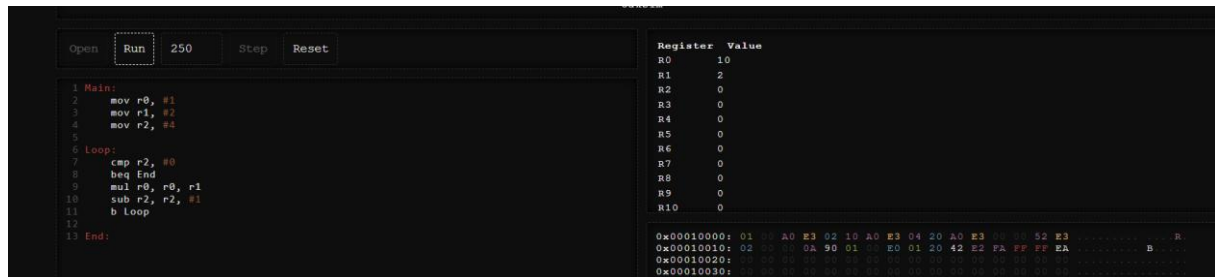
```
sub r2, r2, #1
```

```
b 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](#)