

# Template Week 4 – Software

Student number:

## Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:

The screenshot shows a web-based ARM assembly debugger called OakSim. The assembly code is as follows:

```
1 Loop:
2     mov r0, #10
3     mov r1, #9
4     add r2, r1, r0
5     sub r3, r0, r1
6     cmp r2, r3
7     bne Exit
8     B Loop
9 Exit:
10    mov r4, #8
```

The register values are:

Register	Value
R0	a
R1	9
R2	13
R3	1
R4	8
R5	0
R6	0
R7	0
R8	0

The memory dump shows the stack and heap area starting at address 0x000010000, containing the assembly code and some initial values.

## Assignment 4.2: Programming languages

Take screenshots that the following commands work:

javac –version

```
ubuntu@ubuntu2404:~$ python3: command not found
ubuntu@ubuntu2404:~$ 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.
ubuntu@ubuntu2404:~$ javac --version
Command 'javac' not found, but can be installed with:
sudo apt install openjdk-17-jdk-headless # version 17.0.16+8~us1~0ubuntu1-24.04
.1, or
sudo apt install openjdk-21-jdk-headless # version 21.0.8+9~us1~0ubuntu1-24.04.
1
sudo apt install default-jdk      # version 2:1.17-75
sudo apt install openjdk-11-jdk-headless # version 11.0.28+6~ubuntu1-24.04.1
sudo apt install openjdk-25-jdk-headless # version 25+36-1-24.04.2
sudo apt install openjdk-8-jdk-headless # version 8u462~ga~us1~0ubuntu2-24.04.
2
sudo apt install ecj      # version 3.32.0+eclipse4.26-2
sudo apt install openjdk-19-jdk-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jdk-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jdk-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$
```

java --version

```
ubuntu@ubuntu2404:~$ sudo apt install default-jdk      # version 2:1.17-75
sudo apt install openjdk-11-jdk-headless # version 11.0.28+6~ubuntu1-24.04.1
sudo apt install openjdk-25-jdk-headless # version 25+36-1-24.04.2
sudo apt install openjdk-8-jdk-headless # version 8u462~ga~us1~0ubuntu2-24.04.
2
sudo apt install ecj      # version 3.32.0+eclipse4.26-2
sudo apt install openjdk-19-jdk-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jdk-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jdk-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless # version 17.0.16+8~us1~0ubuntu1-24.04
.1, or
sudo apt install openjdk-21-jre-headless # version 21.0.8+9~us1~0ubuntu1-24.04.
1
sudo apt install default-jre      # version 2:1.17-75
sudo apt install openjdk-11-jre-headless # version 11.0.28+6~ubuntu1-24.04.1
sudo apt install openjdk-25-jre-headless # version 25+36-1-24.04.2
sudo apt install openjdk-8-jre-headless # version 8u462~ga~us1~0ubuntu2-24.04.
2
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jre-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$
```

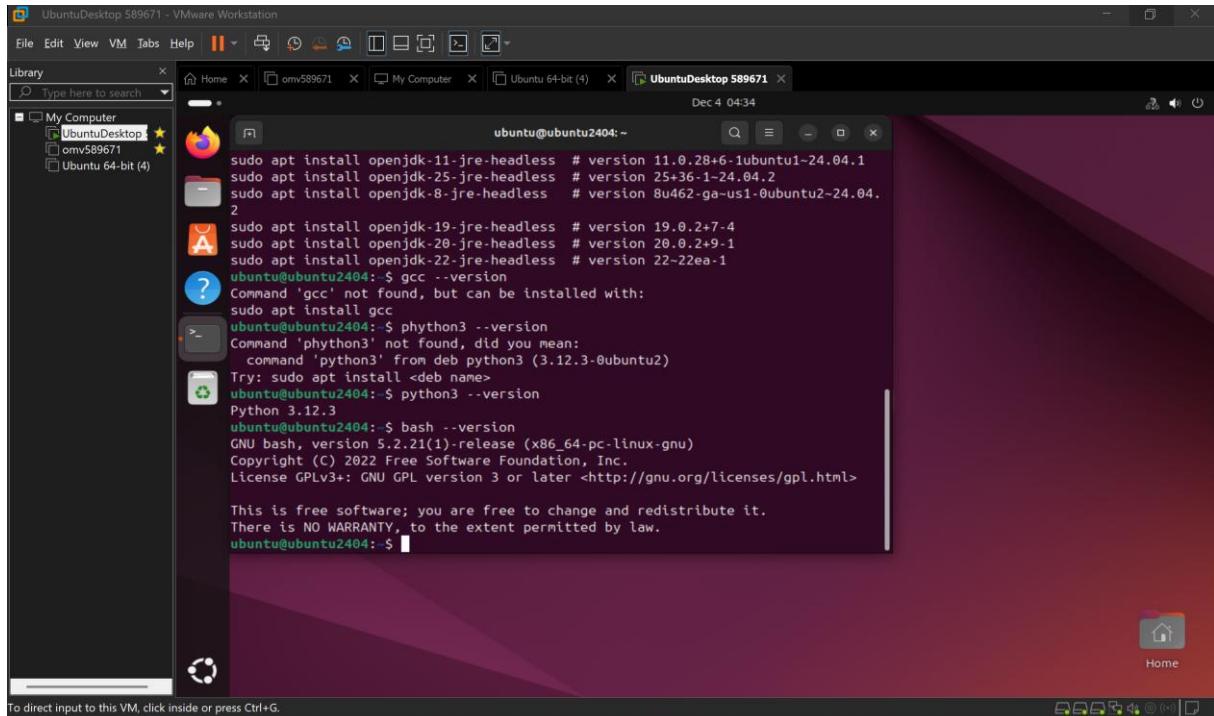
gcc --version

```
ubuntu@ubuntu2404:~$ sudo apt install openjdk-8-jdk-headless # version 8u462-ga-us1-0ubuntu2-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install ecj # version 3.32.0+eclipse4.26-2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-19-jdk-headless # version 19.0.2+7-4
ubuntu@ubuntu2404:~$ sudo apt install openjdk-20-jdk-headless # version 20.0.2+9-1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-22-jdk-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless # version 17.0.16+8-us1-0ubuntu1-24.04.1, or
sudo apt install openjdk-21-jre-headless # version 21.0.8+9-us1-0ubuntu1-24.04.1
ubuntu@ubuntu2404:~$ sudo apt install default-jre # version 2:1.17-75
ubuntu@ubuntu2404:~$ sudo apt install openjdk-11-jre-headless # version 11.0.28+6-1ubuntu1-24.04.1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-25-jre-headless # version 25+36-1-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-8-jre-headless # version 8u462-ga-us1-0ubuntu2-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
ubuntu@ubuntu2404:~$ sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-22-jre-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$ gcc --version
Command 'gcc' not found, but can be installed with:
sudo apt install gcc
ubuntu@ubuntu2404:~$
```

`python3 --version`

```
ubuntu@ubuntu2404:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless # version 17.0.16+8-us1-0ubuntu1-24.04.1, or
sudo apt install openjdk-21-jre-headless # version 21.0.8+9-us1-0ubuntu1-24.04.1
ubuntu@ubuntu2404:~$ sudo apt install default-jre # version 2:1.17-75
ubuntu@ubuntu2404:~$ sudo apt install openjdk-11-jre-headless # version 11.0.28+6-1ubuntu1-24.04.1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-25-jre-headless # version 25+36-1-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-8-jre-headless # version 8u462-ga-us1-0ubuntu2-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
ubuntu@ubuntu2404:~$ sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-22-jre-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$ gcc --version
Command 'gcc' not found, but can be installed with:
sudo apt install gcc
ubuntu@ubuntu2404:~$ python3 --version
Command 'python3' not found, did you mean:
  command 'python3' from deb python3 (3.12.3-0ubuntu2)
Try: sudo apt install <deb name>
ubuntu@ubuntu2404:~$ python3 --version
Python 3.12.3
ubuntu@ubuntu2404:~$
```

`bash --version`



### **Assignment 4.3: Compile**

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

**Javac en GCC. De andere zijn interplers.**

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

**Gcc. Alle andere gebruiken iets van een tussenstap voordat het gerunt kan worden.**

Which source code files are compiled to byte code?

**Javac. Dit compiled het zodat java het kan runnen**

Which source code files are interpreted by an interpreter?

Python en bash

These source code files will perform the same calculation after compilation/interpretation.

Which one is expected to do the calculation the fastest?

**Gcc, deze praat direct met de hardware zonder tussen stap**

How do I run a Java program?

**Een .java bestand word gecompiled door een java compiler. Deze gegevens worden omgezet naar bytecode die de can gerunt word op een JVM.**

How do I run a Python program?

**Een .py bestand staat ergens op de computer. Dan can deze gerunt worden door hem met python op te starten. Bijv python3 helloworld.py**

How do I run a C program?

.c bestand word gerunt. Door GCC. Gcc maakt een uitvoerbaar bestand

How do I run a Bash script?

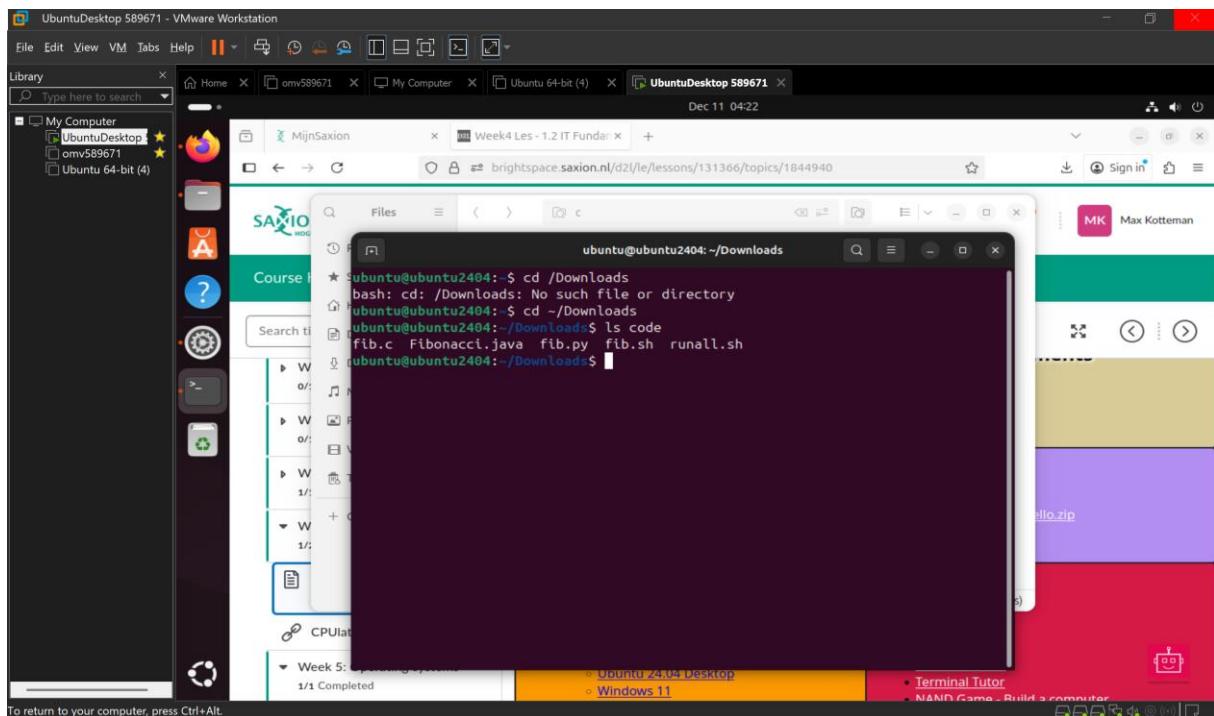
Een Bash-script maak je door een bestand met de extensie .sh te schrijven en bovenaan #!/bin/bash te zetten. Daarna geef je het uitvoerrechten met chmod +x script.sh en start je het met ./script.sh.

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

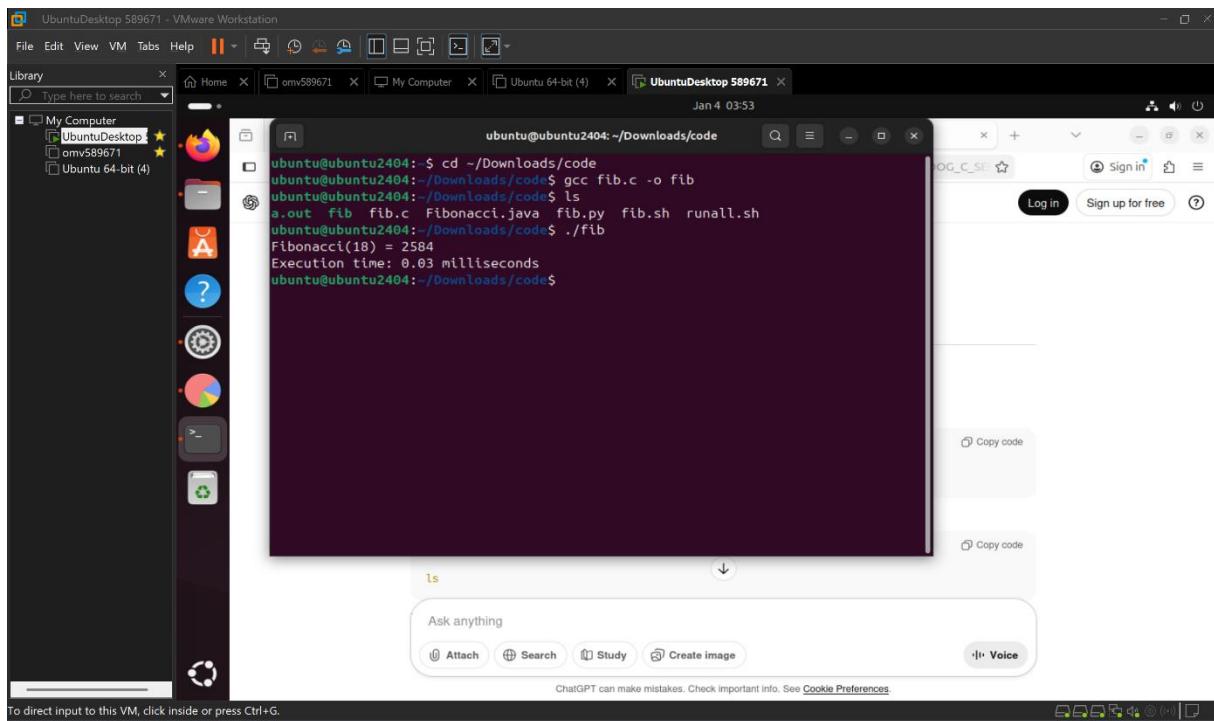
Ja, nieuwe bestanden ontstaan bij Java , C en soms bij Python.

Take relevant screenshots of the following commands:

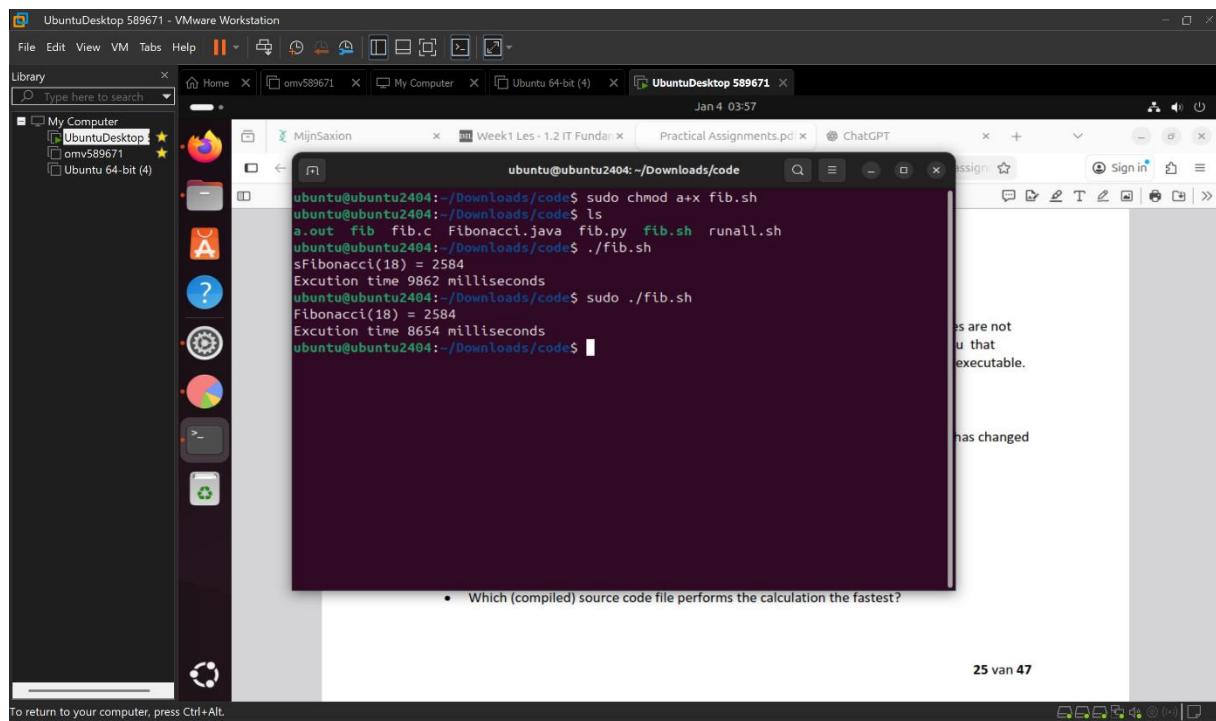
- Compile the source files where necessary
- Make them executable
- Run them
- Which (compiled) source code file performs the calculation the fastest?



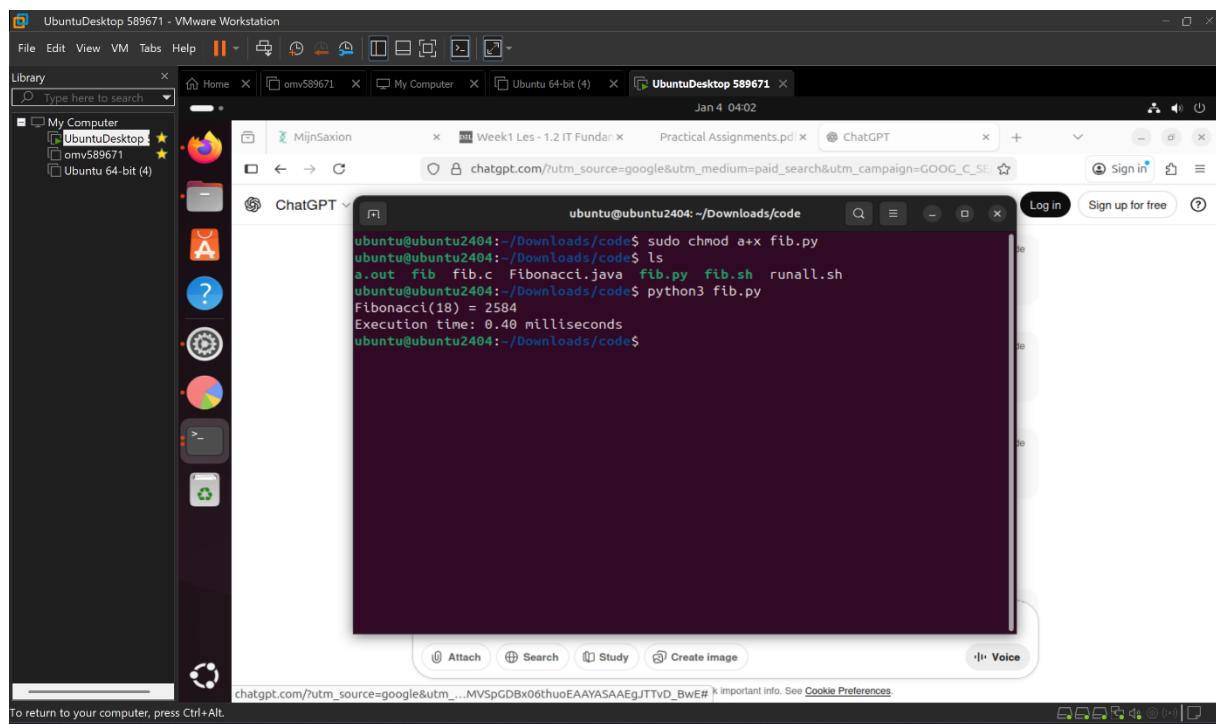
C



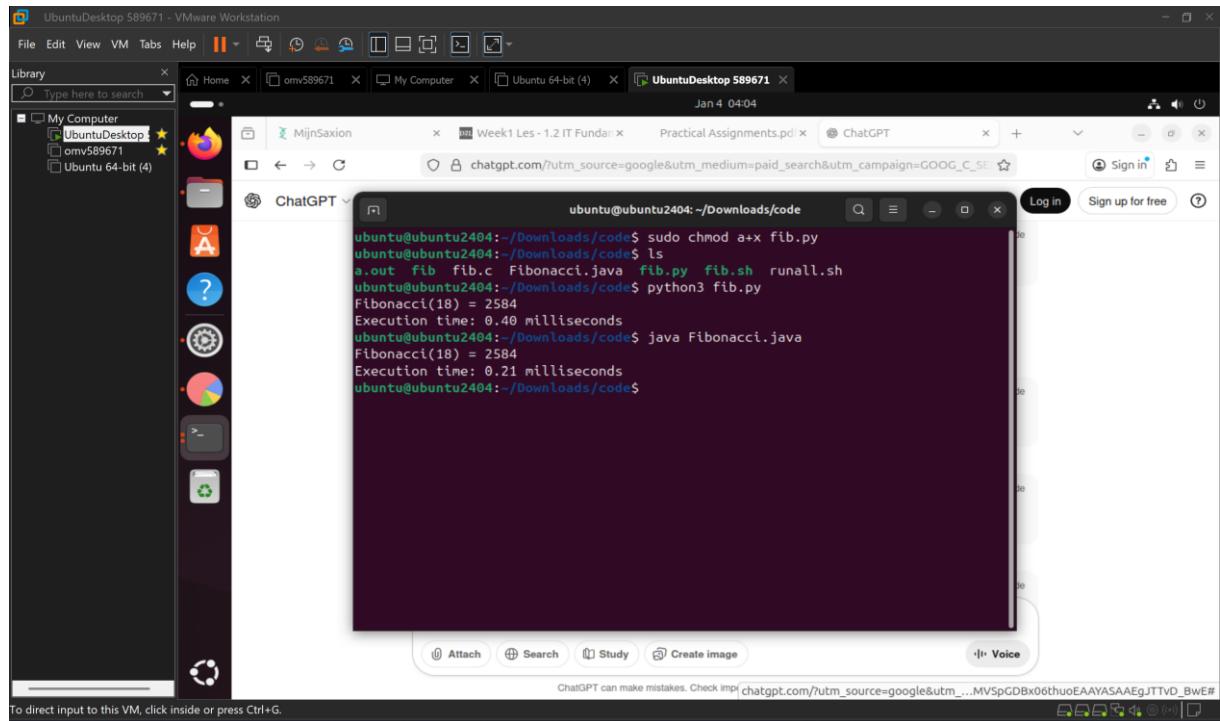
## BASH



## Python



# Java



#### 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.

-o2

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

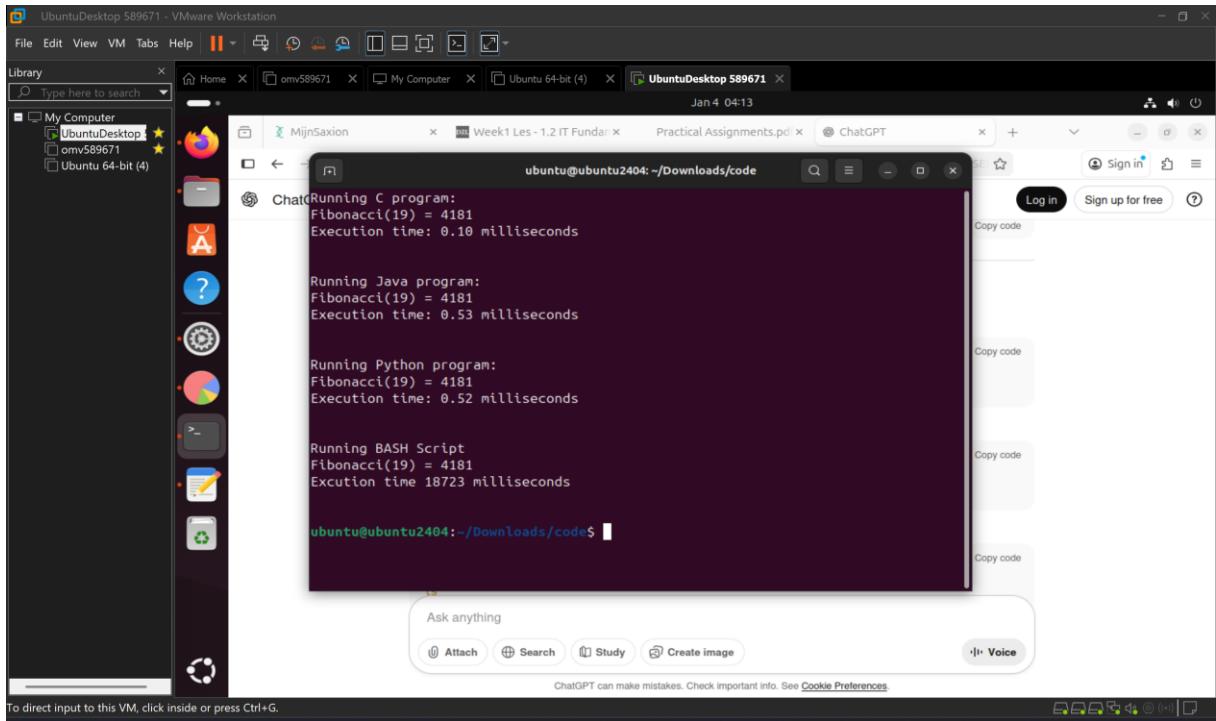
The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "ubuntu@ubuntu2404: ~/Downloads/code". The terminal content shows the following command and its output:

```
ubuntu@ubuntu2404:~/Downloads/code$ gcc -O2 fib.c -o fib_opt
ubuntu@ubuntu2404:~/Downloads/code$ ls
a.out fib fib.c Fibonacci.java fib_opt fib.py fib.sh runall.sh
ubuntu@ubuntu2404:~/Downloads/code$ ./fib_opt
Fibonacci(18) = 2584
Execution time: 0.01 milliseconds
ubuntu@ubuntu2404:~/Downloads/code$
```

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

Ja. Hij is veel sneller

- 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.

The screenshot shows the OakSim assembly debugger interface. At the top, there are several tabs: MijnSaxion, INL Week4 Les - 1.2 IT Funda..., OakSim, Practical Assignments.pdf, and ARM Assembly Power Co... . The main window title is "wunkolo.github.io/OakSim/". Below the tabs, there are browser navigation buttons and a search bar with the placeholder "Verify it's you". The central area is divided into sections: a control panel with "Run" (highlighted), "Step", and "Reset" buttons; assembly code; a register table; and memory dump.

**Control Panel:**

- Open
- Run (highlighted)
- 250
- Step
- Reset

**Assembly Code:**

```
1 Main:  
2     mov r1, #8  
3     mov r2, #2  
4     mov r0, #1  
5  
6 Loop:  
7     cmp r2, #0  
8     beq End  
9  
10    mul r0, r0, r1  
11    sub r2, r2, #1  
12    b Loop  
13  
14 End:  
15
```

**Register Table:**

Register	Value
R0	40
R1	8
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0
RR	0

**Memory Dump:**

Address	Value
0x000010000:	08 10 A0 E3 02 20 A0 E3 01 00 A0 E3 00 00 52 E3
0x000010010:	02 00 00 DA 90 01 E0 01 20 42 E2 FA FF F7 EA
0x000010020:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010030:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010040:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010050:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010060:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010070:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010080:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010090:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000100A0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000100B0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000100C0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000100D0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000100E0:	00 C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000100F0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010100:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010110:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010120:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010130:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010140:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010150:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010160:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000010170:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Ready? Save this file and export it as a pdf file with the name: [week4.pdf](#)