

# Template Week 4 - Software

Student number:582840

## Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:

The screenshot shows the OakSim software interface. On the left, the assembly code for calculating a factorial is displayed:

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

On the right, the register values are listed:

Register	Value
R0	0
R1	78
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10	0
R11	0
R12	0
SP	10000
LR	10028
PC	10028
CPSR	60000013

Below the registers is a memory dump window showing memory addresses from 0x00010000 to 0x00015000. The dump area is mostly filled with zeros, with some non-zero values appearing at higher addresses.

## Assignment 4.2: Programming languages

Take screenshots that the following commands work:

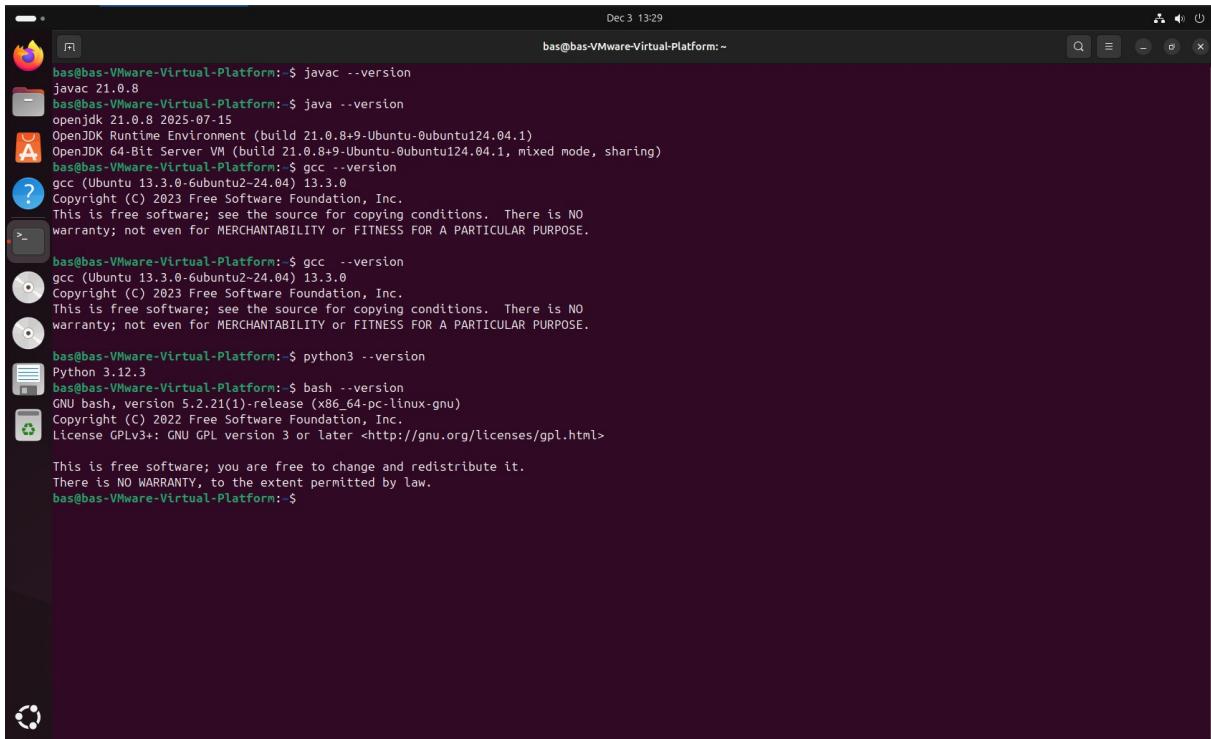
javac --version

java --version

gcc --version

python3 --version

bash --version



```
Dec 3 13:29
bas@bas-VMware-Virtual-Platform:~$ javac --version
javac 21.0.8
bas@bas-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.8 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu124.04.1, mixed mode, sharing)
bas@bas-VMware-Virtual-Platform:~$ gcc --version
gcc (Ubuntu 13.3.0-0ubuntu2-24.04) 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.

bas@bas-VMware-Virtual-Platform:~$ gcc --version
gcc (Ubuntu 13.3.0-0ubuntu2-24.04) 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.

bas@bas-VMware-Virtual-Platform:~$ python3 --version
Python 3.12.3
bas@bas-VMware-Virtual-Platform:~$ 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.
bas@bas-VMware-Virtual-Platform:~$
```

### pAssignment 4.3: Compile

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

Fib.c en Fibonacci.java

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

Fib.c

Which source code files are compiled to byte code?

Fibonacci.java

Which source code files are interpreted by an interpreter?

Fib.py en 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

How do I run a Java program?

javac Fibonacci.java && java Fibonacci

How do I run a Python program?

```
python3 fib.py
```

How do I run a C program?

```
gcc fib.c -o fib
```

How do I run a Bash script?

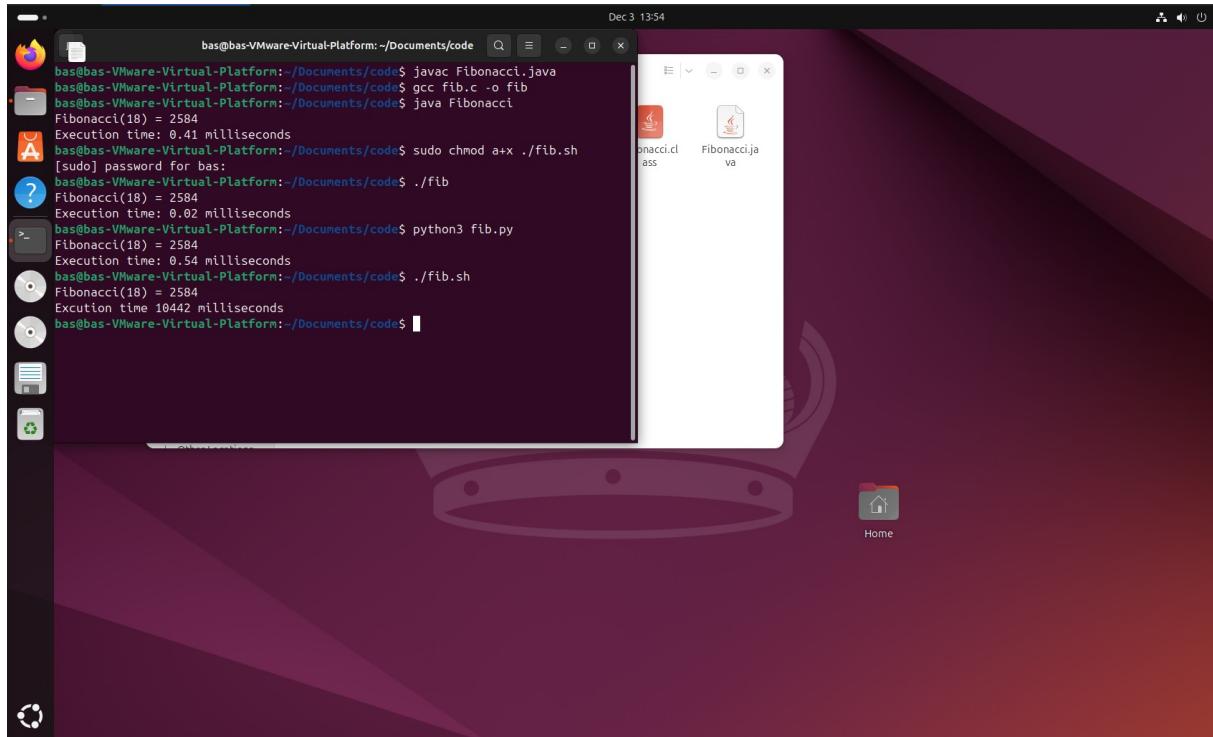
```
./fib.sh
```

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

Ja een fib bestand voor C.

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 screenshot shows a Linux desktop environment with a dark purple background. A terminal window is open in the foreground, displaying the following command-line session:

```
bas@bas-VMware-Virtual-Platform:~/Documents/code$ javac Fibonacci.java
bas@bas-VMware-Virtual-Platform:~/Documents/code$ gcc fib.c -o fib
Fibonacci(18) = 2584
Execution time: 0.41 milliseconds
bas@bas-VMware-Virtual-Platform:~/Documents/code$ sudo chmod a+x ./fib.sh
[sudo] password for bas:
bas@bas-VMware-Virtual-Platform:~/Documents/code$ ./fib
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds
bas@bas-VMware-Virtual-Platform:~/Documents/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.54 milliseconds
bas@bas-VMware-Virtual-Platform:~/Documents/code$ ./fib.sh
Fibonacci(18) = 2584
Execution time: 10442 milliseconds
bas@bas-VMware-Virtual-Platform:~/Documents/code$
```

A file manager window is visible in the background, showing two files: "Fibonacci.class" and "Fibonacci.java".

C is het snelst daarna java dan python en dan scripting.

#### Assignment 4.4: Optimize

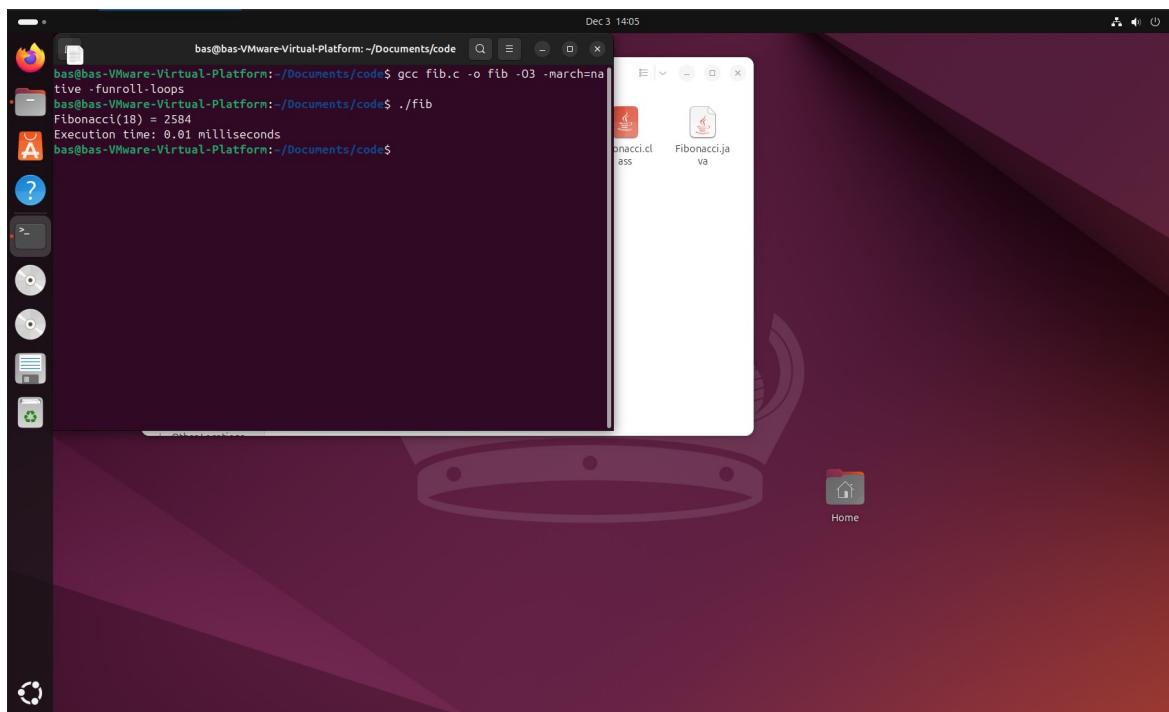
Take relevant screenshots of the following commands:

- 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.

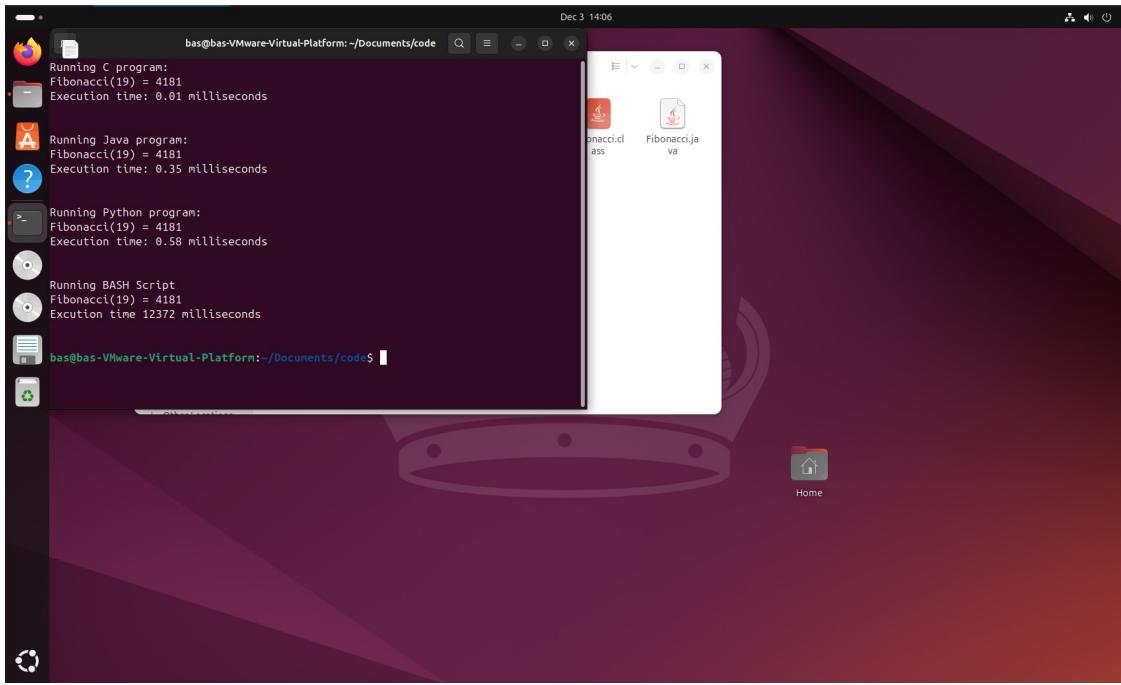
```
gcc fib.c -o fib -O3 -march=native -funroll-loops
```

- 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?  
Ja hij doet het nu onder de 0.01 miliseconden



- 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.

OakSim

Open Run 250 Step Reset

```
2 Main:  
3     mov r1, #2  
4     mov r2, #4  
5     mov r8, #1  
6 Loop:  
7     mul r8, r2, r1  
8     sub r2, r2, #1  
9     CMP r2, #0  
10    bne Loop  
11 End:  
12
```

Register	Value
R0	10
R1	2
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10	0
R11	0
R12	0
SF	10000
LR	0
PC	10030

0x00010000: 02 10 A0 B3 04 20 A0 E1 21 A0 E1 90 01 00 00  
0x00010001: 01 20 42 E2 00 00 02 E3 FB FF FF 14 00 00 00 B...R  
0x00010002: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010003: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010004: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010005: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010006: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010007: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010008: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010009: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001000A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001000B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001000C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001000D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001000E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001000F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010011: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010012: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010013: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010014: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010015: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010017: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010018: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010019: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001001B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001001C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001001D: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001001E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x0001001F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010021: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010022: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010023: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010024: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010025: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010026: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010027: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x00010281: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0x000102A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

© 2017

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