

# Template Week 4 – Software

Student number: 588898

## Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:

The screenshot shows a debugger interface with two main panes. The left pane displays the assembly code for a factorial calculation. The right pane shows the register values.

**Assembly Code:**

```
Open Run 250 Step Reset
1 Main:
2     mov r2, #5
3     mov r1, #1
4
5 Loop:
6     mul r1, r1, r2 ; r1 = r1 * r2
7     sub r2, r2, #1
8     cmp r2, #1      ; als r2 == 1, stop
9     beq End         ; anders opnieuw loopen
10    b Loop          ; anders opnieuw loopen
11
12 End:
```

**Registers:**

Register	Value
R0	0
R1	5
R2	5
R3	0
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10	0

**Memory Dump:**

Address	Value
0x00010000	05 20 A0 E3 01 10 A0 E3 91 02 01 E0 01 00 52 E3 . . . R.
0x00010010	FF FF FF OA 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
0x00010030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

## Assignment 4.2: Programming languages

Take screenshots that the following commands work:

javac –version

```
sven@sven-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
```

java –version

```
sven@sven-VMware-Virtual-Platform:~$ 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)
```

gcc –version

```
sven@sven-VMware-Virtual-Platform:~$ 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.
```

python3 –version

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

bash --version

```
sven@sven-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.
```

### **Assignment 4.3: Compile**

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

C en java

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

C en assembly

Which source code files are compiled to byte code?

Java

Which source code files are interpreted by an interpreter?

Python, Bash

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

C

How do I run a Java program?

Javac programma.java

Java programma

How do I run a Python program?

Python3 programma.py

How do I run a C program?

Gcc main.c -o main

./main

How do I run a Bash script?

Bash programma.sh

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

Bij bash en python niet, maar bij java komt er een .class uit en bij c/gcc komt er a.out of main uit.

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?

#### 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.
- Compile **fib.c** again with the optimization parameters
- Run the newly compiled program. Is it true that it now performs the calculation faster?
- 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**