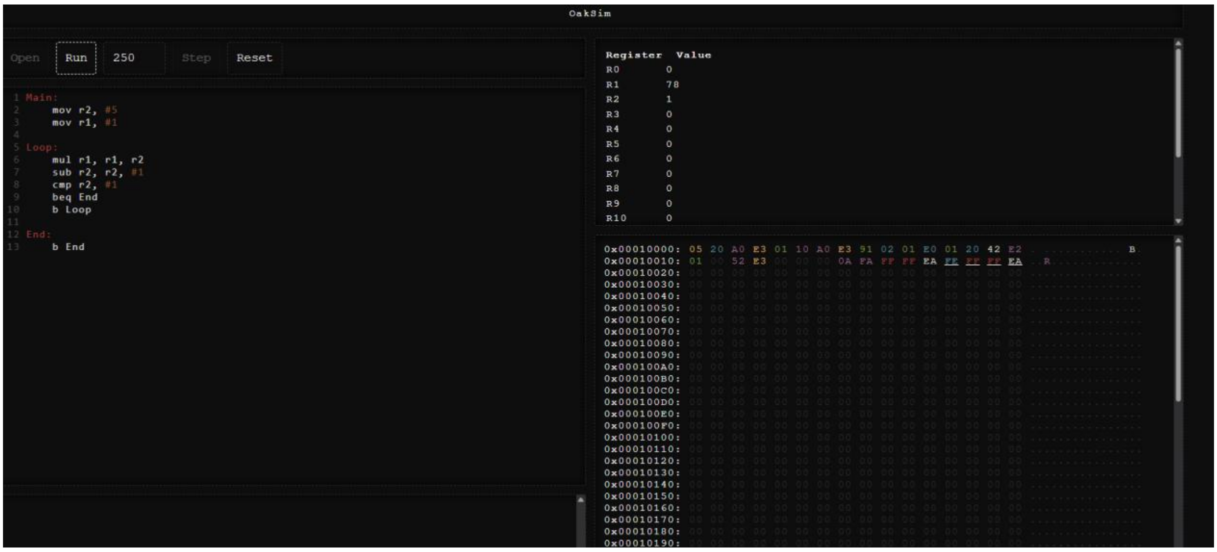
Template Week 4 – Software

Student number: 528668

Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:



```
Main:
    mov r2, #5
    mov r1, #1

Loop:
    mul r1, r1, r2
    sub r2, r2, #1
    cmp r2, #1
    beq End
    b Loop

End:
    b End
```

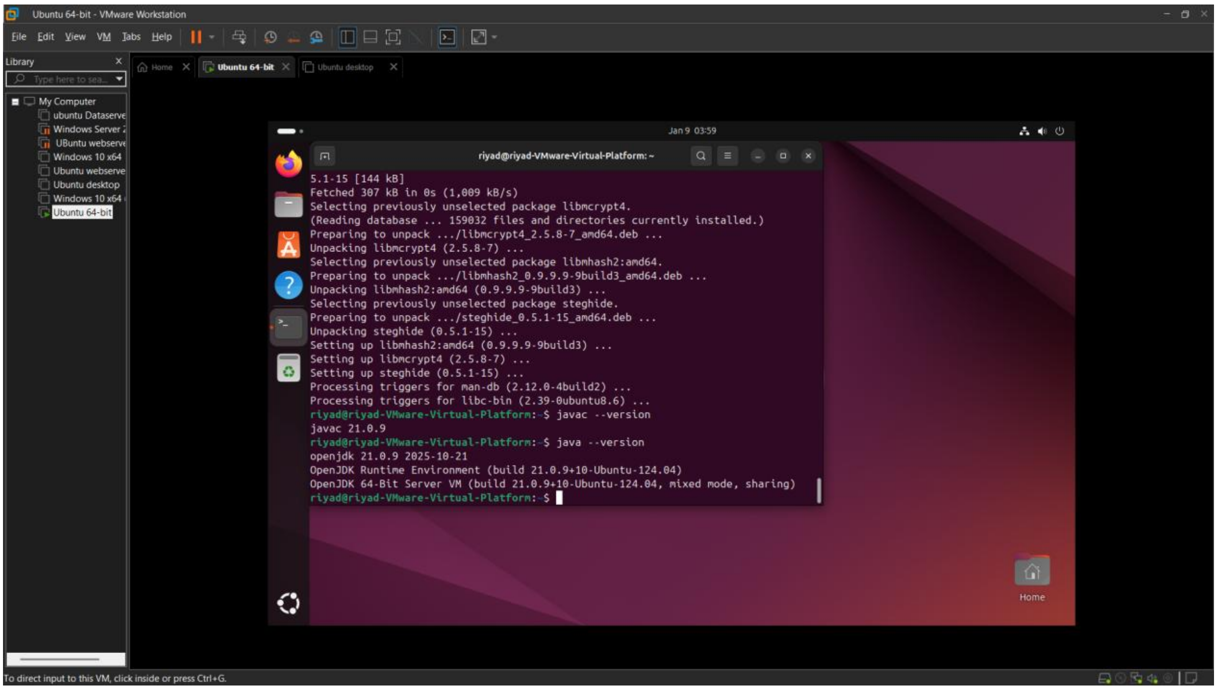
```
Register  Value
R0        0
R1        78
R2        1
R3        0
R4        0
R5        0
R6        0
R7        0
R8        0
R9        0
R10       0
```

```
0x00010000: 05 20 A0 E3 01 10 A0 E3 91 02 01 E0 01 20 42 E2  . ...........B.
0x00010010: 01 00 52 E3 00 00 00 0A FA FF FF EA FE FF FF EA  ..R.............
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  ................
0x00010170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00010180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00010190:
```

Assignment 4.2: Programming languages

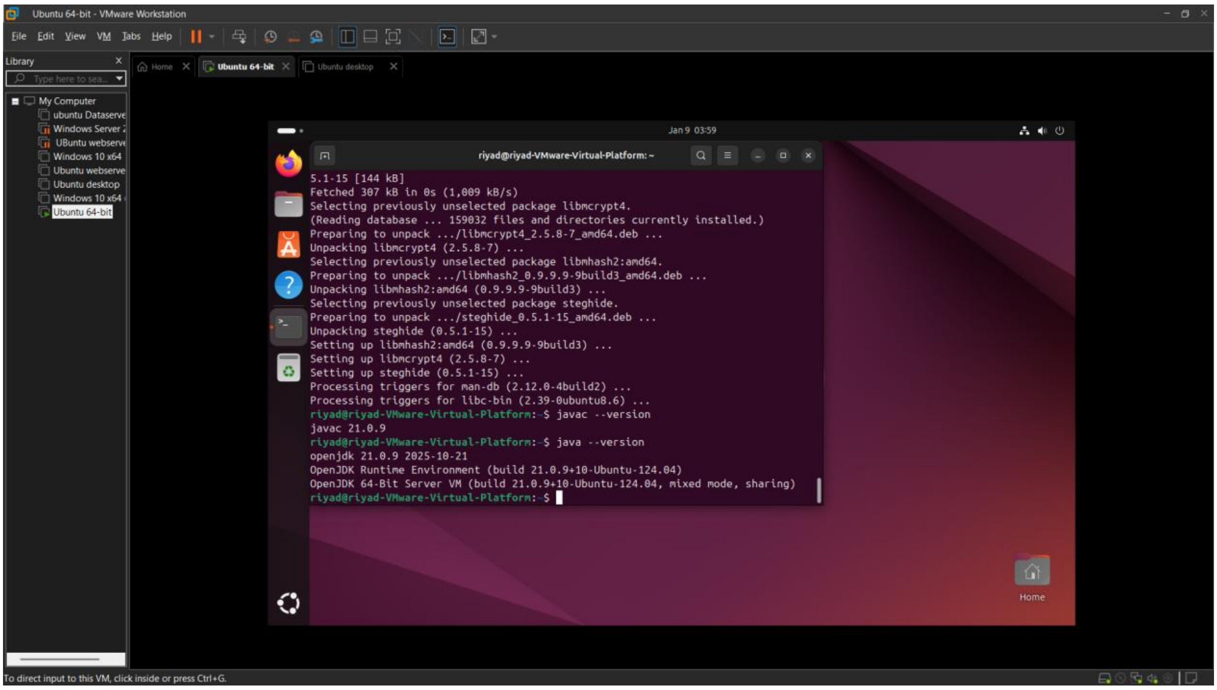
Take screenshots that the following commands work:

javac –version



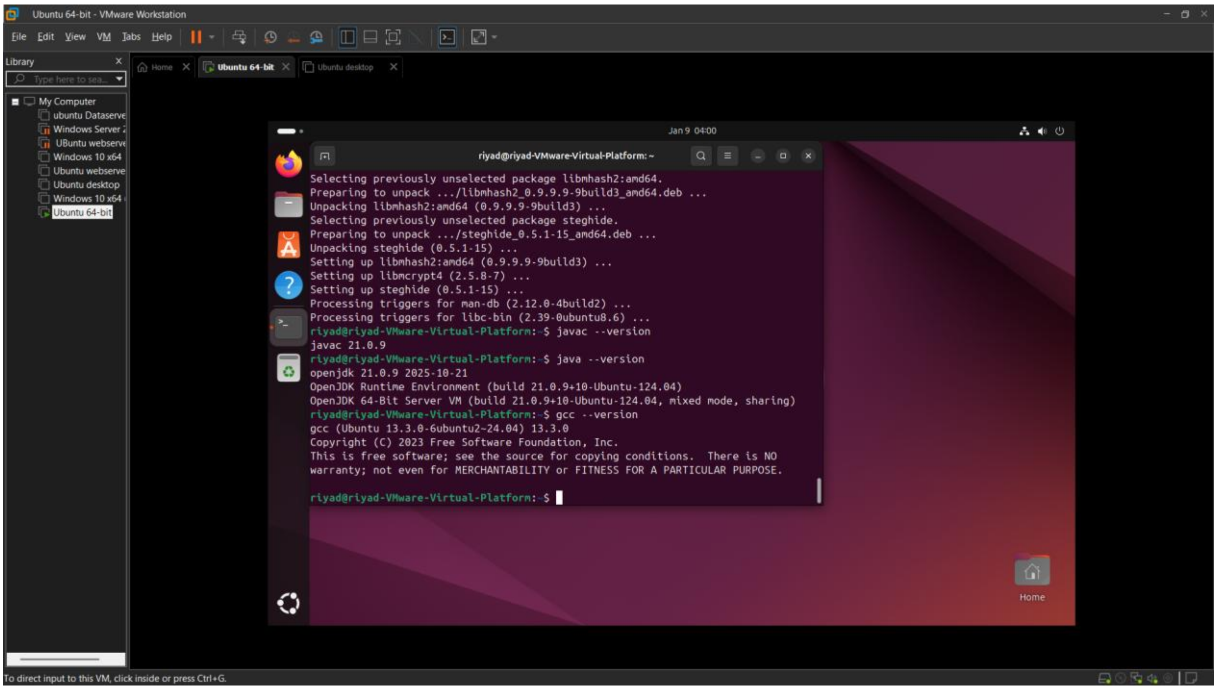
```
5.1-15 [144 kB]
Fetched 307 kB in 0s (1,009 kB/s)
Selecting previously unselected package libmcrypt4.
(Reading database ... 159032 files and directories currently installed.)
Preparing to unpack .../libmcrypt4_2.5.8-7_amd64.deb ...
Unpacking libmcrypt4 (2.5.8-7) ...
Selecting previously unselected package libmhash2:amd64.
Preparing to unpack .../libmhash2_0.9.9.9-9build3_amd64.deb ...
Unpacking libmhash2:amd64 (0.9.9.9-9build3) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-15_amd64.deb ...
Unpacking steghide (0.5.1-15) ...
Setting up libmhash2:amd64 (0.9.9.9-9build3) ...
Setting up libmcrypt4 (2.5.8-7) ...
Setting up steghide (0.5.1-15) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
riyad@riyad-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
riyad@riyad-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)
riyad@riyad-VMware-Virtual-Platform:~$
```

java –version



```
5.1-15 [144 kB]
Fetched 307 kB in 0s (1,009 kB/s)
Selecting previously unselected package libmcrypt4.
(Reading database ... 159032 files and directories currently installed.)
Preparing to unpack .../libmcrypt4_2.5.8-7_amd64.deb ...
Unpacking libmcrypt4 (2.5.8-7) ...
Selecting previously unselected package libmhash2:amd64.
Preparing to unpack .../libmhash2_0.9.9.9-9build3_amd64.deb ...
Unpacking libmhash2:amd64 (0.9.9.9-9build3) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-15_amd64.deb ...
Unpacking steghide (0.5.1-15) ...
Setting up libmhash2:amd64 (0.9.9.9-9build3) ...
Setting up libmcrypt4 (2.5.8-7) ...
Setting up steghide (0.5.1-15) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
riyad@riyad-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
riyad@riyad-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)
riyad@riyad-VMware-Virtual-Platform:~$
```

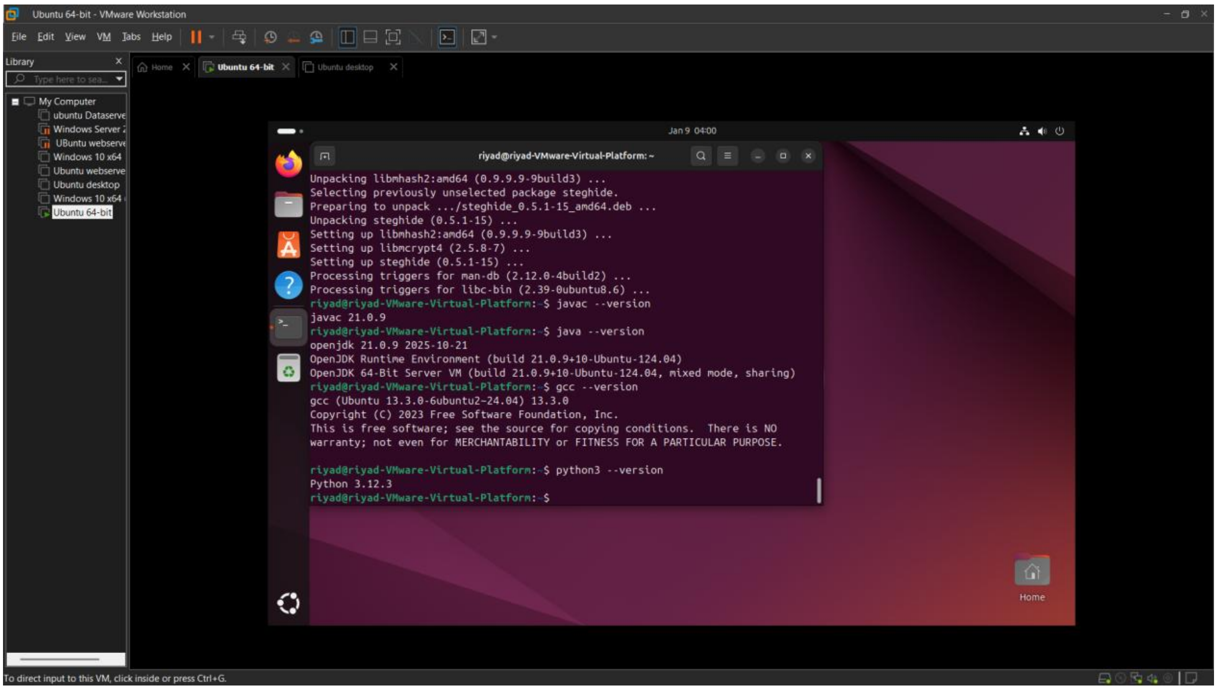
gcc –version



```
Selecting previously unselected package libmhash2:amd64.
Preparing to unpack .../libmhash2_0.9.9.9-9build3_amd64.deb ...
Unpacking libmhash2:amd64 (0.9.9.9-9build3) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-15_amd64.deb ...
Unpacking steghide (0.5.1-15) ...
Setting up libmhash2:amd64 (0.9.9.9-9build3) ...
Setting up libmcrypt4 (2.5.8-7) ...
Setting up steghide (0.5.1-15) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
riyad@riyad-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
riyad@riyad-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)
riyad@riyad-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.

riyad@riyad-VMware-Virtual-Platform:~$
```

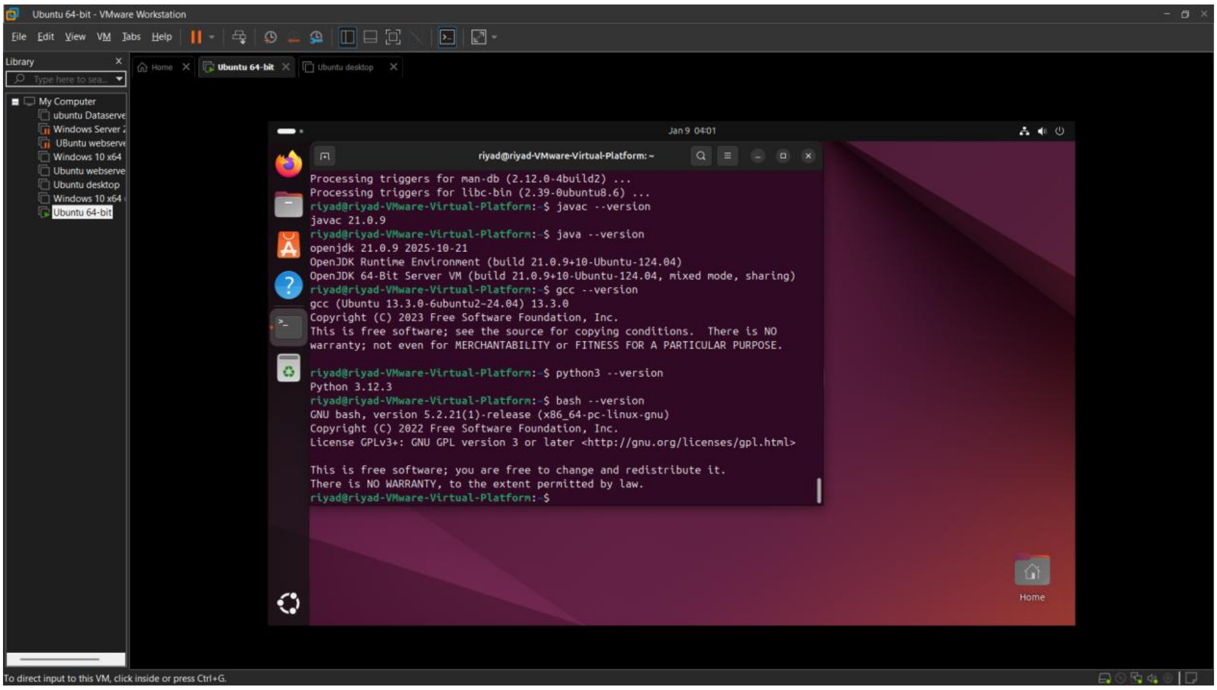
python3 –version



```
Unpacking libmhash2:amd64 (0.9.9.9-9build3) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-15_amd64.deb ...
Unpacking steghide (0.5.1-15) ...
Setting up libmhash2:amd64 (0.9.9.9-9build3) ...
Setting up libmcrypt4 (2.5.8-7) ...
Setting up steghide (0.5.1-15) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
riyad@riyad-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
riyad@riyad-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)
riyad@riyad-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.

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

bash –version



```
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
riyad@riyad-VMware-Virtual-Platform:~$ javac --version
javac 21.0.9
riyad@riyad-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)
riyad@riyad-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.

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

Assignment 4.3: Compile

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

De C-file (.c) en de Java-file (.java).

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

De C-file. De compiler vertaalt dit direct naar instructies voor de processor.

Which source code files are compiled to byte code?

De Java-file. Dit wordt vertaald naar .class bestanden die door de Java Virtual Machine (JVM) worden gelezen.

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?

Omdat dit al machinecode is, hoeft de computer tijdens het uitvoeren niets meer te vertalen

How do I run a Java program?

javac Program.java (compileren) en dan java Program (runnen).

How do I run a Python program?

python3 Program.py

How do I run a C program?

gcc Program.c -o Program (compileren) en dan ./Program (runnen).

How do I run a Bash script?

bash Program.sh of ./Program.sh na chmod +x.

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

Ja. Bij Java krijg je een .class bestand. Bij C krijg je een executable (bijv. a.out of een .exe op Windows)

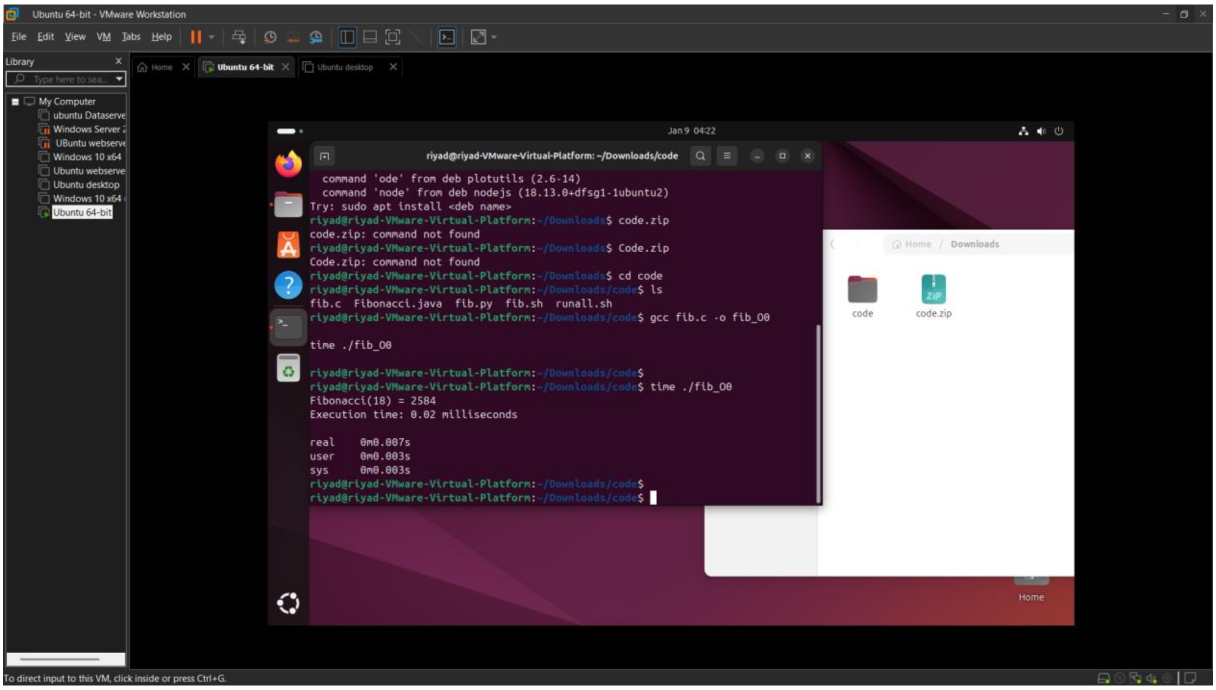
Take relevant screenshots of the following commands:

- 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:

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.



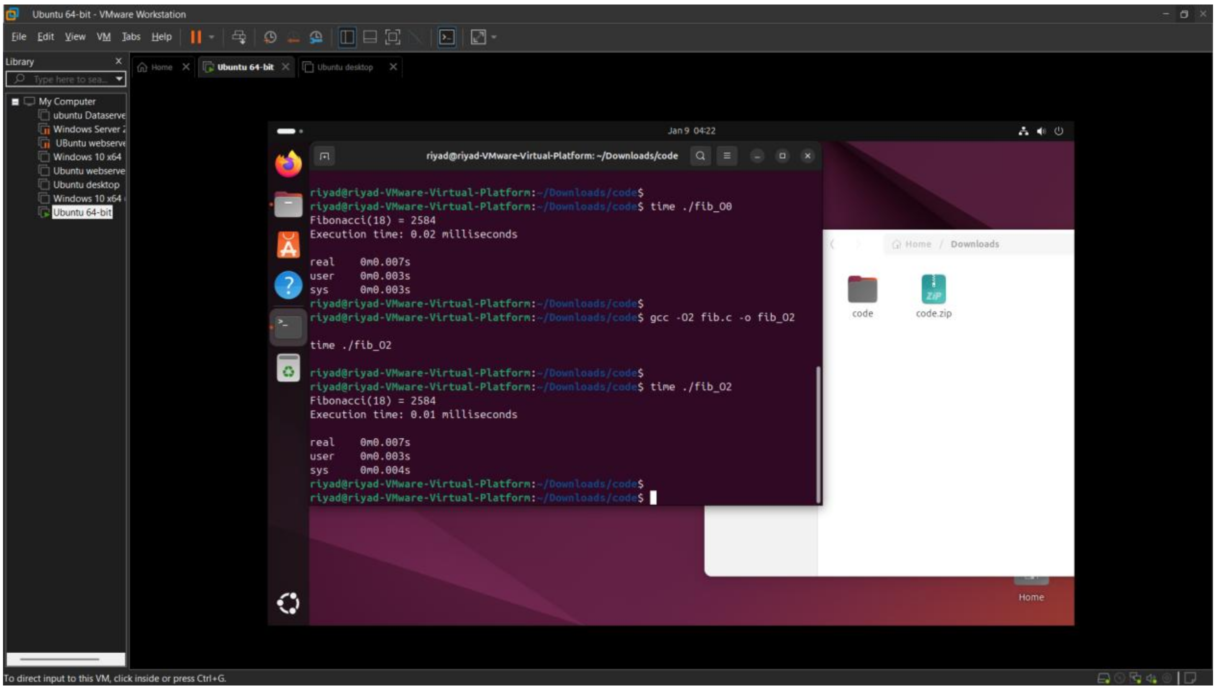
```
  command 'ode' from deb plotutils (2.6-14)
  command 'node' from deb nodejs (18.13.0+dfsg1-1ubuntu2)
Try: sudo apt install <deb name>
riyad@riyad-VMware-Virtual-Platform:~/Downloads$ code.zip
code.zip: command not found
riyad@riyad-VMware-Virtual-Platform:~/Downloads$ Code.zip
Code.zip: command not found
riyad@riyad-VMware-Virtual-Platform:~/Downloads$ cd code
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$ ls
fib.c  Fibonacci.java  fib.py  fib.sh  runall.sh
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$ gcc fib.c -o fib_O0

time ./fib_O0

riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$ time ./fib_O0
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds

real    0m0.007s
user    0m0.003s
sys     0m0.003s
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
```

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



```
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$ time ./fib_O0
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds

real    0m0.007s
user    0m0.003s
sys     0m0.003s
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$ gcc -O2 fib.c -o fib_O2

time ./fib_O2

riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$ time ./fib_O2
Fibonacci(18) = 2584
Execution time: 0.01 milliseconds

real    0m0.007s
user    0m0.003s
sys     0m0.004s
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
riyad@riyad-VMware-Virtual-Platform:~/Downloads/code$
```

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

Het programma met -O2 is sneller dan zonder optimalisatie (-O0), omdat de compiler extra optimalisaties toepast. Hierdoor wordt de machinecode efficiënter en hoeft de CPU minder werk te doen.”

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.

Het programma dat is gecompileerd met -O2 is sneller omdat de compiler tijdens het compileren extra optimalisaties toepast. De compiler verwijdert onnodige instructies, herschikt berekeningen en maakt de machinecode efficiënter. Hierdoor hoeft de processor minder stappen uit te voeren tijdens het uitvoeren van het programma, wat resulteert in een kortere uitvoeringstijd.

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.

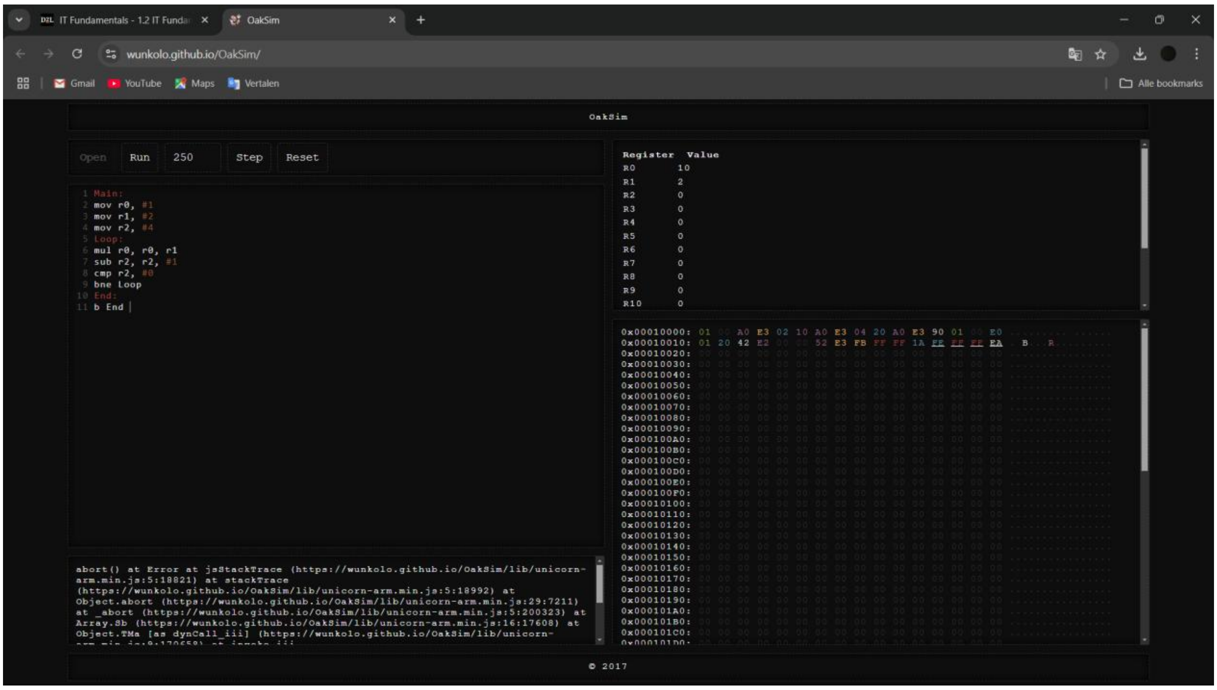
```
Main:
mov r0, #1
mov r1, #2
mov r2, #4

Loop:
mul r0, r0, r1
sub r2, r2, #1
cmp r2, #0
bne Loop

End:
b End
```

Complete the code. See the PowerPoint slides of week 4.

Screenshot of the completed code here.



```
Main:
mov r0, #1
mov r1, #2
mov r2, #4
Loop:
mul r0, r0, r1
sub r2, r2, #1
cmp r2, #0
bne Loop
End:
b End
```

```
Register  Value
R0        10
R1        2
R2        0
R3        0
R4        0
R5        0
R6        0
R7        0
R8        0
R9        0
R10       0
```

```
0x00010000: 01 00 A0 E3 02 10 A0 E3 04 20 A0 E3 90 01 00 E0 ................
0x00010010: 01 20 42 E2 00 00 52 E3 FB FF FF 1A FE FF FF EA ..B..R..........
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 ................
0x00010170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x00010180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x00010190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x000101A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x000101B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x000101C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x000101D0:
```

```
abort() at Error at jsStackTrace (https://wunkolo.github.io/OakSim/lib/unicorn-
arm.min.js:5:18821) at stackTrace
(https://wunkolo.github.io/OakSim/lib/unicorn-arm.min.js:5:18992) at
Object.abort (https://wunkolo.github.io/OakSim/lib/unicorn-arm.min.js:29:7211)
at _abort (https://wunkolo.github.io/OakSim/lib/unicorn-arm.min.js:5:200323) at
Array.Sb (https://wunkolo.github.io/OakSim/lib/unicorn-arm.min.js:16:17608) at
Object.TMa [as dynCall_iii] (https://wunkolo.github.io/OakSim/lib/unicorn-
arm.min.js:9:170658) at invoke_iii
```

Ready? Save this file and export it as a pdf file with the name: **week4.pdf**