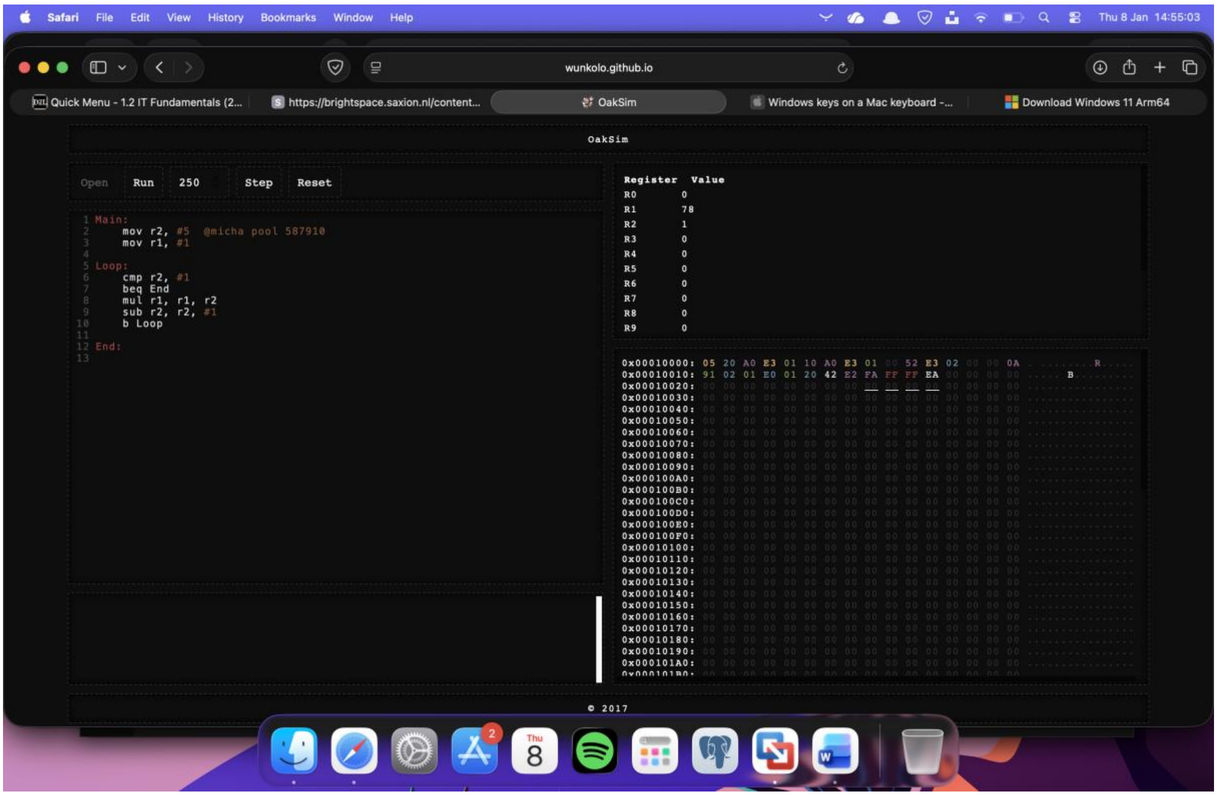
Template Week 4 – Software

Student number: 587910

Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:



Safari File Edit View History Bookmarks Window Help
Thu 8 Jan 14:55:03
wunkolo.github.io
Quick Menu - 1.2 IT Fundamentals (2...
https://brightspace.saxion.nl/content...
OakSim
Windows keys on a Mac keyboard -...
Download Windows 11 Arm64
OakSim
Open
Run
250
Step
Reset
© 2017

```
Main:
    mov r2, #5  @micha pool 587910
    mov r1, #1

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

End:
```

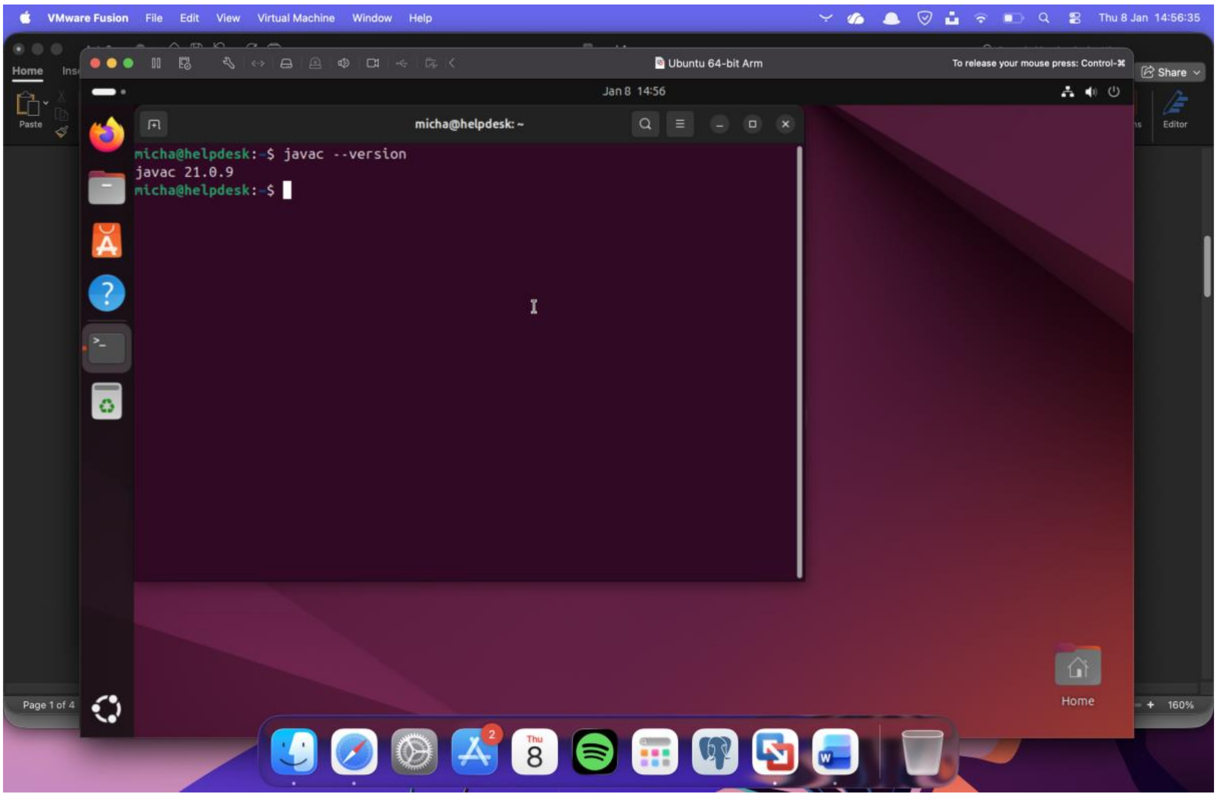
```
Register  Value
R0        0
R1        78
R2        1
R3        0
R4        0
R5        0
R6        0
R7        0
R8        0
R9        0
```

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

Assignment 4.2: Programming languages

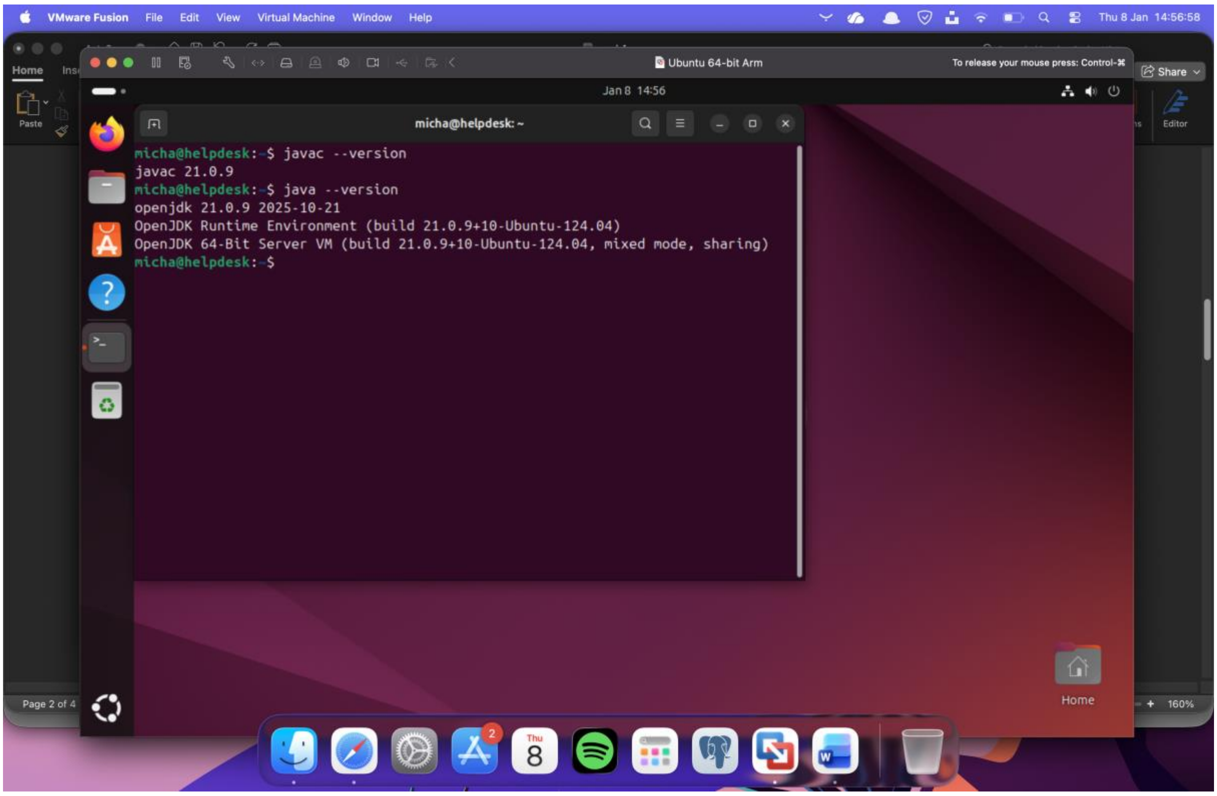
Take screenshots that the following commands work:

javac –version


VMware Fusion File Edit View Virtual Machine Window Help
Thu 8 Jan 14:56:35
Ubuntu 64-bit Arm
To release your mouse press: Control-⌘
Jan 8 14:56
micha@helpdesk: ~
Home

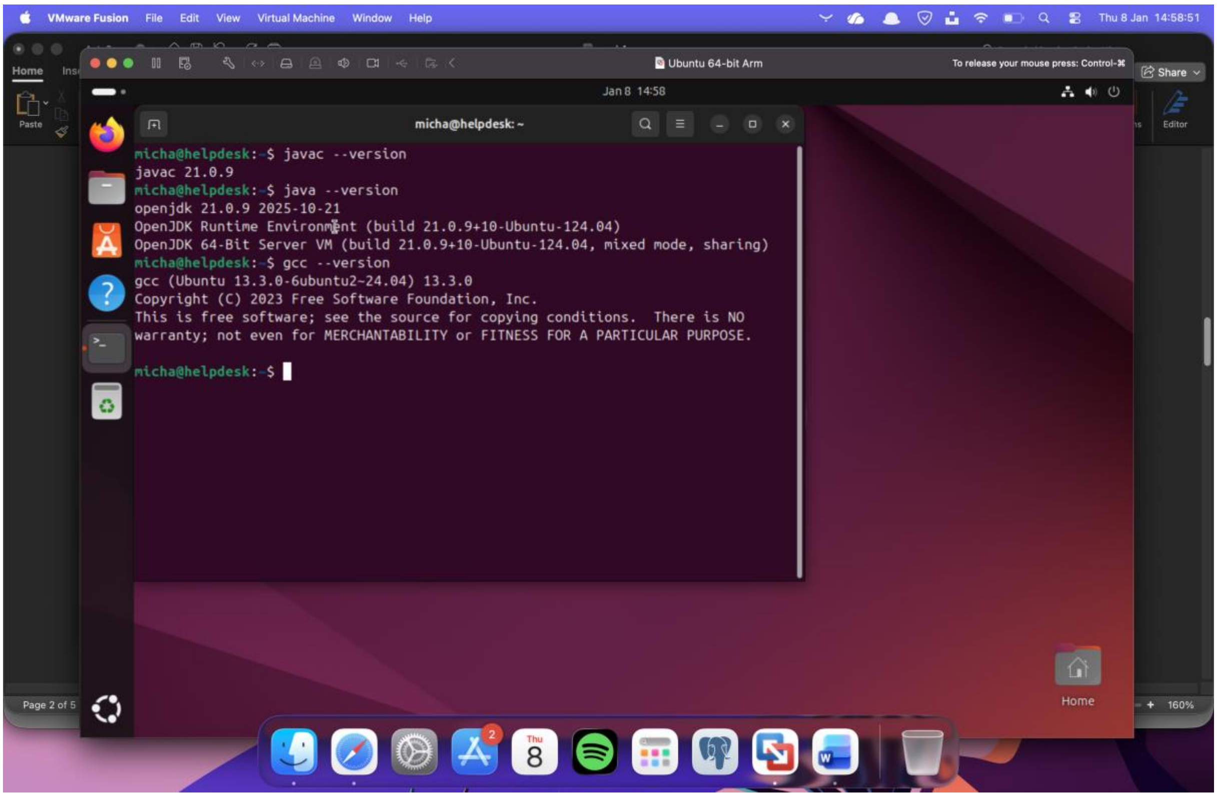
```
micha@helpdesk:~$ javac --version
javac 21.0.9
micha@helpdesk:~$ 
```

java –version


VMware Fusion File Edit View Virtual Machine Window Help
Thu 8 Jan 14:56:58
Ubuntu 64-bit Arm
To release your mouse press: Control-⌘
Jan 8 14:56
micha@helpdesk: ~
Home

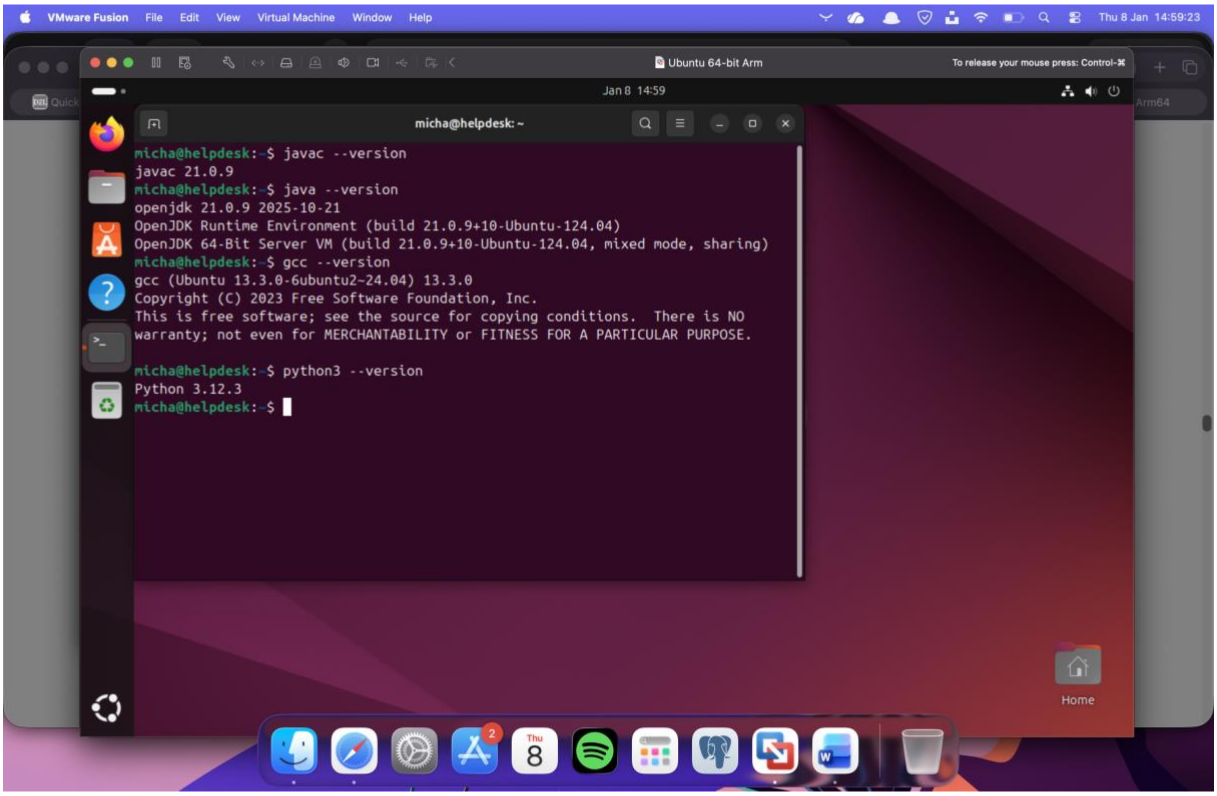
```
micha@helpdesk:~$ javac --version
javac 21.0.9
micha@helpdesk:~$ java --version
openjdk 21.0.9 2025-10-21
OpenJDK Runtime Environment (build 21.0.9+10-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 21.0.9+10-Ubuntu-124.04, mixed mode, sharing)
micha@helpdesk:~$
```

gcc –version


VMware Fusion File Edit View Virtual Machine Window Help
Thu 8 Jan 14:58:51
Ubuntu 64-bit Arm
To release your mouse press: Control-⌘
Jan 8 14:58
micha@helpdesk: ~
Home
Page 2 of 5
160%

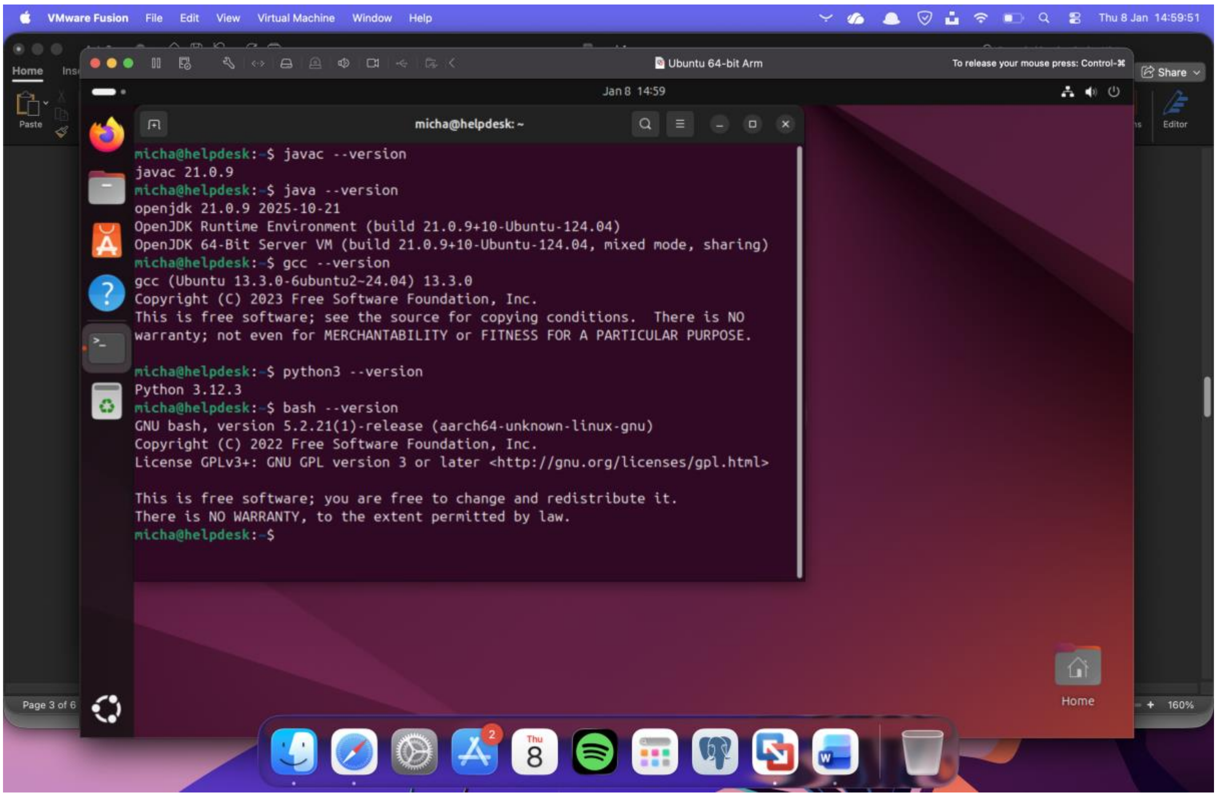
```
micha@helpdesk:~$ javac --version
javac 21.0.9
micha@helpdesk:~$ java --version
openjdk 21.0.9 2025-10-21
OpenJDK Runtime Environment (build 21.0.9+10-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 21.0.9+10-Ubuntu-124.04, mixed mode, sharing)
micha@helpdesk:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 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.

micha@helpdesk:~$ 
```

python3 –version


VMware Fusion File Edit View Virtual Machine Window Help
Thu 8 Jan 14:59:23
Ubuntu 64-bit Arm
To release your mouse press: Control-⌘
Jan 8 14:59
micha@helpdesk: ~
Home

```
micha@helpdesk:~$ javac --version
javac 21.0.9
micha@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)
micha@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.

micha@helpdesk:~$ python3 --version
Python 3.12.3
micha@helpdesk:~$ 
```

bash –version



```
micha@helpdesk:~$ javac --version
javac 21.0.9
micha@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)
micha@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.

micha@helpdesk:~$ python3 --version
Python 3.12.3
micha@helpdesk:~$ bash --version
GNU bash, version 5.2.21(1)-release (aarch64-unknown-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.
micha@helpdesk:~$
```

Assignment 4.3: Compile

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

De C-bestanden en Java-bestanden moeten altijd gecompileerd worden voordat je ze kunt uitvoeren.

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

C-bestanden. De C-compiler vertaalt de broncode direct naar machinecode die de processor van je computer direct begrijpt.

Which source code files are compiled to byte code?

Java-bestanden. De Java-compiler vertaalt de broncode naar bytecode.

Which source code files are interpreted by an interpreter?

Python-bestanden en Bash-scripts. Een interpreter leest en voert deze code regel voor regel uit tijdens het draaien.

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

C is waarschijnlijk het snelst. Omdat de code direct is vertaald naar de taal van de processor, is er geen extra vertaling nodig tijdens het uitvoeren.

How do I run a Java program?

Java wordt gecompileerd naar bytecode en daarna uitgevoerd door de Java Virtual Machine.

Compileren: Typ javac Naam.java. Dit maakt een .class bestand aan.

Uitvoeren: Typ java Naam

How do I run a Python program?

Python is een geïnterpreteerde taal, dus je hoeft zelf niets te compileren.

Uitvoeren: Typ python3 naam.py.

How do I run a C program?

C is een gecompileerde taal. Je moet de code eerst vertalen naar machinecode.

Compileren: Typ gcc -o mijnprogramma naam.c. Dit maakt een nieuw bestand aan genaamd mijnprogramma.

Uitvoeren: Typ ./mijnprogramma om het programma te starten.

How do I run a Bash script?

Net als Python wordt een Bash-script direct gelezen door een interpreter.

Uitvoeren: Typ bash script.sh.

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

Ja, bij compilatie ontstaan er nieuwe bestanden op je computer:

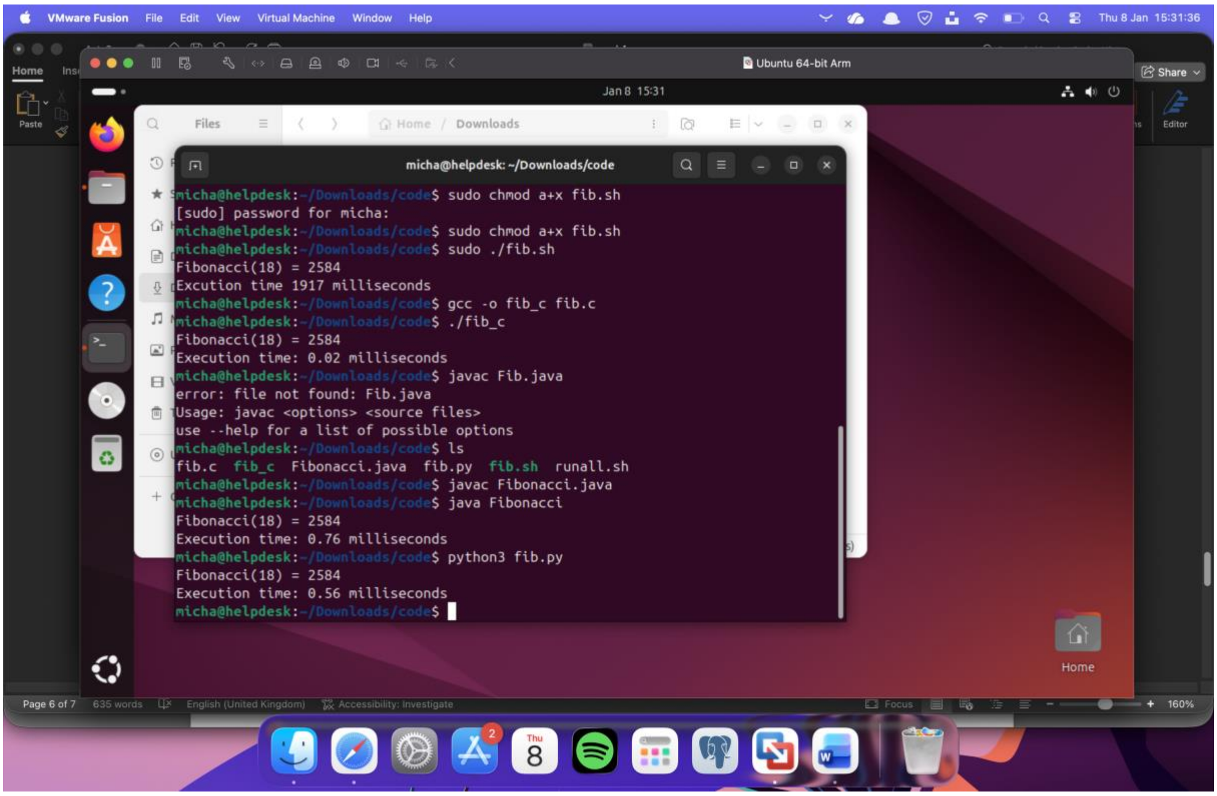
Bij C: Er wordt een executable aangemaakt. Als je gcc -o output naam.c gebruikt, heet het nieuwe bestand output.

Bij Java: Er wordt een .class bestand aangemaakt. Dit bestand bevat de bytecode.

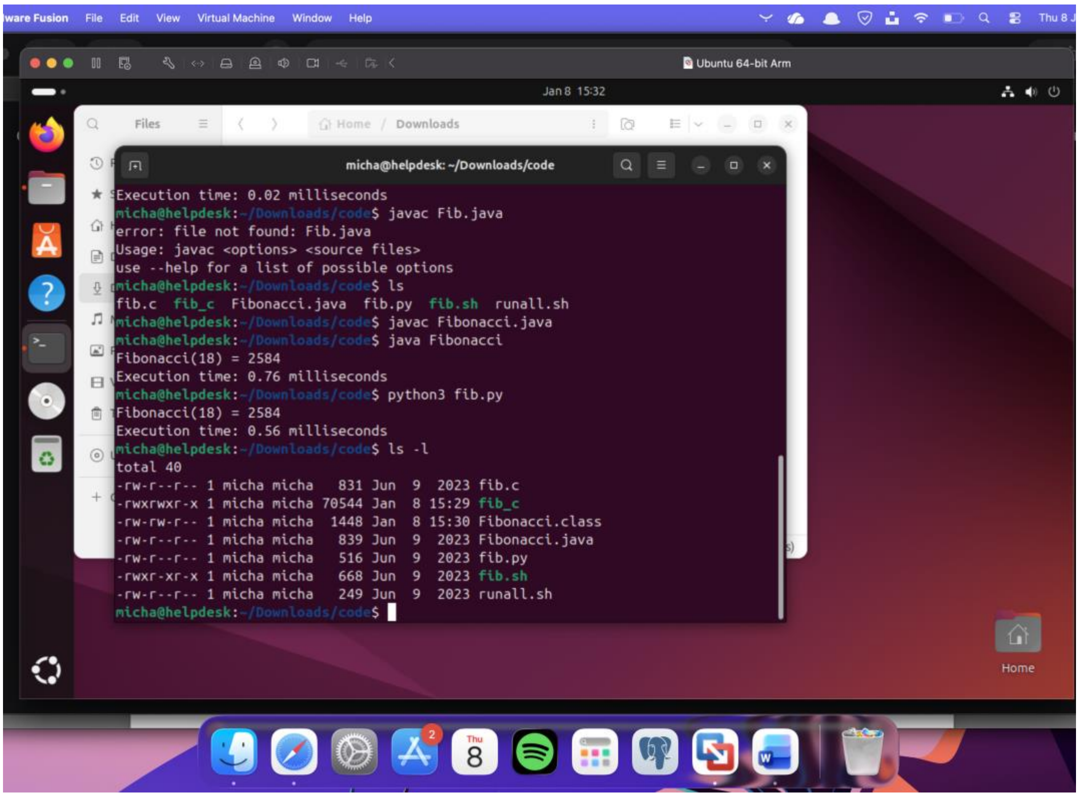
Bij Python/Bash: Er worden normaal gesproken geen nieuwe zichtbare bestanden aangemaakt; de interpreter voert de broncode direct uit.

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?



```
micha@helpdesk:~/Downloads/code$ sudo chmod a+x fib.sh
[sudo] password for micha:
micha@helpdesk:~/Downloads/code$ sudo chmod a+x fib.sh
micha@helpdesk:~/Downloads/code$ sudo ./fib.sh
Fibonacci(18) = 2584
Excution time 1917 milliseconds
micha@helpdesk:~/Downloads/code$ gcc -o fib_c fib.c
micha@helpdesk:~/Downloads/code$ ./fib_c
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds
micha@helpdesk:~/Downloads/code$ javac Fib.java
error: file not found: Fib.java
Usage: javac <options> <source files>
use --help for a list of possible options
micha@helpdesk:~/Downloads/code$ ls
fib.c  fib_c  Fibonacci.java  fib.py  fib.sh  runall.sh
micha@helpdesk:~/Downloads/code$ javac Fibonacci.java
micha@helpdesk:~/Downloads/code$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.76 milliseconds
micha@helpdesk:~/Downloads/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.56 milliseconds
micha@helpdesk:~/Downloads/code$
```



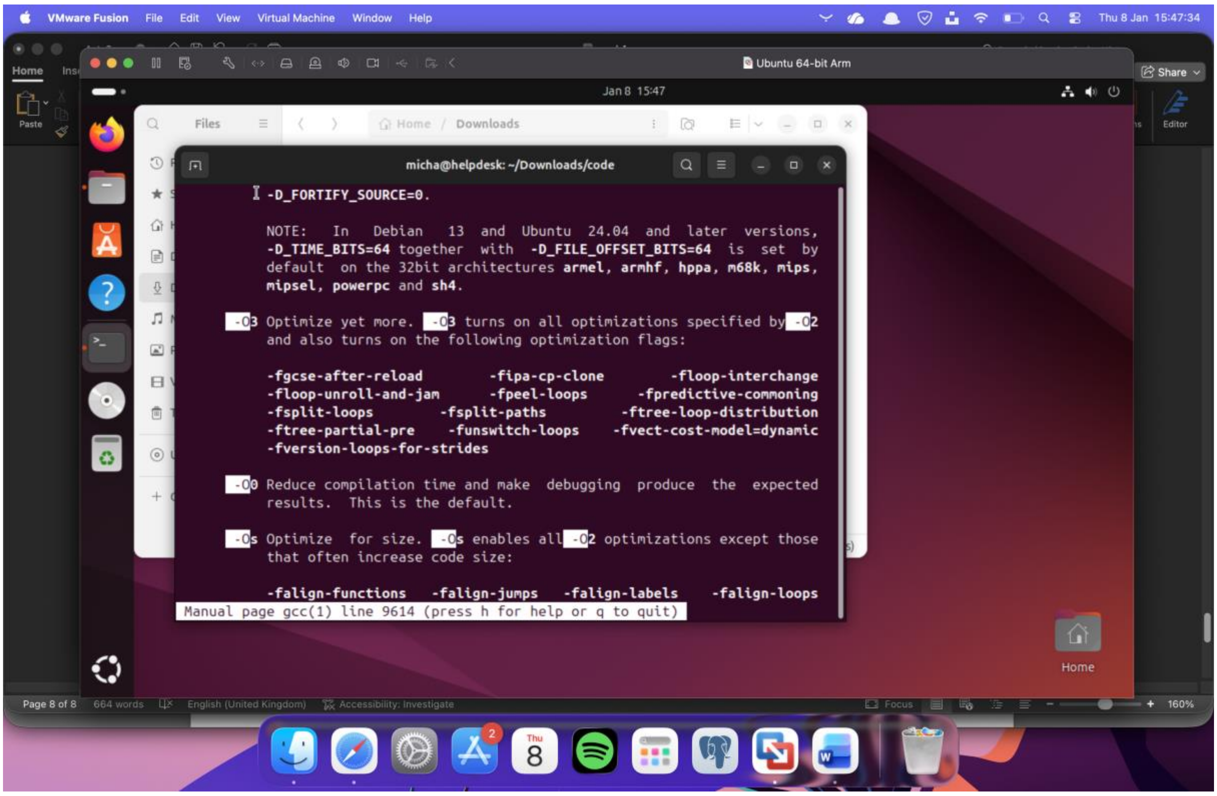
```
Execution time: 0.02 milliseconds
micha@helpdesk:~/Downloads/code$ javac Fib.java
error: file not found: Fib.java
Usage: javac <options> <source files>
use --help for a list of possible options
micha@helpdesk:~/Downloads/code$ ls
fib.c  fib_c  Fibonacci.java  fib.py  fib.sh  runall.sh
micha@helpdesk:~/Downloads/code$ javac Fibonacci.java
micha@helpdesk:~/Downloads/code$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.76 milliseconds
micha@helpdesk:~/Downloads/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.56 milliseconds
micha@helpdesk:~/Downloads/code$ ls -l
total 40
-rw-r--r-- 1 micha micha   831 Jun  9  2023 fib.c
-rwxrwxr-x 1 micha micha 70544 Jan  8 15:29 fib_c
-rw-rw-r-- 1 micha micha  1448 Jan  8 15:30 Fibonacci.class
-rw-r--r-- 1 micha micha   839 Jun  9  2023 Fibonacci.java
-rw-r--r-- 1 micha micha   516 Jun  9  2023 fib.py
-rwxr-xr-x 1 micha micha   668 Jun  9  2023 fib.sh
-rw-r--r-- 1 micha micha   249 Jun  9  2023 runall.sh
micha@helpdesk:~/Downloads/code$
```

Je ziet in de screenshots dat C het snelste is met 0.02 milliseconden en dat er bij de tweede screenshot twee keer een ‘x’ achter staat wat executable betekent.

Assignment 4.4: Optimize

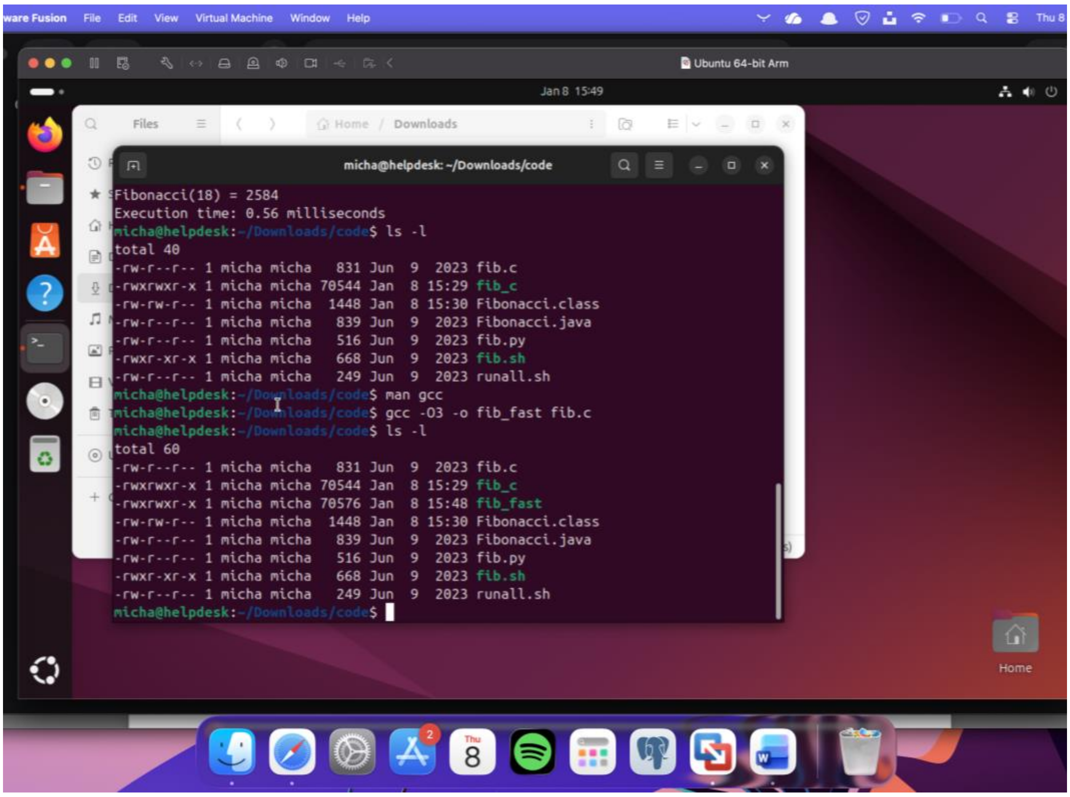
Take relevant screenshots of the following commands:

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.



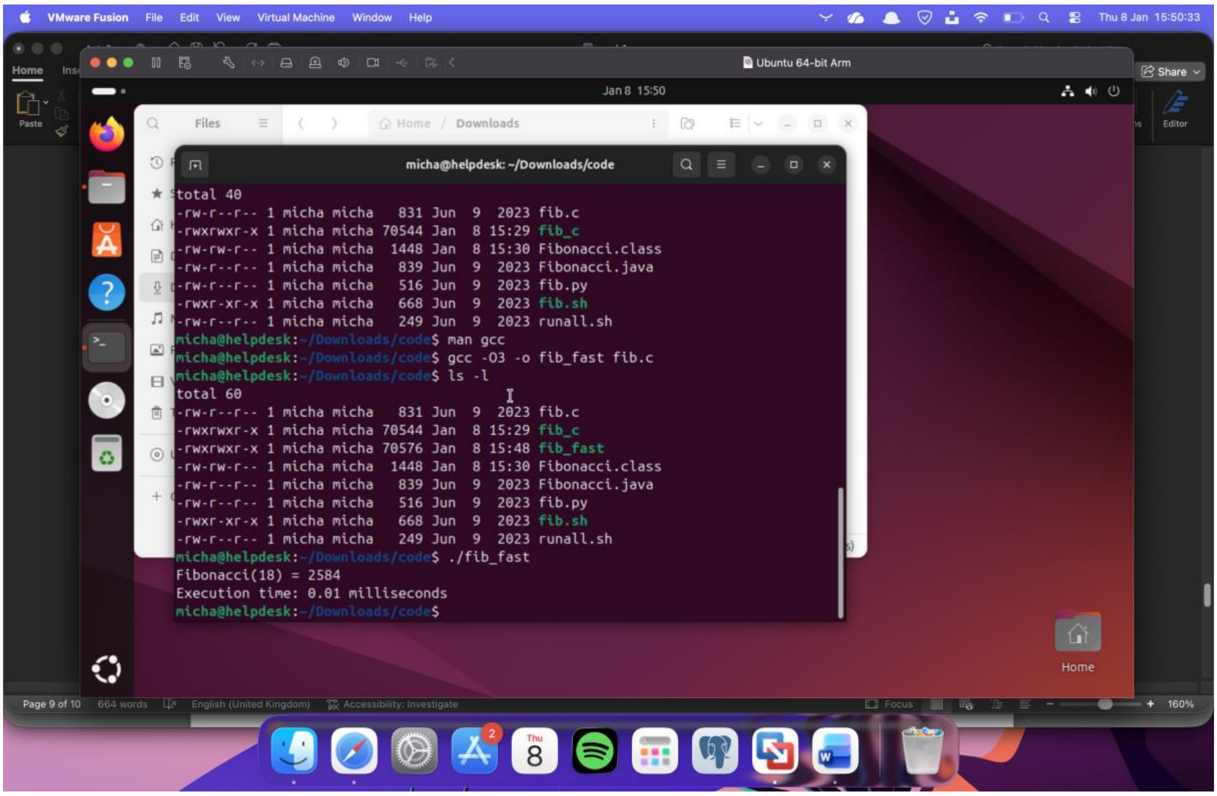
```
    -D_FORTIFY_SOURCE=0.

    NOTE:  In  Debian  13 and Ubuntu 24.04 and later versions,
    -D_TIME_BITS=64 together with -D_FILE_OFFSET_BITS=64 is set by
    default  on the 32bit architectures armel, armhf, hppa, m68k, mips,
    mipsel, powerpc and sh4.

-O3 Optimize yet more. -O3 turns on all optimizations specified by -O2
    and also turns on the following optimization flags:

    -fgcse-after-reload        -fipa-cp-clone        -floop-interchange
    -floop-unroll-and-jam      -fpeel-loops      -fpredictive-commoning
    -fsplit-loops          -fsplit-paths        -ftree-loop-distribution
    -ftree-partial-pre    -funswitch-loops    -fvect-cost-model=dynamic
    -fversion-loops-for-strides

-O0 Reduce compilation time and make debugging produce the expected
    results.  This is the default.

-Os Optimize  for size. -Os enables all -O2 optimizations except those
    that often increase code size:

    -falign-functions   -falign-jumps   -falign-labels   -falign-loops
Manual page gcc(1) line 9614 (press h for help or q to quit)
```

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



```
Fibonacci(18) = 2584
Execution time: 0.56 milliseconds
micha@helpdesk:~/Downloads/code$ ls -l
total 40
-rw-r--r-- 1 micha micha   831 Jun  9  2023 fib.c
-rwxrwxr-x 1 micha micha 70544 Jan  8 15:29 fib_c
-rw-rw-r-- 1 micha micha  1448 Jan  8 15:30 Fibonacci.class
-rw-r--r-- 1 micha micha   839 Jun  9  2023 Fibonacci.java
-rw-r--r-- 1 micha micha   516 Jun  9  2023 fib.py
-rwxr-xr-x 1 micha micha   668 Jun  9  2023 fib.sh
-rw-r--r-- 1 micha micha   249 Jun  9  2023 runall.sh
micha@helpdesk:~/Downloads/code$ man gcc
micha@helpdesk:~/Downloads/code$ gcc -O3 -o fib_fast fib.c
micha@helpdesk:~/Downloads/code$ ls -l
total 60
-rw-r--r-- 1 micha micha   831 Jun  9  2023 fib.c
-rwxrwxr-x 1 micha micha 70544 Jan  8 15:29 fib_c
-rwxrwxr-x 1 micha micha 70576 Jan  8 15:48 fib_fast
-rw-rw-r-- 1 micha micha  1448 Jan  8 15:30 Fibonacci.class
-rw-r--r-- 1 micha micha   839 Jun  9  2023 Fibonacci.java
-rw-r--r-- 1 micha micha   516 Jun  9  2023 fib.py
-rwxr-xr-x 1 micha micha   668 Jun  9  2023 fib.sh
-rw-r--r-- 1 micha micha   249 Jun  9  2023 runall.sh
micha@helpdesk:~/Downloads/code$ 
```

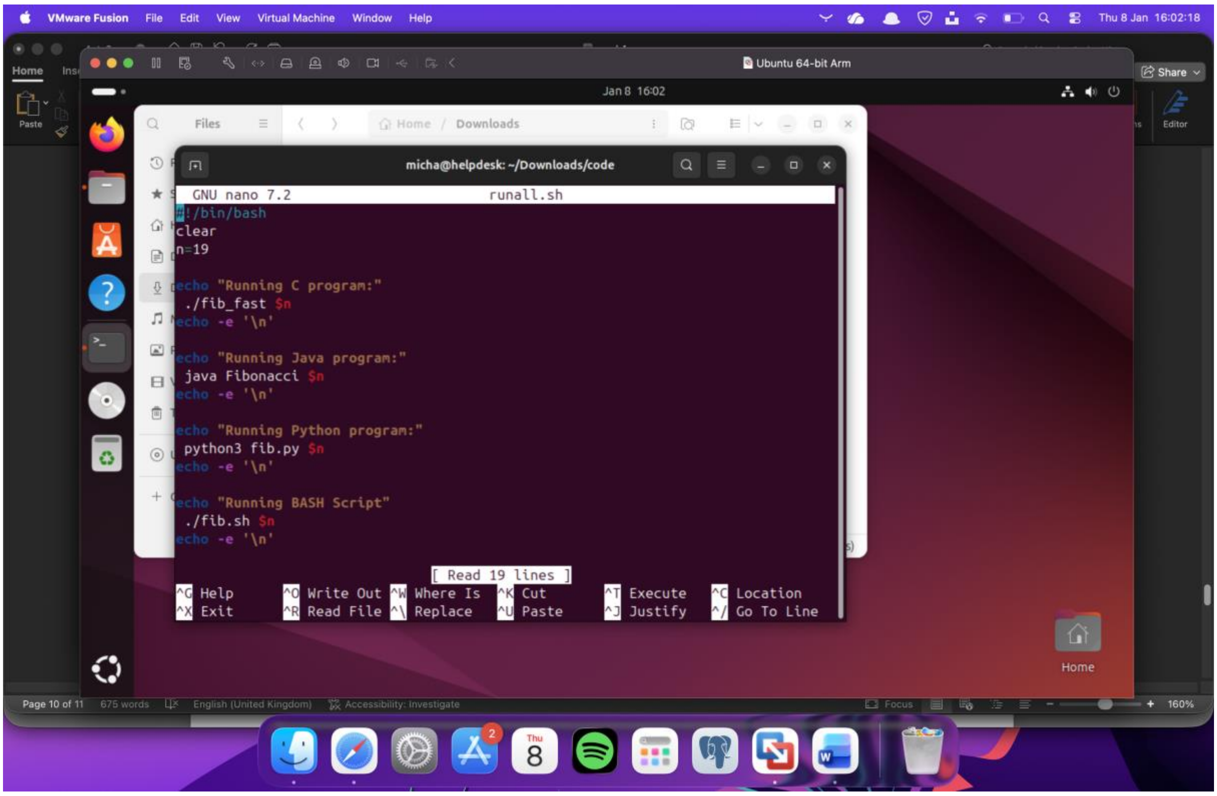
c) Run the newly compiled program. Is it true that it now performs the calculation faster?



```
total 40
-rw-r--r-- 1 micha micha   831 Jun  9  2023 fib.c
-rwxrwxr-x 1 micha micha 70544 Jan  8 15:29 fib_c
-rw-rw-r-- 1 micha micha  1448 Jan  8 15:30 Fibonacci.class
-rw-r--r-- 1 micha micha   839 Jun  9  2023 Fibonacci.java
-rw-r--r-- 1 micha micha   516 Jun  9  2023 fib.py
-rwxr-xr-x 1 micha micha   668 Jun  9  2023 fib.sh
-rw-r--r-- 1 micha micha   249 Jun  9  2023 runall.sh
micha@helpdesk:~/Downloads/code$ man gcc
micha@helpdesk:~/Downloads/code$ gcc -O3 -o fib_fast fib.c
micha@helpdesk:~/Downloads/code$ ls -l
total 60
-rw-r--r-- 1 micha micha   831 Jun  9  2023 fib.c
-rwxrwxr-x 1 micha micha 70544 Jan  8 15:29 fib_c
-rwxrwxr-x 1 micha micha 70576 Jan  8 15:48 fib_fast
-rw-rw-r-- 1 micha micha  1448 Jan  8 15:30 Fibonacci.class
-rw-r--r-- 1 micha micha   839 Jun  9  2023 Fibonacci.java
-rw-r--r-- 1 micha micha   516 Jun  9  2023 fib.py
-rwxr-xr-x 1 micha micha   668 Jun  9  2023 fib.sh
-rw-r--r-- 1 micha micha   249 Jun  9  2023 runall.sh
micha@helpdesk:~/Downloads/code$ ./fib_fast
Fibonacci(18) = 2584
Execution time: 0.01 milliseconds
micha@helpdesk:~/Downloads/code$
```

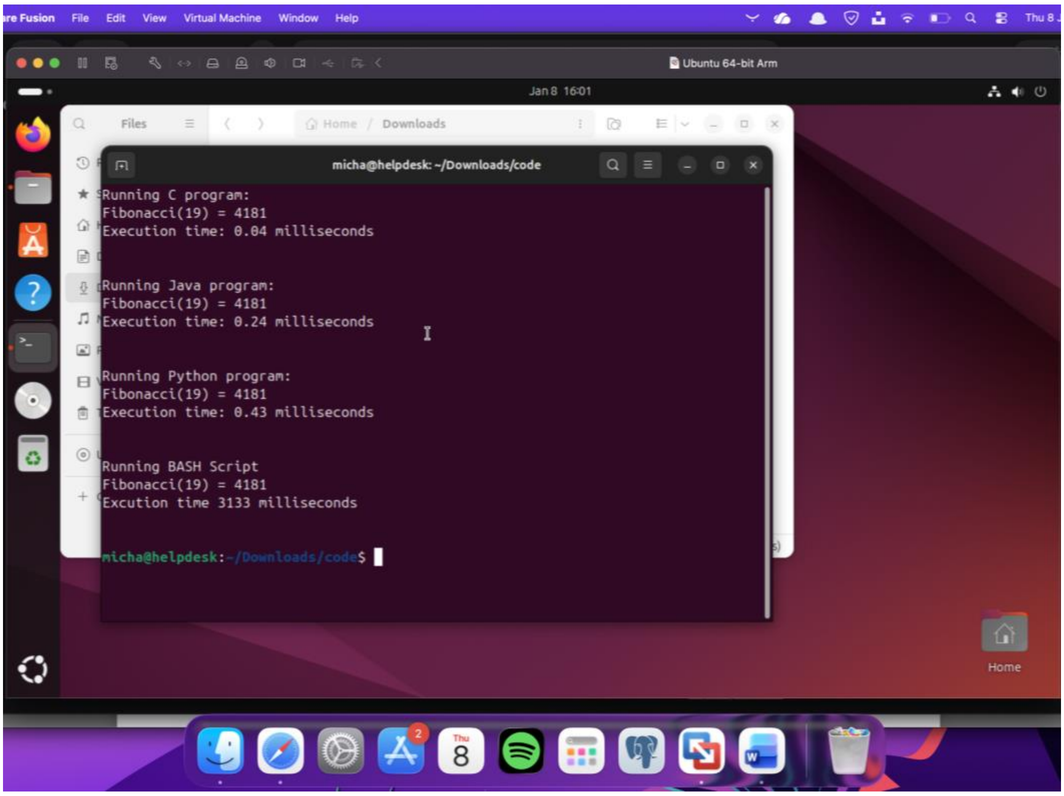
Het was eerst 0.02 milliseconden en nu 0.01 milliseconden, dus sneller.

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.



VMware Fusion File Edit View Virtual Machine Window Help
Thu 8 Jan 16:02:18
Ubuntu 64-bit Arm
Jan 8 16:02
Files
Home / Downloads
micha@helpdesk: ~/Downloads/code
Home

```
GNU nano 7.2                         runall.sh
#!/bin/bash
clear
n=19

echo "Running C program:"
 ./fib_fast $n
echo -e '\n'

echo "Running Java program:"
 java Fibonacci $n
echo -e '\n'

echo "Running Python program:"
 python3 fib.py $n
echo -e '\n'

echo "Running BASH Script"
 ./fib.sh $n
echo -e '\n'

                                [ Read 19 lines ]
^G Help       ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



VMware Fusion File Edit View Virtual Machine Window Help
Ubuntu 64-bit Arm
Jan 8 16:01
Files
Home / Downloads
micha@helpdesk: ~/Downloads/code
Home

```
Running C program:
Fibonacci(19) = 4181
Execution time: 0.04 milliseconds


Running Java program:
Fibonacci(19) = 4181
Execution time: 0.24 milliseconds


Running Python program:
Fibonacci(19) = 4181
Execution time: 0.43 milliseconds


Running BASH Script
Fibonacci(19) = 4181
Excution time 3133 milliseconds


micha@helpdesk:~/Downloads/code$
```

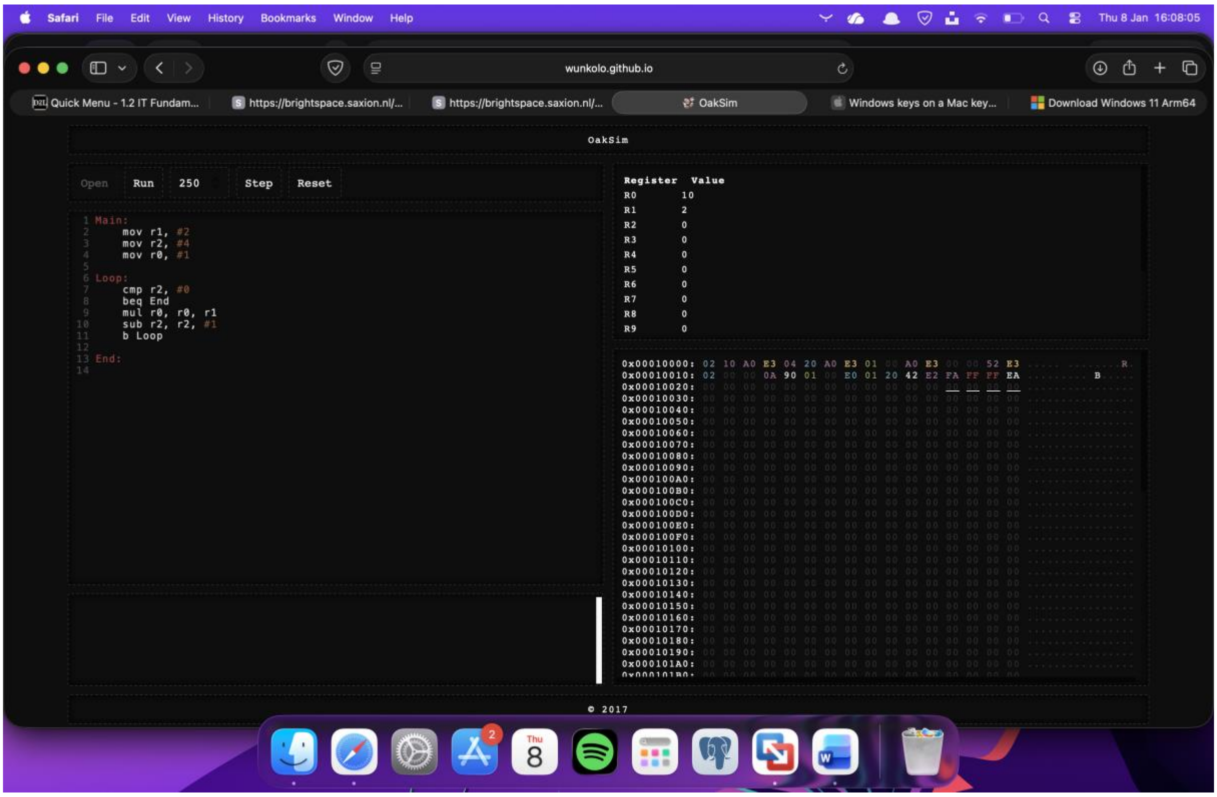
Assignment 4.5: More ARM Assembly

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.


Safari
File
Edit
View
History
Bookmarks
Window
Help
Thu 8 Jan 16:08:05
wunkolo.github.io
Quick Menu - 1.2 IT Fundam...
https://brightspace.saxion.nl/...
https://brightspace.saxion.nl/...
OakSim
Windows keys on a Mac key...
Download Windows 11 Arm64
OakSim
Open
Run
250
Step
Reset
© 2017
Thu
8

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

Loop:
    cmp r2, #0
    beq End
    mul r0, r0, r1
    sub r2, r2, #1
    b Loop

End:
```

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

```
0x00010000: 02 10 A0 E3 04 20 A0 E3 01 00 A0 E3 00 00 52 E3 ..............R.
0x00010010: 02 00 00 0A 90 01 00 E0 01 20 42 E2 FA FF FF EA ......... B.....
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
```

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