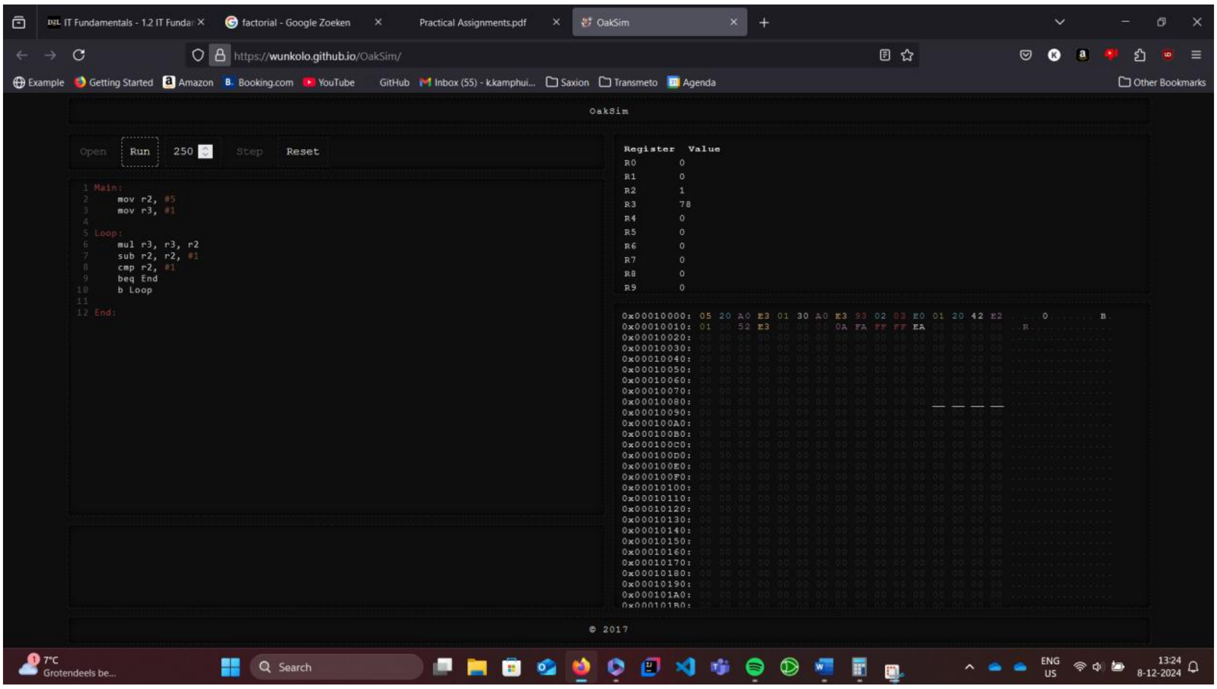
Template Week 4 – Software

Student number: 573190

Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:



```
Main:
    mov r2, #5
    mov r3, #1

Loop:
    mul r3, r3, r2
    sub r2, r2, #1
    cmp r2, #1
    beq End
    b Loop

End:
```

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

```
0x00010000: 05 20 A0 E3 01 30 A0 E3 93 02 03 E0 01 20 42 E2  . ...0....... B.
0x00010010: 01 00 52 E3 00 00 00 0A FA FF FF EA 00 00 00 00  ..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:
```

Assignment 4.2: Programming languages

Take screenshots that the following commands work:

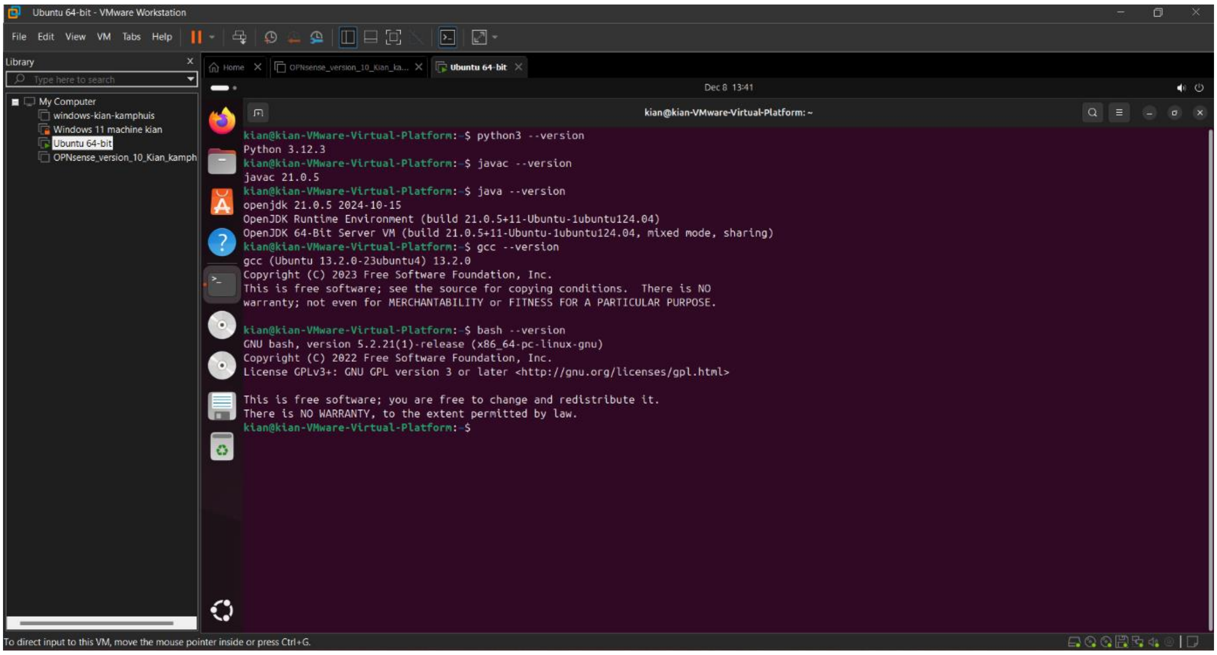
javac –version

java --version

gcc --version

python3 --version

bash --version


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
windows-kian-kamphuis
Windows 11 machine kian
Ubuntu 64-bit
OPNsense_version_10_Kian_kamph
Home
OPNsense_version_10_Kian_ka...
Ubuntu 64-bit
Dec 8 13:41
kian@kian-VMware-Virtual-Platform: ~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
kian@kian-VMware-Virtual-Platform:~$ python3 --version
Python 3.12.3
kian@kian-VMware-Virtual-Platform:~$ javac --version
javac 21.0.5
kian@kian-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, sharing)
kian@kian-VMware-Virtual-Platform:~$ gcc --version
gcc (Ubuntu 13.2.0-23ubuntu4) 13.2.0
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.

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

Assignment 4.3: Compile

Which of the above files need to be compiled before you can run them?
The Java and C files need to be compiled before running.

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

Which source code files are compiled to byte code?
The Java file

Which source code files are interpreted by an interpreter?
The Python and Bash files.

These source code files will perform the same calculation after compilation/interpretation. Which one is expected to do the calculation the fastest?
Generally, the C program (fib.c) is expected to perform the calculation the fastest because it is compiled into machine code, which is directly executed by the processor

How do I run a Java program?
javac Fibonacci.java
java Fibonacci

How do I run a Python program?
python3 fib.py

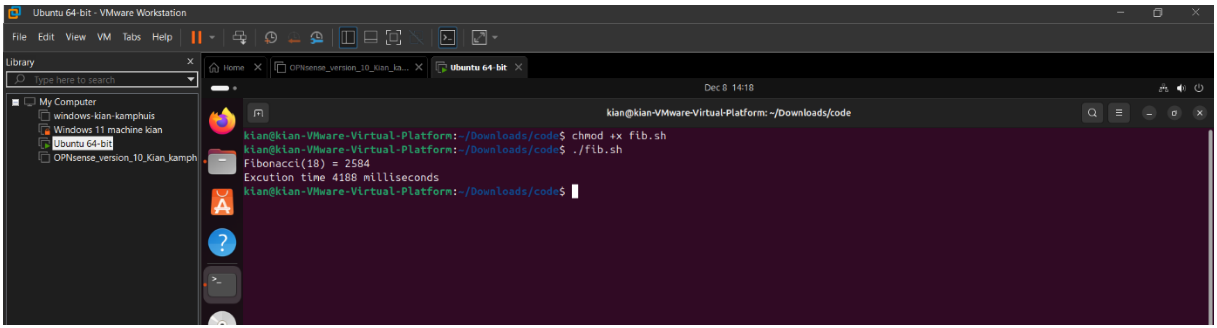
How do I run a C program?
gcc fib.c -o fib
./fib

How do I run a Bash script?
chmod +x fib.sh
./fib.sh

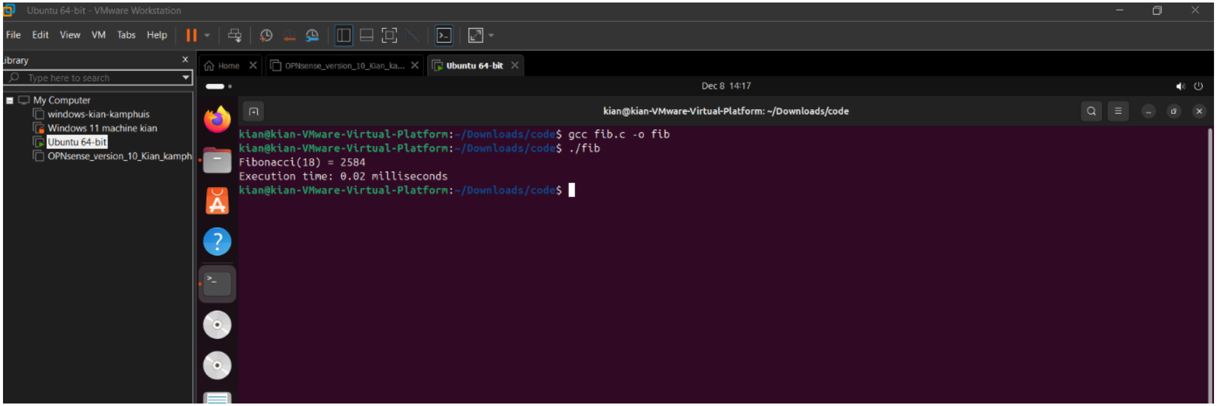
If I compile the above source code, will a new file be created? If so, which file?
Java (Fibonacci.java): Compiling creates a byte code file Fibonacci.class.
C (fib.c): Compiling creates an executable file fib.

Take relevant screenshots of the following commands:

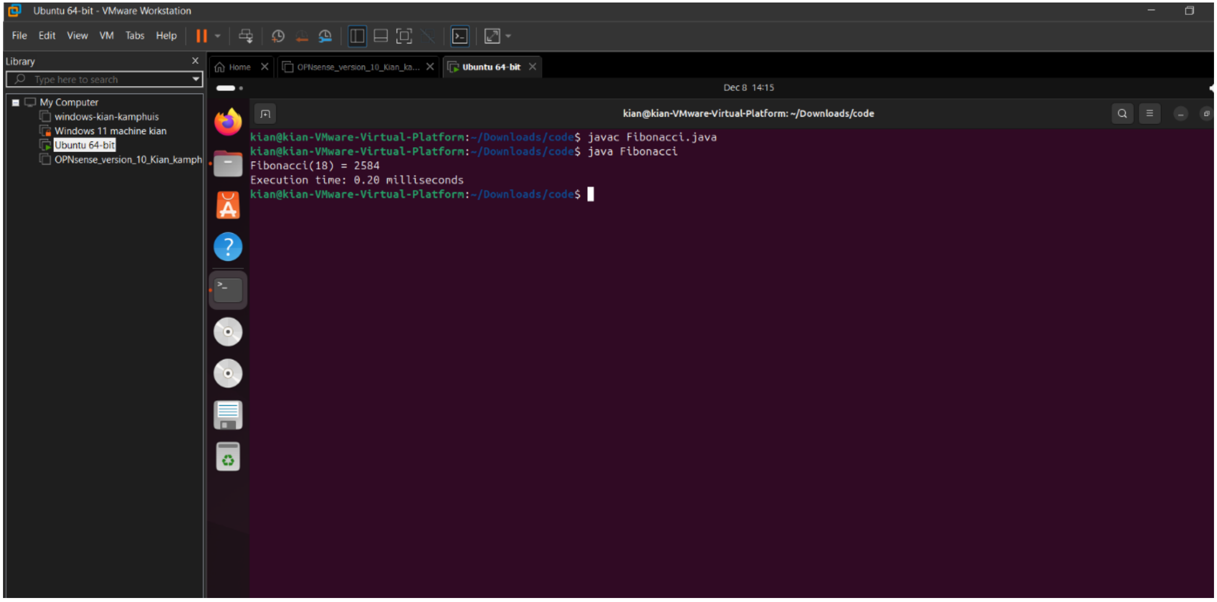
- Compile the source files where necessary
- Make them executable
- Run them
- Which (compiled) source code file performs the calculation the fastest?



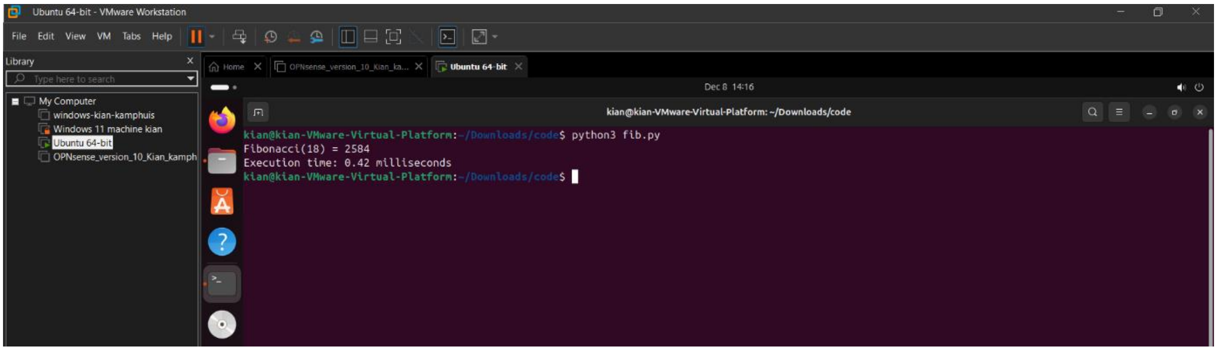
```
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ chmod +x fib.sh
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ ./fib.sh
Fibonacci(18) = 2584
Excution time 4188 milliseconds
kian@kian-VMware-Virtual-Platform:~/Downloads/code$
```



```
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ gcc fib.c -o fib
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ ./fib
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds
kian@kian-VMware-Virtual-Platform:~/Downloads/code$
```



```
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ javac Fibonacci.java
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.20 milliseconds
kian@kian-VMware-Virtual-Platform:~/Downloads/code$
```



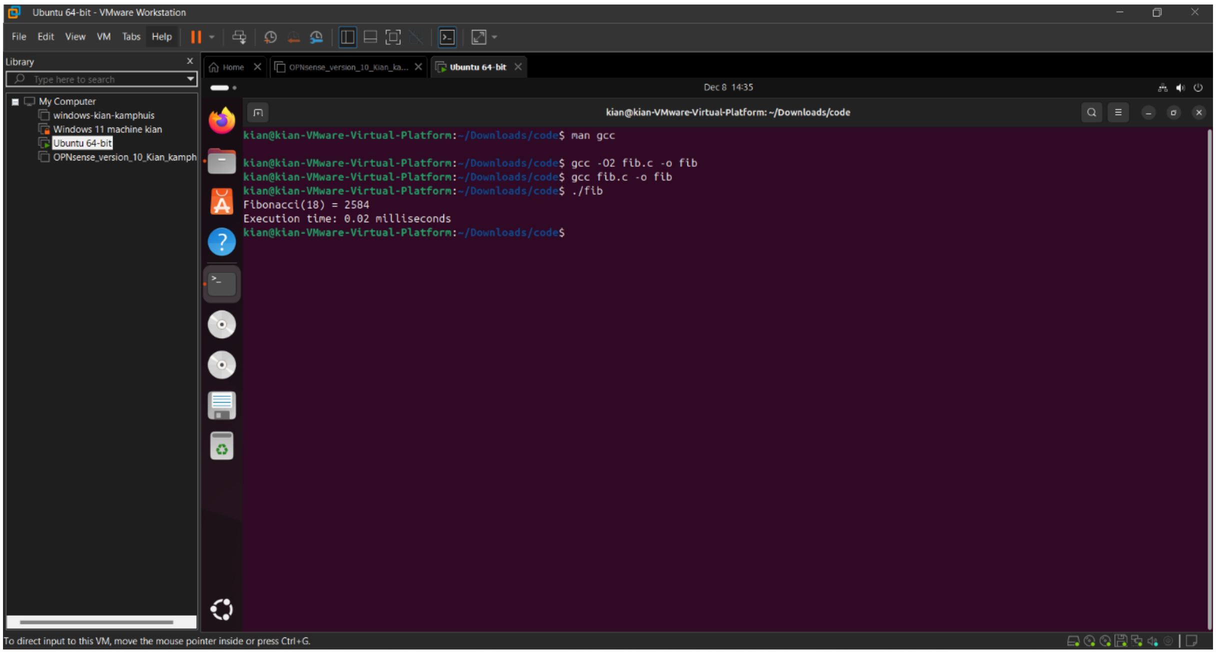
```
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.42 milliseconds
kian@kian-VMware-Virtual-Platform:~/Downloads/code$
```

The C one runs 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.


Ubuntu 64-bit - VMware Workstation
File
Edit
View
VM
Tabs
Help
Library
Type here to search
My Computer
windows-kian-kamphuis
Windows 11 machine kian
Ubuntu 64-bit
OPNsense_version_10_Kian_kamph
Home
OPNsense_version_10_Kian_ka...
Ubuntu 64-bit
Dec 8 14:35
kian@kian-VMware-Virtual-Platform: ~/Downloads/code
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ man gcc

kian@kian-VMware-Virtual-Platform:~/Downloads/code$ gcc -O2 fib.c -o fib
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ gcc fib.c -o fib
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ ./fib
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds
kian@kian-VMware-Virtual-Platform:~/Downloads/code$
```

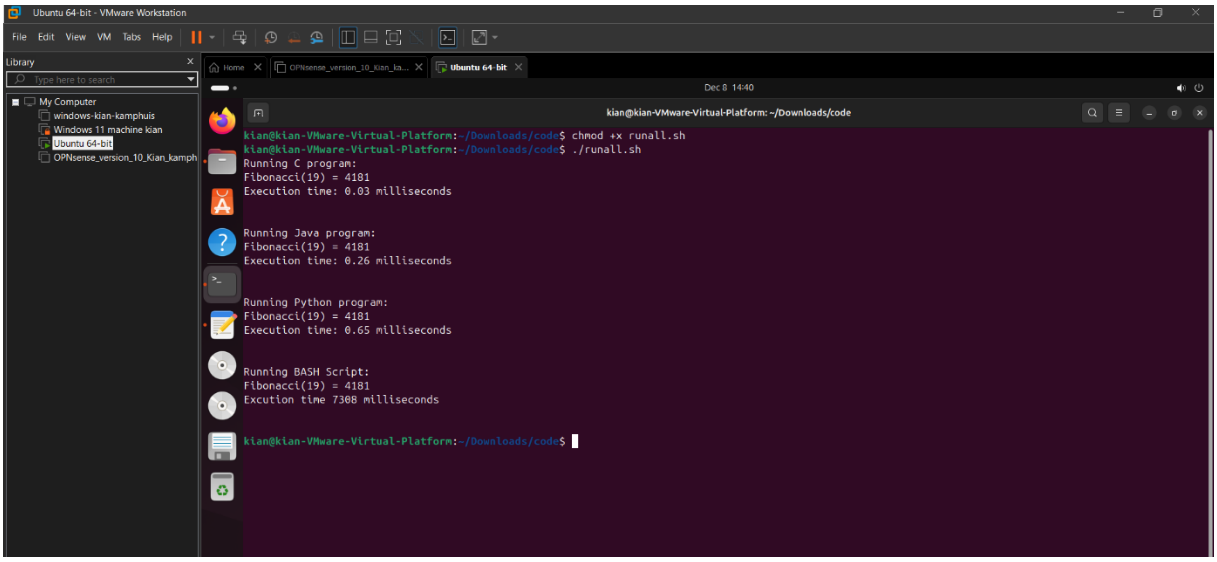
b) Compile **fib.c** again with the optimization parameters

See image 4.4 A

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

No, it runs the same.

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.



Ubuntu 64-bit - VMware Workstation
File
Edit
View
VM
Tabs
Help
Library
Type here to search
My Computer
windows-kian-kamphuis
Windows 11 machine kian
Ubuntu 64-bit
OPNsense_version_10_Kian_kamph
Home
OPNsense_version_10_Kian_ka...
Ubuntu 64-bit
Dec 8 14:40
kian@kian-VMware-Virtual-Platform: ~/Downloads/code

```
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ chmod +x runall.sh
kian@kian-VMware-Virtual-Platform:~/Downloads/code$ ./runall.sh
Running C program:
Fibonacci(19) = 4181
Execution time: 0.03 milliseconds


Running Java program:
Fibonacci(19) = 4181
Execution time: 0.26 milliseconds


Running Python program:
Fibonacci(19) = 4181
Execution time: 0.65 milliseconds


Running BASH Script:
Fibonacci(19) = 4181
Excution time 7308 milliseconds


kian@kian-VMware-Virtual-Platform:~/Downloads/code$ 
```

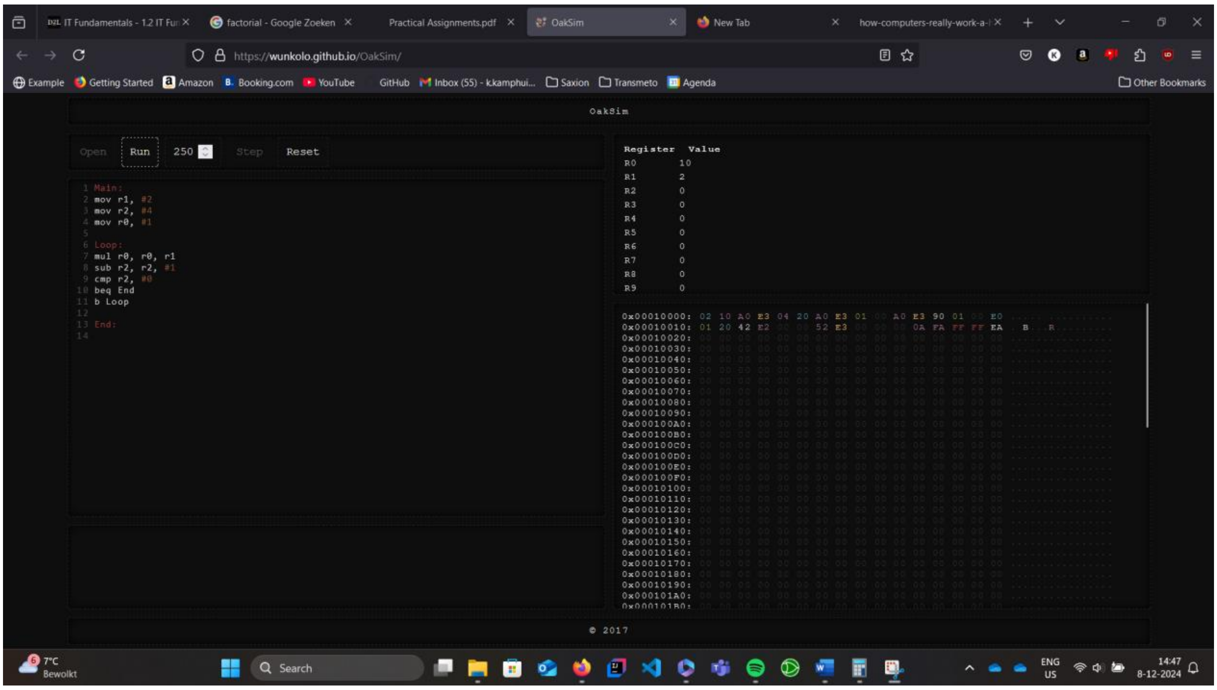
Bonus point assignment – week 4

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

mov r0, #1


Loop:

mul r0, r0, r1

sub r2, r2, #1

cmp r2, #0

beq End

b Loop


End:
```

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

Screenshot of the completed code here.



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

Loop:
mul r0, r0, r1
sub r2, r2, #1
cmp r2, #0
beq End
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 90 01 00 E0 ................
0x00010010: 01 20 42 E2 00 00 52 E3 00 00 00 0A FA 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 ................
```

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