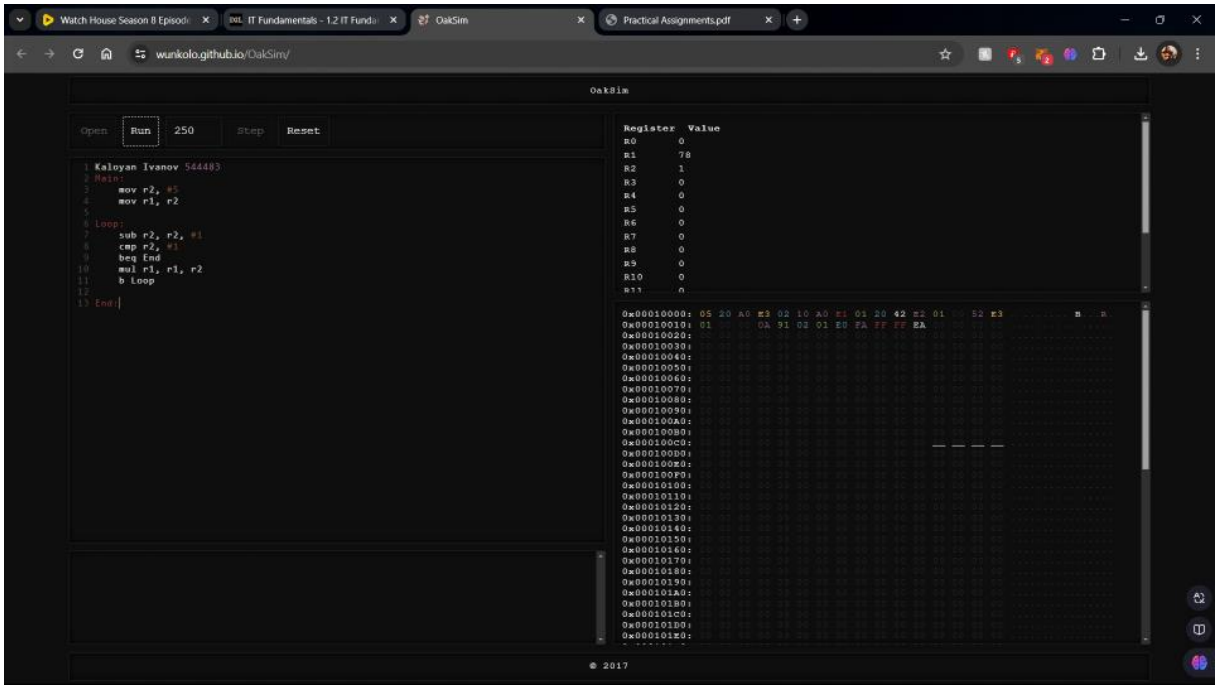
Template Week 4 – Software

Student number: 544483

Assignment 4.1: ARM assembly

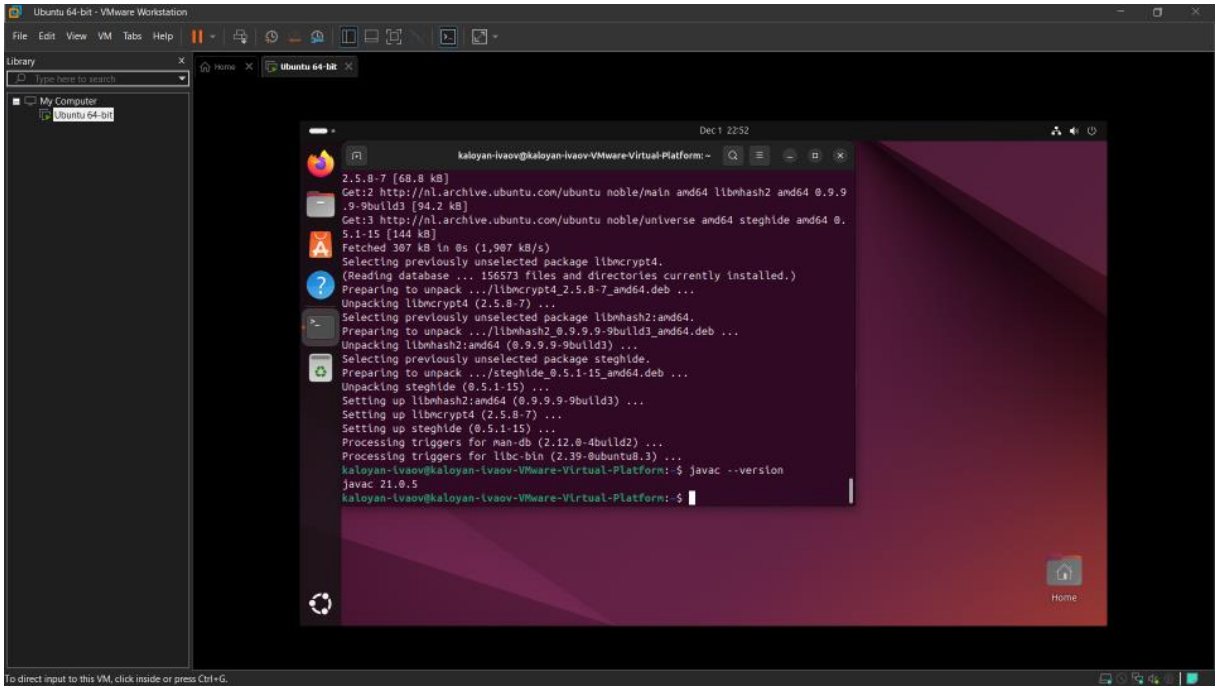
Screenshot of working assembly code of factorial calculation:



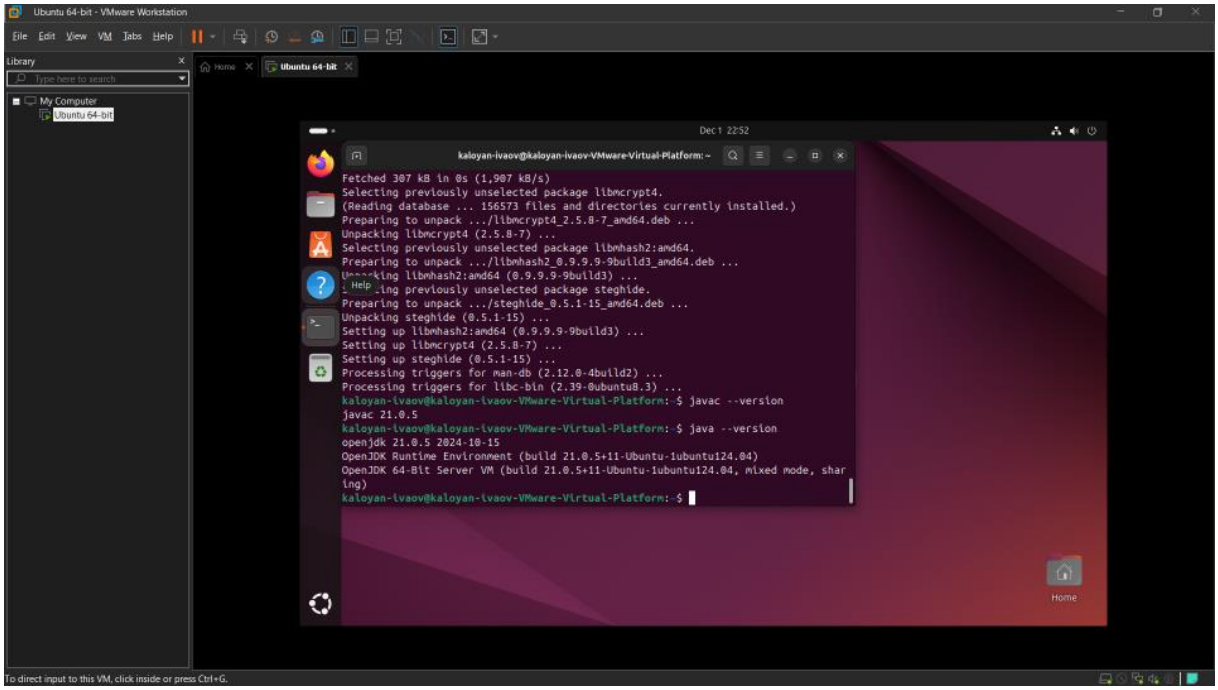
Assignment 4.2: Programming languages

Take screenshots that the following commands work:

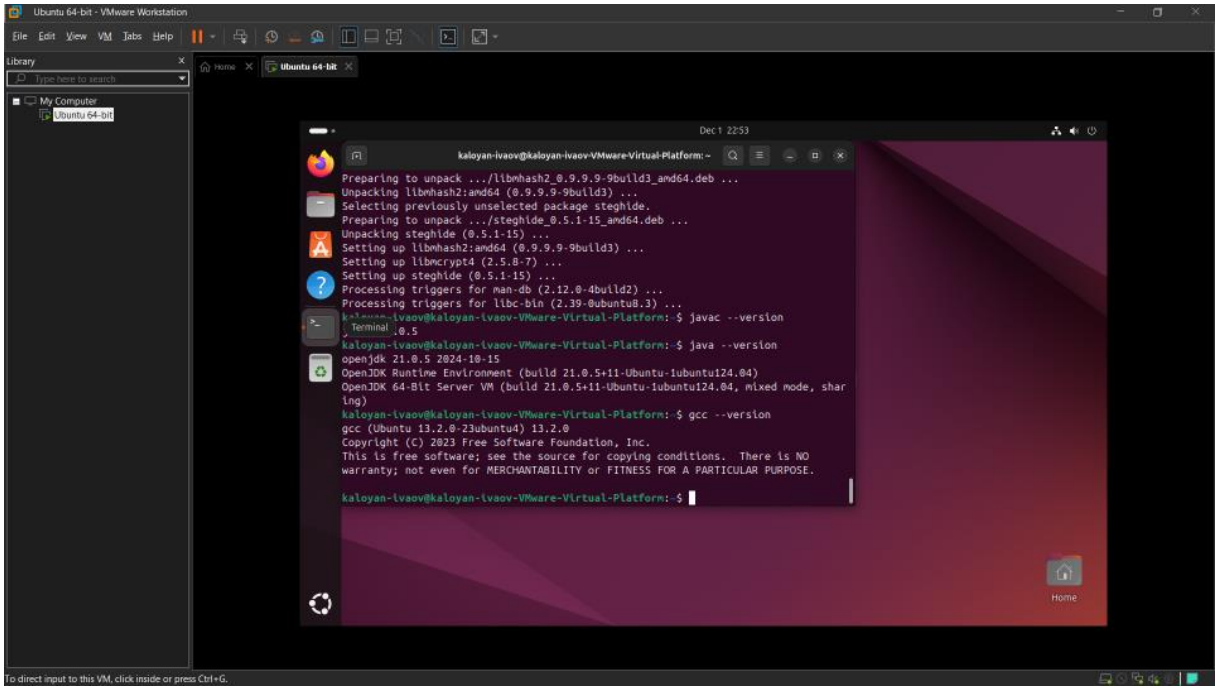
javac –version



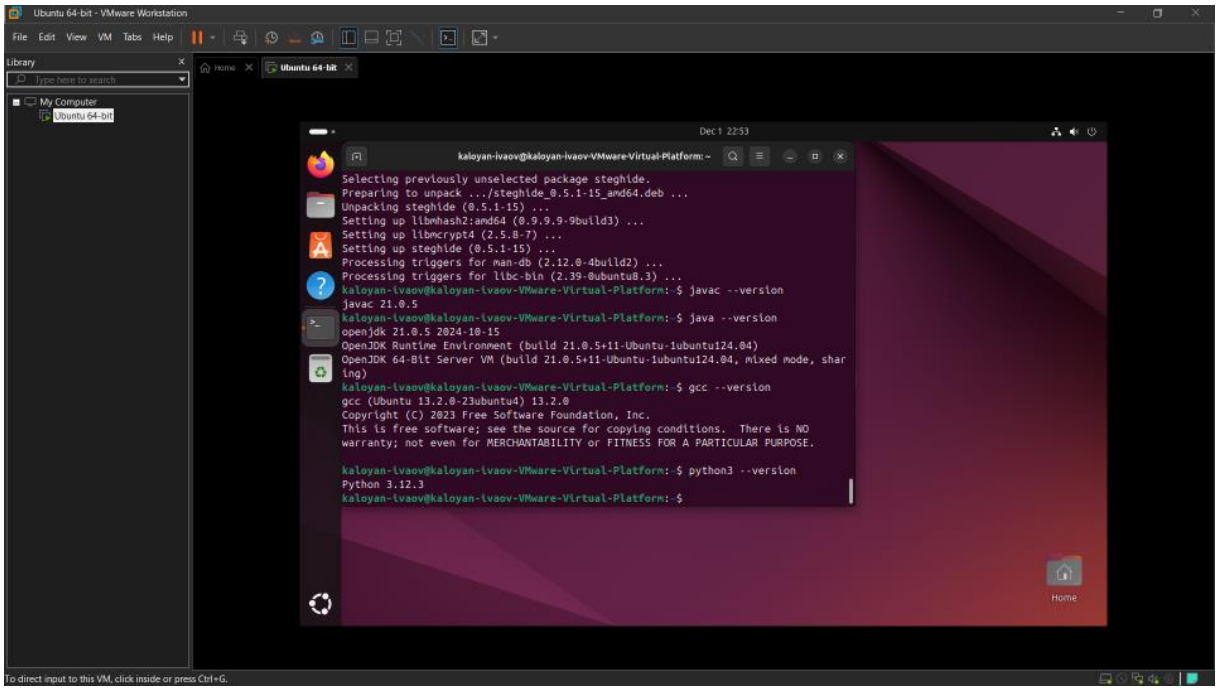
java –version



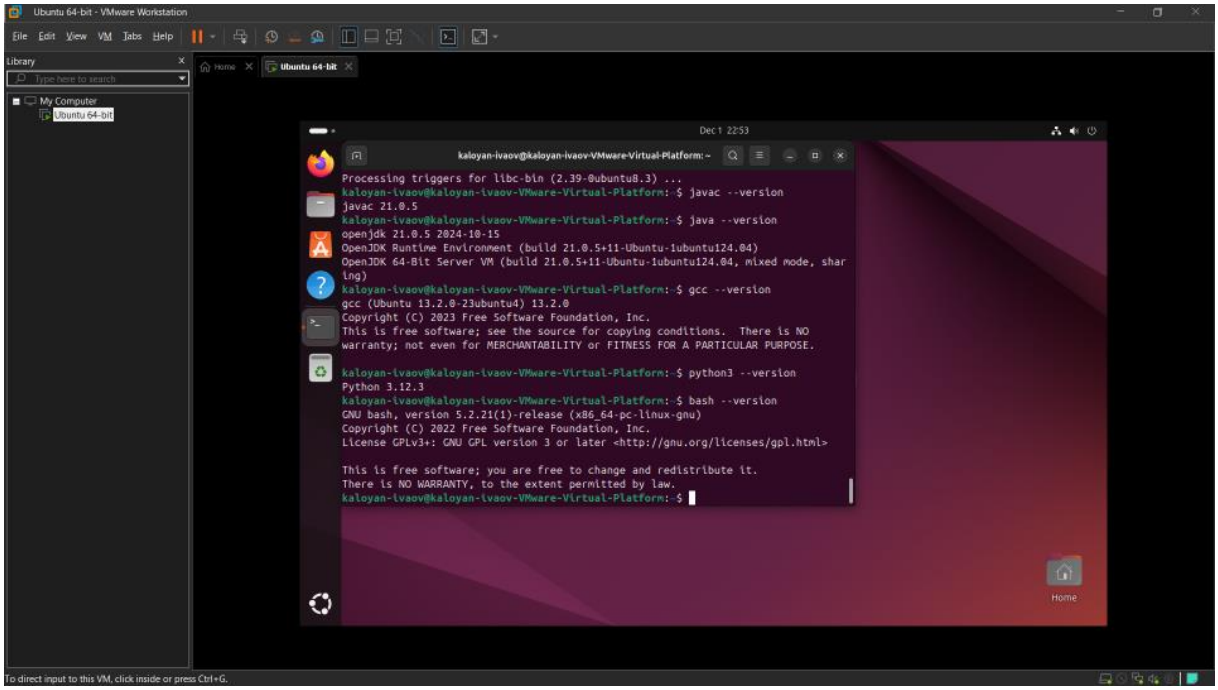
gcc –version



python3 –version



bash --version


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit
Home
Ubuntu 64-bit
Dec 1 22:53
kaloyan-ivaov@kaloyan-ivaov-VMware-Virtual-Platform: ~
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
kaloyan-ivaov@kaloyan-ivaov-VMware-Virtual-Platform:~$ javac --version
javac 21.0.5
kaloyan-ivaov@kaloyan-ivaov-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, shar
ing)
kaloyan-ivaov@kaloyan-ivaov-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.
kaloyan-ivaov@kaloyan-ivaov-VMware-Virtual-Platform:~$ python3 --version
Python 3.12.3
kaloyan-ivaov@kaloyan-ivaov-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.
kaloyan-ivaov@kaloyan-ivaov-VMware-Virtual-Platform:~$
Home
To direct input to this VM, click inside or press Ctrl+G.

Assignment 4.3: Compile

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

Fibonacci.java and fib.c

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

Fibonacci.java and fib.c

Which source code files are compiled to byte code?

Fibonacci.java

Which source code files are interpreted by an interpreter?

fib.py and fib.c

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?

Install JDK, create a new directory or use an existing one, use the "touch" command to make a new Java file and edit that file with "nano".

How do I run a Python program?

Install Python, create a new directory or use an existing one, use the "touch" command to make a new Phyton file and edit that file with "nano".

How do I run a C program?

Have a GCC installed create a new directory or use an existing one, "touch" the GCC file in the folder, this will make a C file, then edit the newly created C file with "nano".

How do I run a Bash script?

Create a bash script by making it a .sh file, make the file executable by using "sudo chmod a+x fib.sh ", the use nano to edit the file.

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

Fibbonacci.java: yes, a new file will be created, it'll be a java file.

Fib.c: yes, a new file will be created, it'll be a C file.

Fib.py: no a new file will not be created.

Fib.sh: no a new file will not be created.

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.

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?

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.

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.

Kaloyan Ivanov 544483

Main:

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

Loop:

```
cmp r2, #1
mul r0, r0, r1
sub r2,r2,#1
beq End
b 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**