Operating system

Assignment 1

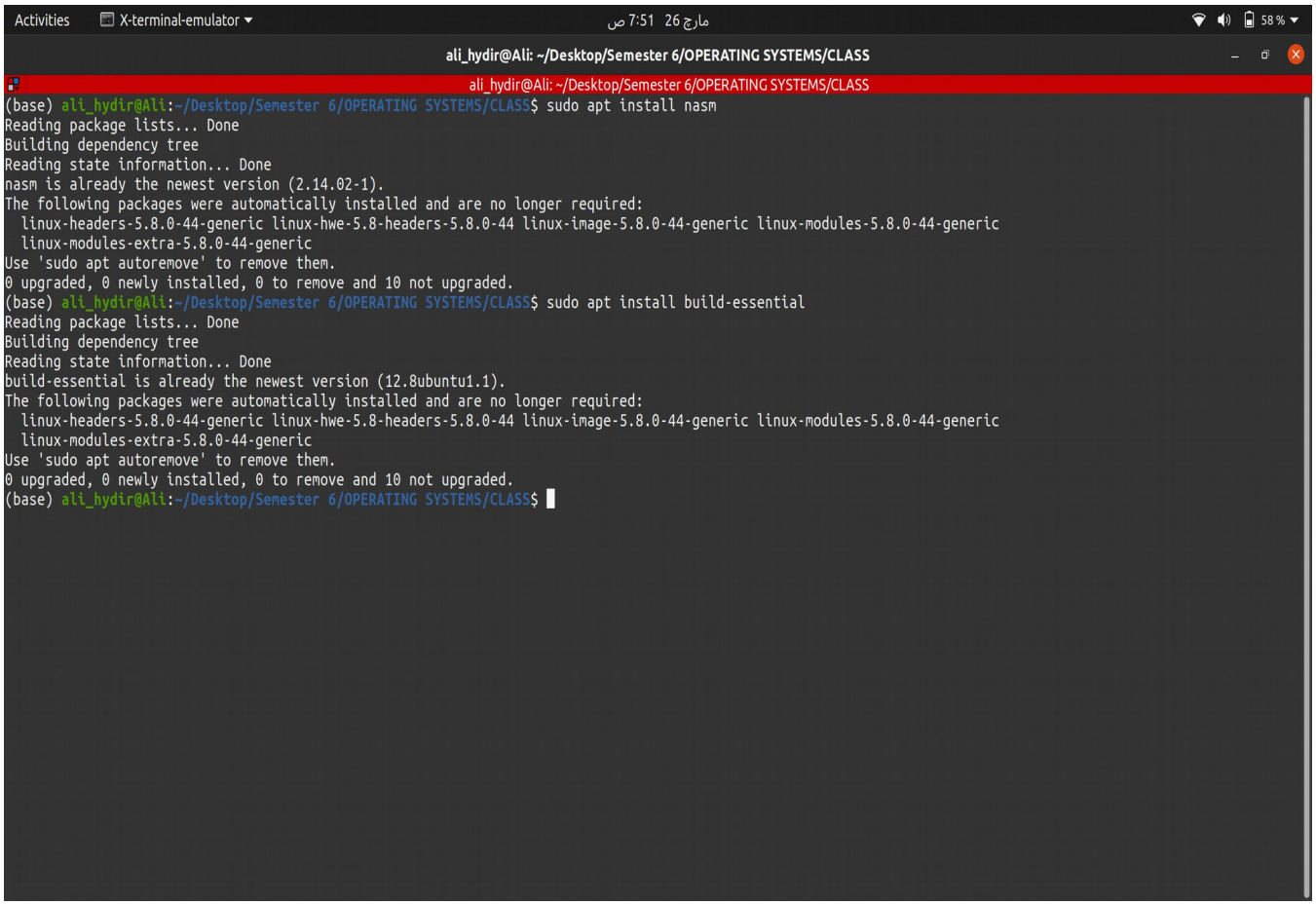
NAME: Ali Haider
ROLL_CALL:P18-0071
SEMESTER : 6

Number of experiments run:
N = **50**
Average ‘user time’ for hello (int-based calls):
I = **0.22000000000000003**

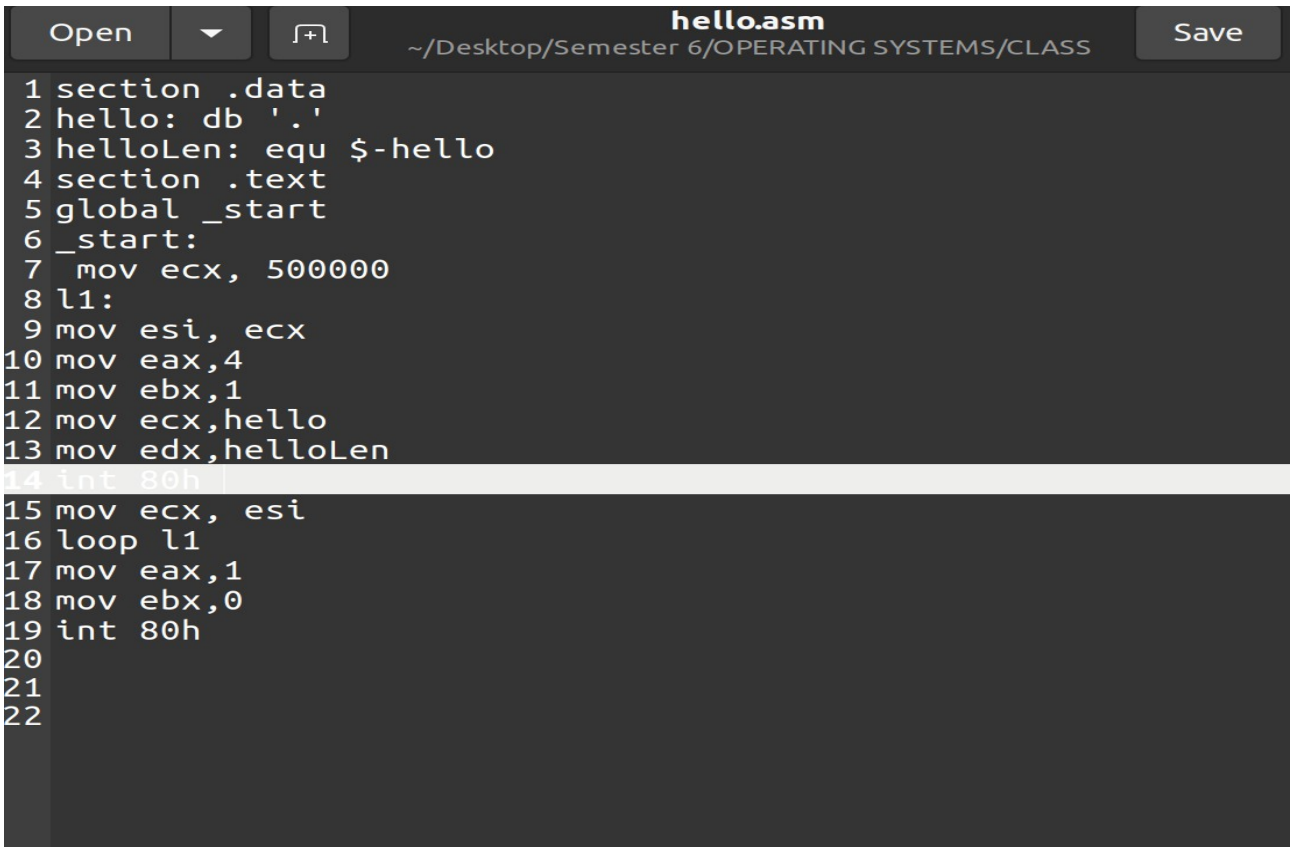
Average ‘user time’ for hello2 (syscall-based calls): S = **0.16906**

Percentage speedup: (I-S)*100/I
= **23.154545454545474%**

1) installing tools



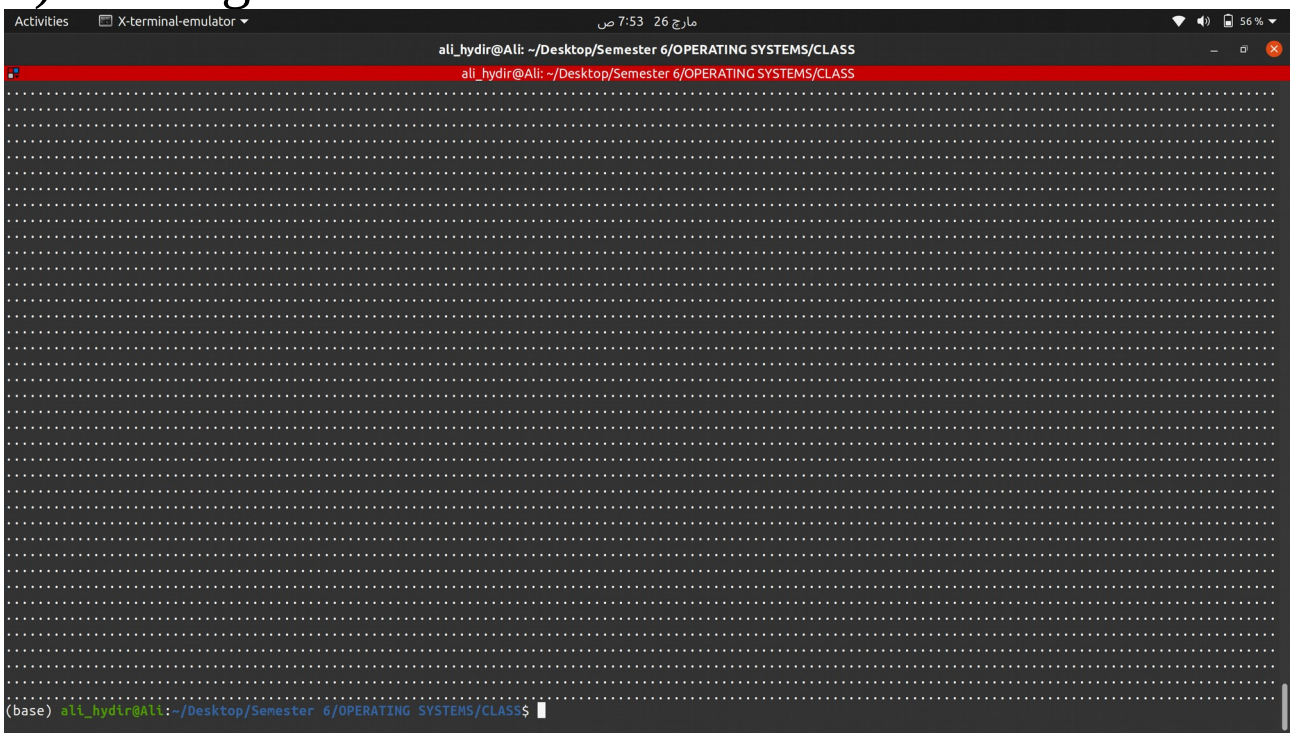
2) copying program



3) compiling and runing files

```
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ nasm -f elf64 hello.asm
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ ld -s -o hello hello.o
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$
```

4) running



5)

```
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ time ./hello > /dev/null

real    0m0.365s
user    0m0.228s
sys     0m0.136s
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$
```

6) hello2 file

Open ~/Desktop/Semester 6/OPERATING SY **hello2.asm**

```
1 section .data
2  hello: db '.'
3  helloLen: equ $-hello
4 section .text
5 global _start
6 _start:
7 mov ecx, 500000
8 l1:
9 mov ebx, ecx
10 mov rdi, 1
11 mov rsi, hello
12 mov rdx, helloLen
13 mov rax, 1
14 syscall
15 mov ecx, ebx
16 loop l1
17 mov rdi, 0
18 mov rax, 60
19 syscall
20
```

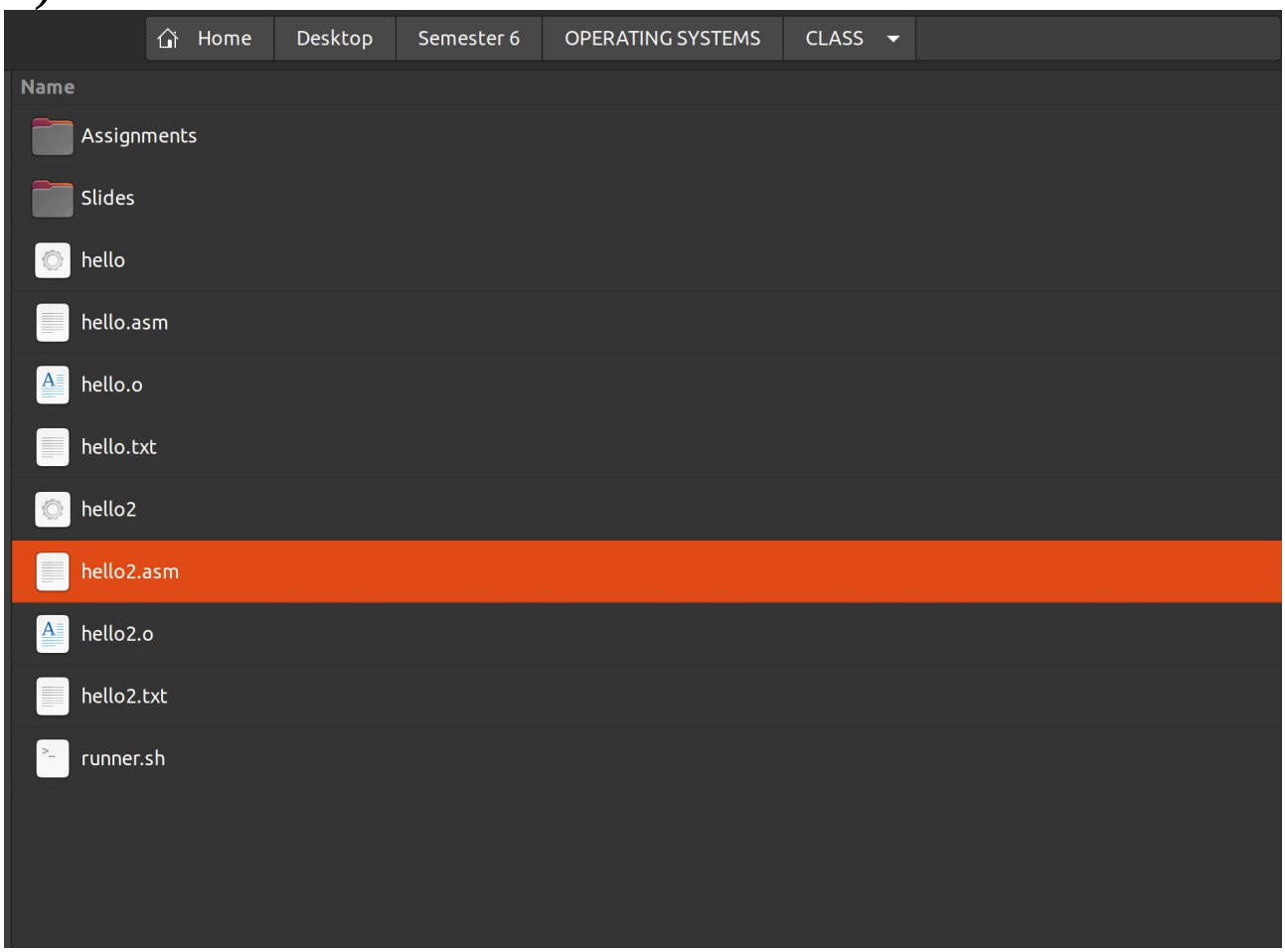
7) compiling and linking both files

```
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ nasm -f elf64 -o hello.o hello.asm
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ nasm -f elf64 -o hello2.o hello2.asm
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ ld -s -o hello hello.o
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ ld -s -o hello2 hello2.o
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$
```

8) runner.sh file

```
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ chmod +x runner.sh
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$ ./runner.sh
(base) ali_hydir@Ali:~/Desktop/Semester 6/OPERATING SYSTEMS/CLASS$
```

9)saved files



10) calculating the average time

```
def file(d1):
    l1 = []
    summ = 0
    with open(d1, "r") as f:
        for line in f.read().split("\n")[::2]:
            y =line.split("0m")[-1]
            y = y[:-1]
            l1.append(y)
        for i in range(1 , len(l1) , 2):
            f = float(l1[i])
            summ = summ + f
            average = summ/50
        #print(summ)
        return average
```

```
x = file("/home/ali_hydir/Desktop/Semester 6/OPERATING SYSTEMS/CLASS/hello.txt")
print(x)
y = file("/home/ali_hydir/Desktop/Semester 6/OPERATING SYSTEMS/CLASS/hello2.txt")
print(y)
```

```
0.22000000000000003
0.16906
```

11) comparing both files

```
Percentage speedup: (I-S)*100/I
```

```
def compare_files(a1 , a2):
    x = file(a1)
    y = file(a2)
    print("file1 = " ,x)
    print("file2 = ",y)
    if y<x:
        print("file2 takes less amount of time  ") # which in our case is system call instruction
        percentage = (x-y)*100/x
        print("percentage = " , percentage)
        return y
    else:
        print("file1 takes less amount of time ")
        return x
```

```
compare_files("/home/ali_hydir/Desktop/Semester 6/OPERATING SYSTEMS/CLASS/hello.txt" , "/home/ali_hydir/Desktop/S
```

```
file1 =  0.22000000000000003
file2 =  0.16906
file2 takes less amount of time
percentage =  23.15454545454547
```

```
7]: 0.16906
```