

Average 'user time' for hello (int-based calls): I = 0.15576

Average 'user time' for hello2 (syscall-based calls): S =0.11647999999999997

Percentage speedup:  $(I-S)*100/I = 25.21828454031846$

Output of the code :

[illegible]

Executing command :

A screenshot of a Linux terminal window titled "iffishells@iffishells: ~/Desktop/Assignment\_1". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". It contains several blank dotted lines representing scrolling history. At the bottom, three commands are entered:  
`. . . . . iffishells@iffishells:~/Desktop/Assignment_1$ nasm -f elf64 hello2.asm`  
`iffishells@iffishells:~/Desktop/Assignment_1$ ld -s -o hello2 hello2.o`  
`iffishells@iffishells:~/Desktop/Assignment_1$ ./hello2`

### code to Extract data from file

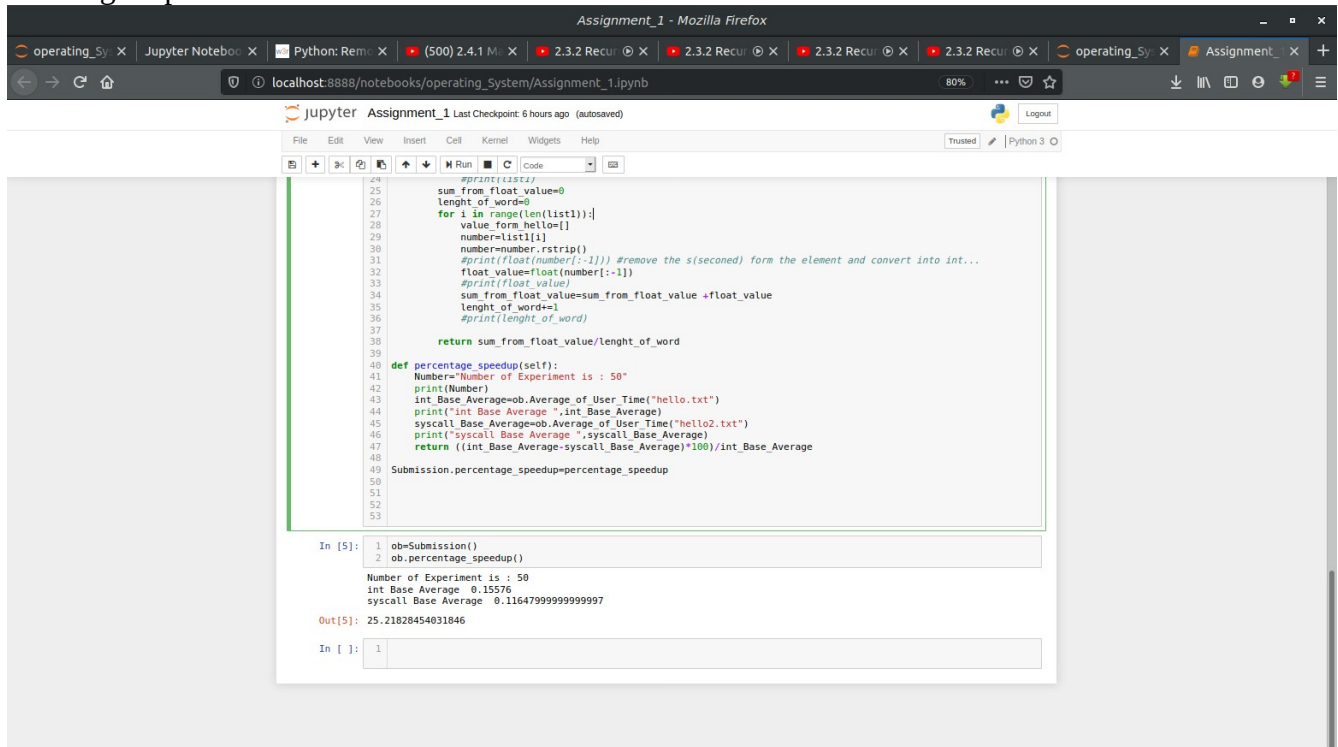
The screenshot displays a Jupyter Notebook environment within a Mozilla Firefox browser. The browser's address bar indicates the notebook is located at `localhost:8888/notebooks/operating_System/Assignment_1.ipynb`. The Jupyter interface shows the notebook title as "Assignment\_1" with a note that the last checkpoint was 6 hours ago and that there are unsaved changes. A "Logout" button is visible in the top right corner. The main workspace contains a single code cell with the following Python code:

```

In [4]: 1 class Submission:
2
3     def Average_of_User_Time(self, file_Name):
4         path = "home/jiffishells/Desktop/Assignment_1/"
5         path to file = path + file_Name
6         #print(path to file)
7         with open(path to file, "r") as file:
8             data = file.readlines()
9             count = 0
10            count_to_append_in_list = 0
11            user_time_list = []
12            list1 = []
13            for element in data:
14                if element[0] == "u":
15                    user_time_list.append(element)
16                    val = user_time_list[count_to_append_in_list] #putting element with ./ and "s"
17                    count_to_append_in_list += 1
18                    #print(val[5:])
19                    list1.append(val[8:]) #putting element with remove "/n" and "s"
20                    count += 1
21                if count == 200:
22                    break
23            #print(list1)
24            sum_from_float_value = 0
25            length_of_word = 0
26            for i in range(len(list1)):
27                value = float(list1[i])
28                number = list1[i]
29                number = number.rstrip()
30                #print(float(number[:1])) #remove the s(second) from the element and convert into int...
31                float_value = float(number[:1])
32                #print(float_value)
33                sum_from_float_value = sum_from_float_value + float_value
34                length_of_word += 1
35                #print(length_of_word)
36
37            return sum_from_float_value / length_of_word
38
39
40 def percentage_speedup(self):
41     Number = "Number of Experiment is : 50"
42     print(Number)
43     int_Base_Average = ob.Average_of_User_Time("hello.txt")
44     print("int Base Average ", int_Base_Average)
45     syscall_Base_Average = ob.Average_of_User_Time("hello2.txt")
46     print("syscall Base Average ", syscall_Base_Average)
47     return (syscall_Base_Average - int_Base_Average) / int_Base_Average

```

## showing output



The screenshot shows a Jupyter Notebook titled "Assignment\_1" in a Mozilla Firefox browser. The notebook contains a Python script with several functions and a class. The output of the script is displayed below the code cells.

```
24 #print(lists)
25 sum_from_float_value=0
26 lenght_of_word=0
27 for i in range(len(list1)):
28     value_form_hello=[]
29     number=list1[i]
30     number=number.rstrip()
31     #print(float(number[:-1])) #remove the s(seconed) form the element and convert into int...
32     float_value=float(number[:-1])
33     #print(float_value)
34     sum_from_float_value=sum_from_float_value +float_value
35     lenght_of_word+=1
36     #print(lenght_of_word)
37
38     return sum_from_float_value/lenght_of_word
39
40 def percentage_speedup(self):
41     Number="Number of Experiment is : 50"
42     print(Number)
43     int_Base_Average=obj.Average_of_User_Time("hello.txt")
44     print("int_Base_Average :",int_Base_Average)
45     syscall_Base_Average=obj.Average_of_User_Time("hello2.txt")
46     print("syscall_Base_Average :",syscall_Base_Average)
47     return ((int_Base_Average-syscall_Base_Average)*100)/int_Base_Average
48
49 Submission.percentage_speedup=percentage_speedup
50
51
52
53
```

In [5]:

```
1 ob=Submission()
2 ob.percentage_speedup()

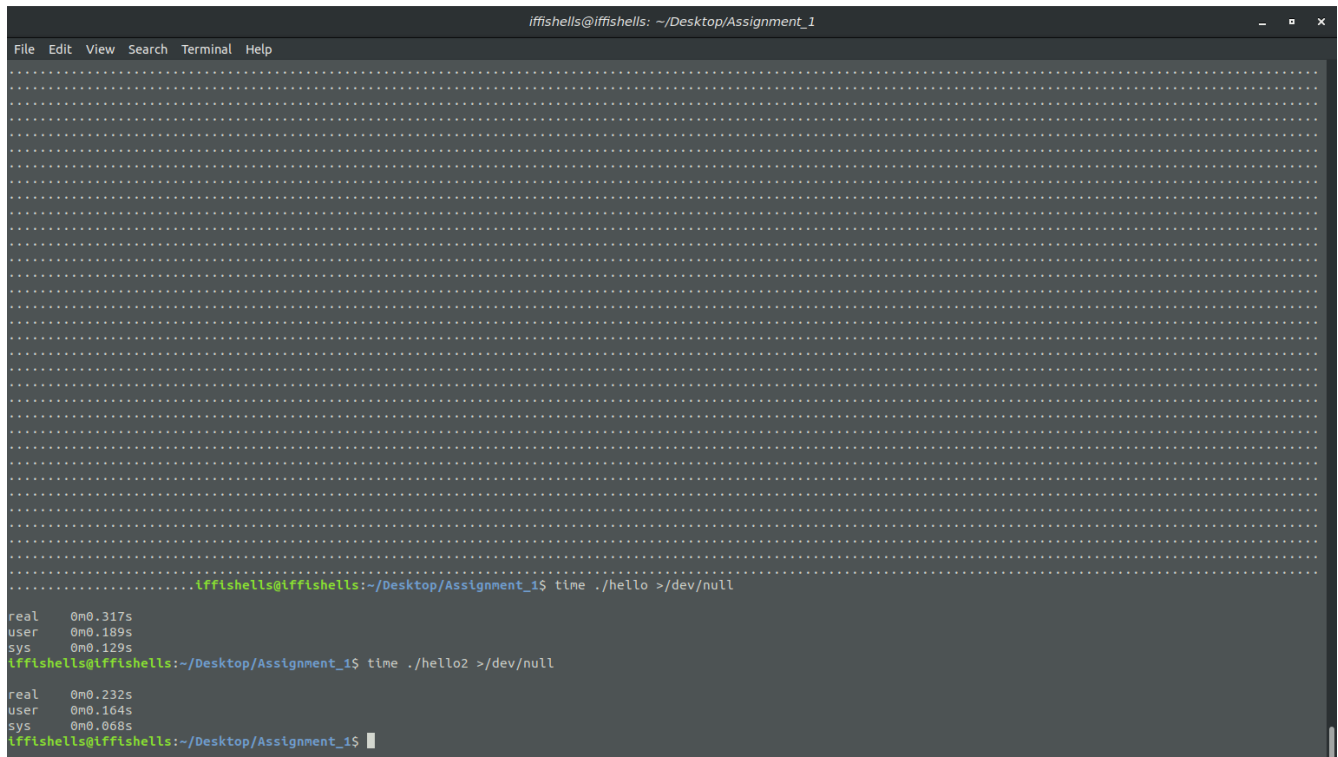
Number of Experiment is : 50
int_Base_Average 0.15576
syscall_Base_Average 0.11647999999999997

Out[5]: 25.21828454031846
```

In [ ]:

```
1
```

## show user time



The screenshot shows a terminal window with the prompt "iffishells@iffishells: ~/Desktop/Assignment\_1". The user runs two commands to measure the execution time of two programs, "hello" and "hello2".

```
iffishells@iffishells:~/Desktop/Assignment_1$ time ./hello >/dev/null

real    0m0.317s
user    0m0.189s
sys     0m0.129s

iffishells@iffishells:~/Desktop/Assignment_1$ time ./hello2 >/dev/null

real    0m0.232s
user    0m0.164s
sys     0m0.068s
```