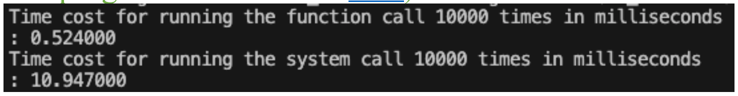
1. Are system calls more expensive than function calls? Write a simple test program to compare the "cost" of a simple function call to a simple system call.
   Function calls are more expensive than system calls. Here is an example underpining this: (The code of the program can be found here)

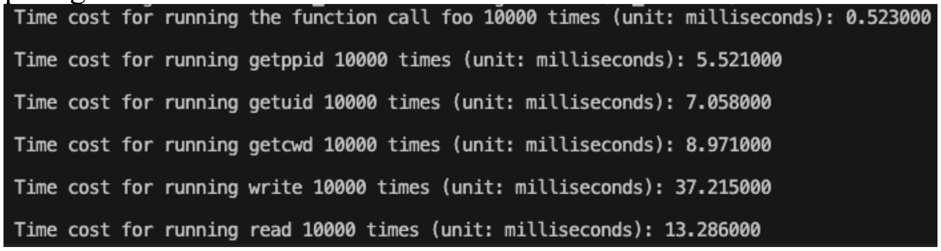


```
Time cost for running the function call 10000 times in milliseconds
: 0.524000
Time cost for running the system call 10000 times in milliseconds
: 10.947000
```

2. Explain the difference (if any) between the time required by simple function* call and simple system call by discussing what work each call must do be specific.
   System calls are more expensive than functions calls because system calls must do the context switching from the user mode to the kernel mode. After finishing any oeprations of the system level (hardware level), system calls have to switch back to the user mode to ensure the surcuirty of the system.

   Function calls, on the contrary, doesn’t require any interaction with the kernel or the OS, and therefore it won’t cost too much time.

3. Comparing cost



```
Time cost for running the function call foo 10000 times (unit: milliseconds): 0.523000

Time cost for running getppid 10000 times (unit: milliseconds): 5.521000

Time cost for running getuid 10000 times (unit: milliseconds): 7.058000

Time cost for running getcwd 10000 times (unit: milliseconds): 8.971000

Time cost for running write 10000 times (unit: milliseconds): 37.215000

Time cost for running read 10000 times (unit: milliseconds): 13.286000
```

   In short, the function call (foo) cost the least time, followed by the system calls getppid() and getuid(). The system calls write() and read() cost the most time.