

Assignment #3

CMPT 300 – Operating Systems – Fall, 2017

TASK: Calculating the time to perform various function calls.

SUBMISSION INSTRUCTIONS: Your assignments must be handed in as before on the due date via email. You will electronically submit a single .ZIP, .RAR or .TAR file that will contain:

- your source code file/files, and
- Makefile for compiling your source

Your program must be in standard C/C++.

It is your responsibility to make sure your program compiles and runs smoothly under Linux, with the kind of environment. This is the only submission method.

DEADLINE: The deadline for homework is 11:59:59 PM on Tuesday, November 27th , 2016. Submissions after this deadline will be considered late and WILL NOT be accepted.

GRADING: Your homework assignment is worth 5 points (5%), and will be marked according to the following criteria

- 1: (1 point) Measure the cost of a minimal function call in C/C++ (cost in terms of time taken). The minimal cost could be emulated by measuring a bare function call that neither takes any parameter nor does anything inside the function.
- 2: (1 point) Measure the cost of a minimal system call in C/C++. Unlike a regular function call, a system call traps into the operating system kernel. The minimal cost can be emulated by measuring the cost of getpid () which doesn't really do anything.
- 3: (2 point) Measure the cost of a read/write system call in C/C++. (Hint: You can fork a child process and transfer one byte between parent and child)

For the above measurements, you will need to repeat the experiment many times and then take the average. You will need to use a high resolution timer, see here:

http://linux.die.net/man/3/clock_gettime

The goal is to have STABLE measurement results. You are encouraged to be innovative in designing your own test.

Good programming style (1 point)

- Your code is well-organized and compartmentalized, properly indented, and easy to read and understand.
- Your code is thoroughly commented and includes a comment header with your name, student number and date.
- Functions and variables have meaningful, easy to understand names.

IMPORTANT: Your program must COMPILE AND RUN or you will get a mark of ZERO on your assignment.

Assignment grades will be open for review for one week only.

ADVANCED REMARKS:

Here is a typical student email well into this assignment:

```
I had a question regarding the timing of minimal function calls and minimal
system calls. When I run the minimal function call for just one instance and
measure it out the amount of time it takes, my tests range from around 80 to
500 nanoseconds. As for the minimal system call, my tests range from around
1000 to 6000 nanoseconds.
>
> However, when I make a while loop to run the minimal function call 1000000
times and then divide the total time taken by 1000000, I average around 20-22
nanoseconds. I get a similar result with the minimal system call. Is this
normal?
```

The answer is as follows:

```
This is because of cpu caching your last memory access. To get proper results
you have to device a way to avoid it.
```

NOTE: *This is much simpler than the previous two assignments.* However, you have the option to do an in-class presentation on an advanced topic (details have been already given in class). You MUST email me your decision with an abstract if you chose the presentation BEFORE THE ANNOUNCED DATE IN CLASS. Otherwise, you must do this programming assignment.