**OS Lab Ex: 5 21/9/21**

**System Calls vs. Procedure Calls:**

Write a simple test program to compare the cost of a simple procedure call to a simple system call (getpid(), fork() etc.,). To prevent the optimizing compiler from “optimizing out” your procedure calls, do not compile with optimization on. You should use a system call such as the UNIX gettimeofday() for time measurements. Design your code such that the measurement overhead is negligible. Also, be aware that timer values in some systems have limited resolution (e.g., millisecond resolution). Explain the difference (if any) between the time required by your simple procedure call and simple system call by discussing what work each call must do.

To use gettimeofday:

struct timeval t1, t2;

double elapsedTime;

// start timer

gettimeofday(&t1, NULL);

// do something

// ...

// stop timer

gettimeofday(&t2, NULL);

// compute and print the elapsed time in millisec

elapsedTime = (t2.tv\_sec - t1.tv\_sec) \* 1000.0; // sec to ms

elapsedTime += (t2.tv\_usec - t1.tv\_usec) / 1000.0; // us to ms

fprintf(stderr, "elapsedTime = %5.3fms \n", elapsedTime);

Procedure call:

* Simple Addition
* Any function containing some loop execution of your own.

Compare the time taken and justify your answer.