Documentation
Assignment - 1
Operating Systems

Name : Saurav vara prasad Channuri
Roll no: ES16BTECH11007

The goal is to find the time to execute a command by running this file by using multiple processes. When we fork a process two independent parallel processes will be created. We executed the command by the input in the child process where we start out time counter. Here we would have to note the starting time and we note the ending time in parent process when the child process completely executes. But since both the processes are different, their memory locations are different. So we created a shared memory and stored the start time of the process using **gettimeofday()** from library **sys/time.**

Snippet where we created shared memory

```
int SIZE = 4096;
 /* name of the shared memory object */
  const char *name = "OS";

int shm_fd;

   struct timeval *start, end;

   shm_fd = shm_open(name, O_CREAT | O_RDWR, 0666);

   ftruncate(shm_fd, SIZE);

start = (struct timeval *)mmap(0, SIZE,PROT_READ | PROT_WRITE, MAP_SHARED, shm_fd, 0);
```

Libraries needed for shm

```
#include <unistd.h>
#include <sys/types.h>
#include <unistd.h>
```

Libraries needed for creating a child process

```
#include <sys/mman.h>
#include <sys/stat.h>
#include <sys/types.h>
```

We allocated some shared memory of size 4 bytes and pointed the shared memory with the **timeval** variable **start**

We noted the time using **gettimeofday** in child process just before execution of the input command and when the child process completes and the parent process runs, we again note the time using gettimeofday. The function **gettimeofday** gives two values as output, one as seconds and one as microseconds.

Now we noted the start and end time of the process. We take the difference of the end times and start times for the duration of execution of the process. Both the values of seconds and microseconds will be in type long integers so we have to take the difference of end and start time of microsecons and divide it by 100000.0 to convert it to seconds with a decimal point.

Now we add this to the difference in end and start times of seconds. And we get the execution time of the input command.