

ASSIGNMENT-1

PNVSSK HAVISH,EE16BTECH11023

November 16, 2018

1 Design specifics of the Program

This assignment is a basic implementation of shared memory technique of Inter Process Communication where the memory shared is between the parent and its child. To start with, a shared memory file descriptor is created using the function *shm_open()*. This memory is of read and write format. The function *ftruncate()* is used to truncate this shared memory up to 4096 bytes. Using the *mmap()* function, shared memory is created and *ptr* is used to point to that specific address. *ptr* is of the data type *struct timeval**. Now, the *fork()* function is used to create a child process. This will return a pid of 0 for the child process and a non negative integer. The *gettimeofday()* function gives a structure which holds two variables one being seconds and the other being micro seconds. The *execvp()* function executes a given command line argument. The *wait(NULL)* function makes the parent process wait until the child execution is completed. The start and finish variables hold the details of initial and final time stamps.

2 Analysis of the output

When the program is executed, shared memory is created under the name "OS" and the pointer *ptr* points to its location. After this, the *fork()* function is called which creates two processes. The child process with pid=0 is executed first. In the child process, the initial time stamp is first stored in the shared memory location and the argument given on command line is executed. After the child is terminated, control will be transferred back to parent process. The final time stamp will be stored in finish variable and the initial time stamp is read back from the shared memory. The difference between the final and initial time stamps is calculated(in seconds) and displayed on the screen. The time was around 0.002302 seconds for executing *ls* command.