CONTENTS OF THIS FILE

PURPOSE AND OVERVIEW

A multi-process C program involving IPC to find the time needed to execute an instruction from the command line. The program makes use of the fork(), execvp() and gettimeofday() functions to achieve its goals.

PROGRAM STRUCTURE
---

 * Importing required header files

 * Creating a struct to store the starting times, which will eventually be written to the shared memory

 * Declaring parameters for command line arguments (int argc, char *argv[])

 * Declaration of required variables

   - variables for the shared memory space

   - variables and using struct timeval for checking the time

 * Creating the shared memory object (shm_open()), configuring it's size (ftruncate()), and mapping it to a pointer (mmap())

 * Creating a child process (using fork()) which will run the entered command

 * Using an if-else-if ladder to execute the next set of instructions depending on the current process (using the pid to check the current process):

   - fork() fails: print error message

   - child process: use gettimeofday() to get current time, write the starting time to the shared memory,

and then execute the entered command using execvp()

- parent process: wait for the child process to finish executing, check ending time, extract starting time from the shared memory and pring the time elapsed

* Unlinking the shared memory space

* Returning 0 for main()

OUTPUT

The output of this program displays the following:

* Output of the command passed as a command line argument

* The time taken by the aforementioned command to run

* Example:
./Asgn1-ES19BTECH11017 ls

Asgn1-ES19BTECH11017 Asgn1-ES19BTECH11017.c
Elapsed time: 0.002744

AUTHOR

Soumi Chakraborty
es19btech11017@iith.ac.in