Lab 2 Report

Name: Ricky Fan
Email: fanh11@mcmaster.ca
Student ID: 400248976
Date: 23/10/2022

time-shm

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/mman.h>
#include <sys/wait.h>
#include <sys/types.h>

int main(int argc, char *argv[])
{
    const int SIZE = 4096;
    const char *name = "OS";
    int fd;
    void *ptr;
    pid_t pid;
    struct timeval current_time;

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

    ftruncate(fd, SIZE);
    ptr = mmap(0, SIZE, PROT_READ | PROT_WRITE,
       MAP_SHARED, fd, 0);

    pid = fork();
```

```
    if (pid < 0)
    {
        fprintf(stderr, "Fork failed");
        return -1;
    }
    else if (pid == 0)
    {
        gettimeofday(&current_time, NULL);
        sprintf(ptr, "%lf", current_time.tv_sec +
           current_time.tv_usec/1000000.0);
        execvp(argv[1], &argv[1]);
    }
    else
    {
        wait(NULL);
        gettimeofday(&current_time, NULL);

        fd = shm_open(name, O_RDONLY, 0666);
        ptr = mmap(0, SIZE, PROT_READ, MAP_SHARED, fd,
            0);
        if (ptr == MAP_FAILED)
        {
            fprintf(stderr, "Map failed");
            return -1;
        }

        char *rmn;
        double st_sec = strtod((char *) ptr,&rmn);
        double ct_sec = current_time.tv_sec+
           current_time.tv_usec/1000000.0;

        printf("\nElasped time: %lf seconds\n", ct_sec
            - st_sec);

        shm_unlink(name);
    }

    return 0;
}
```

time-pipe

```
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/time.h>

#define READ_END 0
#define WRITE_END 1

int main(int argc, char *argv[])
{
    int fd[2];
    pid_t pid;
    struct timeval current_time;

    struct timeval *ptr = &current_time;


    if (pipe(fd) == -1)
    {
        fprintf(stderr, "Pipe failed");
        return -1;
    }

    pid = fork();

    if (pid < 0)
    {
        fprintf(stderr, "Fork failed");
        return -1;
    }
    else if (pid == 0)
    {
        close(fd[READ_END]);
        gettimeofday(&current_time, NULL);
```

```
        write(fd[WRITE_END], ptr, sizeof(struct
            timeval));

        close(fd[WRITE_END]);

        execvp(argv[1], &argv[1]);
    }
    else
    {
        wait(NULL);

        close(fd[WRITE_END]);

        read(fd[READ_END], ptr, sizeof(struct timeval)
            );

        double st_sec, ct_sec;

        st_sec = ptr->tv_sec+ptr->tv_usec/1000000.0;

        gettimeofday(&current_time, NULL);
        ct_sec = current_time.tv_sec+current_time.
            tv_usec/1000000.0;

        printf("\nElapsed time: %lf seconds\n", ct_sec
            - st_sec);

        close(fd[READ_END]);
    }

    return 0;
}
```