Lab 3 Report

Name: Ricky Fan
Email: fanh11@mcmaster.ca
Student ID: 400248976
Date: 06/1/2022

stats.c

This C program determines the amount of time needed for a command to run on the command line using the shared memory IPC mechanism. The program can be run as

./time-shm <command>

and will output and amount of time elasped to run the input command.

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

#define NUMBER_OF_THREADS 3

typedef struct
{
    int value;
} result;

int size;
int *nums;

void *get_avg(void *res)
{
    result *r = (result *)res;
    int sum = 0;

    for (int i = 0; i < size; i++)
    {
```

```
        int num = *(nums + i);
        sum += num;
    }

    r->value = sum / size;
    pthread_exit(0);
}

void *get_min(void *res)
{
    result *r = (result *)res;
    int min = *nums;

    for (int i = 1; i < size; i++)
    {
        int num = *(nums + i);
        if (num < min)
        {
            min = num;
        }
    }

    r->value = min;
    pthread_exit(0);
}

void *get_max(void *res)
{
    result *r = (result *)res;
    int max = *nums;

    for (int i = 0; i < size; i++)
    {
        int num = *(nums + i);
        if (num > max)
        {
            max = num;
        }
    }
```

```
    r->value = max;
    pthread_exit(0);
}

int main(int argc, const char *argv[])
{
    argc--;
    argv++;

    size = argc;

    int _nums[size];
    for (int i = 0; i < size; i++)
    {
        int num = (int)strtol(argv[i], NULL, 10);
        _nums[i] = num;
    }

    nums = _nums;

    pthread_t workers[NUMBER_OF_THREADS];

    result *avg_res = (result *)malloc(sizeof(result))
       ;
    result *min_res = (result *)malloc(sizeof(result))
       ;
    result *max_res = (result *)malloc(sizeof(result))
       ;

    pthread_create(&workers[0], 0, get_avg, avg_res);
    pthread_create(&workers[1], 0, get_min, min_res);
    pthread_create(&workers[2], 0, get_max, max_res);

    for(int w = 0; w < NUMBER_OF_THREADS; w++){
        pthread_join(workers[w], NULL);
    }


    printf("The average value is %d\n", avg_res->value
       );
```

```
    printf("The minimym value is %d\n", min_res->value
        );
    printf("The maximum value is %d\n", max_res->value
        );

    return 0;
}
```

prime.c

This C program determines the amount of time needed for a command to run on the command line using the pipe IPC mechanism. The program can be run as

./time-pipe <command>

and will output and amount of time elasped to run the input command.

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

#define NUMBER_OF_THREADS 1

typedef struct
{
    int num;
} parameters;

int is_prime(int num)
{

    for (int i = 2; i < num; i++)
    {
        if (num % i == 0)
        {
            return 0;
        }
    }
```

```
    return 1;
}

void *get_primes(void *params)
{
    parameters *data = (parameters *)params;

    for (int i = 2; i <= data->num; i++)
    {
        if (is_prime(i) == 1)
        {
            printf("%d\n", i);
        }
    }

    pthread_exit(0);
}

int main(int argc, const char *argv[])
{
    pthread_t workers[NUMBER_OF_THREADS];

    int num = strtol(argv[1], 0, 10);

    parameters *data = (parameters *)malloc(sizeof(
       parameters));
    data->num = num;
    pthread_create(&workers[0], 0, get_primes, data);

    pthread_join(workers[0], 0);

    return 0;
}
```