

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CS6413 - OPERATING SYSTEM LAB

Lab exercise 10: Application using Multithreads

1. Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers

90 81 78 95 79 72 85

The program will report

The average value is 82

The minimum value is 72

The maximum value is 95

The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited. (We could obviously expand this program by creating additional threads that determine other statistical values, such as median and standard deviation.)

Note:

Compile the program

cc file.c -pthread

Example:

Main thread creates a thread to find the summation of given 'n' and another thread to find max of the given 'n'

```
#include <pthread.h>
#include <stdio.h>
int sum, max; /* this data is shared by the thread(s) */
void *worker_sum(void *p); /* threads call this function */
void *worker_max(void *p);

struct input
{
    char **data;
    int count;
}in;
int main(int argc, char *argv[])
{
    pthread_t tid,tid1,tid2; /* the thread identifier */
    pthread_attr_t attr; /* set of thread attributes */
```

```

/* get the default attributes */
pthread_attr_init(&attr);
/* create the thread */
in.data=argv;
in.count=argc;
pthread_create(&tid,&attr,worker_sum,&in);
pthread_create(&tid1,&attr,worker_max,&in);

/* wait for the thread to exit */
pthread_join(tid,NULL);
pthread_join(tid1,NULL);

printf("sum = %d\n",sum);
printf("max = %d\n",max);
}

/* The thread will begin control in this function */
void *worker_sum(void *param)
{
    struct input *t=(struct input *)param;
    int i;
    sum = 0;
    for (i = 0; i < t->count; i++)
        sum += atoi(t->data[i]);
    pthread_exit(0);
}

void *worker_max(void *param)
{
    struct input *t=(struct input *)param;
    int i;
    max = 0;
    for (i = 0; i < t->count; i++)
        if (atoi(t->data[i])>max)
            max=atoi(t->data[i]);
    pthread_exit(0);
}

```

Reference:

<https://computing.llnl.gov/tutorials/pthreads/>