Operating Systems
Fall 2018
Hw No. 3

(Due on Oct. 18)

ID / NAME : 2014310407 / 이준혁(Lee JunHyuk)

1. Textbook exercise 4.16 (p. 193). (50) (For the pdf version, p. 195, exercise 4.21)

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

#define MAX_INPUT_SIZE 100
#define To_Num(c) ((c) - ('0'))

double g_ave =0;
int g_min = 0, g_max = 0, size = 0;
```

- Use the pthread to handle the multithread
- Define the global variables which are storing the average, maximum, minimum, size

```c
void* cal_ave(void* input)
{
        int* arr = (int*)input;
        double sum = 0;

        for(int i = 0; i < size; i++)
                sum += arr[i];

        sum /= size;
        g_ave = sum;
        return NULL;
}

void* cal_min(void* input)
{
        int* arr = (int*)input;
        int min = arr[0];
        int i = 0;

        while(i < size)
        {
                if(arr[i] < min)
                        min = arr[i];
                i++;
        }

        g_min = min;
        return NULL;
}

void* cal_max(void* input)
{
        int* arr = (int*)input;
        int max = arr[0];
        int i = 0;

        while(i < size)
        {
                if(arr[i] > max)
                        max = arr[i];
                i++;
        }

        g_max = max;
        return NULL;
}
```

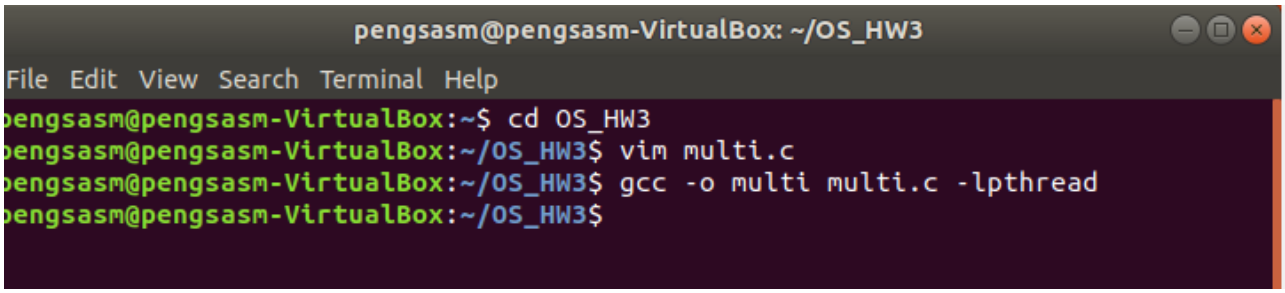
- Simply define the thread target functions for calculating min, max, ave

```c
int main()
{
        pthread_t p_ave, p_min, p_max;
        int arr[MAX_INPUT_SIZE];
        int temp = 0;
        char c;

        while((c = getchar()) != '\n' && c != EOF)
        {

                if(c >= '0' && c <= '9')
                {
                        temp *= 10;
                        temp += To_Num(c);
                }

                else if(c == ' ')
                {
                        arr[size++] = temp;
                        temp = 0;
                }

                else
                        exit(1);


                if(size == MAX_INPUT_SIZE)
                        break;

        }

        if(size != MAX_INPUT_SIZE)
                arr[size++] = temp;
```

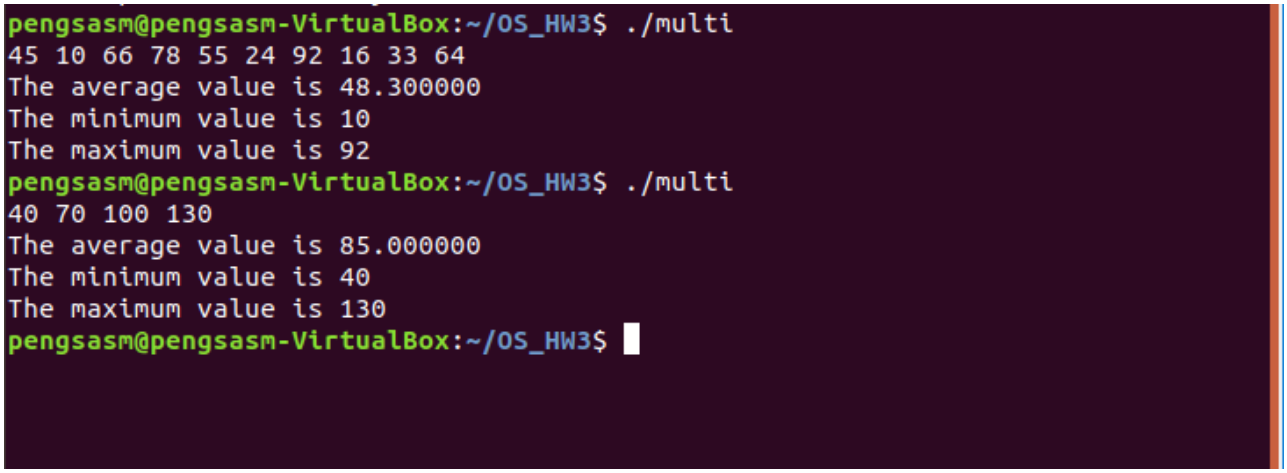
- Store the inputs at the array

```c
        pthread_create(&p_ave, NULL, cal_ave, (void *)arr);
        pthread_create(&p_min, NULL, cal_min, (void *)arr);
        pthread_create(&p_max, NULL, cal_max, (void *)arr);

        pthread_join(p_ave,NULL);
        pthread_join(p_min,NULL);
        pthread_join(p_max,NULL);

        printf("The average value is %lf\n", g_ave);
        printf("The minimum value is %d\n", g_min);
        printf("The maximum value is %d\n", g_max);
```

- By using pthread_create function, run three thread concurrently.
- Using pthread_join, wait until all these threads are terminated.



- Using lpthread option, compiles the source code



- Results