Operating Systems Design Lab
Computer Engineering Department
Fall 2023/2024
Lab 05: Threads

Objectives

1. To understand the basics of threads and multi-threaded programming.

Prelab

1. Read through chapter 4 of the textbook.
2. Read the manual pages of the following functions of the `pthread` library:

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>

int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
void *(*start_routine) (void *), void *arg);
int pthread_join(pthread_t thread, void **retval);
void pthread_exit(void *retval);
int isupper(int c);
int islower(int c);
```

Experiment

1. The following program creates a thread to convert a string to uppercase and an integer to its absolute value. Test this program and show the output to your lab instructor.

```
#include<stdio.h>
#include<pthread.h>
#include<string.h>
#include<ctype.h>

struct Data{
     int x;
     char str[20];
};

void* convert(void* param){
     struct Data* d = (struct Data*) param;
     d->x = ( d->x > 0 ) ? d->x : -1*d->x;
     int i;
     for(i=0;i<strlen(d->str);i++)
     {
          d->str[i] = toupper(d->str[i]);
     }
}

void main(){
     struct Data data;
     data.x = -7;
     strcpy(data.str, "Converting");
```

```
    pthread_t tid;
    pthread_create(&tid, NULL, convert, &data);
    pthread_join(t, NULL);
    printf("X = %d, STR = %s\n", d.x, d.str);
}
```

2. **Returning values from threads:** Modify the program in the previous part as follows. Test the program and make sure it returns the same output as the previous part.

```
void* convert(void* param){
    struct Data* output = (struct Data*) malloc(sizeof(struct Data));
    struct Data* input = (struct Data*)param;
    output->x = ( input->x > 0 ) ? input->x : -1*input->x;
    int i;
    for(i=0;i<sizeof(d->str);i++)
    {
        output->str[i] = toupper(input->str[i]);
    }
    return output;
}

void main(){
    struct Data data;
    data.x = -7;
    strcpy(data.str, "Converting");
    pthread_t tid;
    pthread_create(&tid, NULL, convert, &data);

    struct Data* result;
    pthread_join(t, &result);
    printf("X = %d, STR = %s\n", result.x, result.str);
}
```

3. Write a program that receives a line of text from the user (maximum of 50 characters). When the line is ready, the program creates two threads to process it. The first thread counts the number of uppercase characters in the string, while the second threads counts the number of lowercase characters. Each thread returns the result to the main thread via `pthread_join` (just like part 2). The main thread then prints the results from both threads to the standard output.