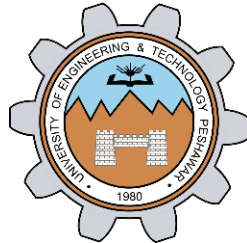


Operating Systems LAB 09
Threads: Passing Arguments to Threads



Spring 2024

Submitted by: **Hassan Zaib**

Registration No: **22pwsce2144**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

Engr. Abdullah Hamid

Month Day, Year (23 May, 2024)

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

Objective:

This lab examines aspects of threads and multiprocessing (and multithreading) and how the arguments may be passed to the child threads.

Title:

Threads: Passing Arguments to Threads

Task1:

Example: pthread_create() argument passing :

Code:

```
GNU nano 4.8                                     p1.c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

void *ChildThread(void *arg)
{
    int count = *((int *)arg);
    for (int i = 0; i < count; i++)
    {
        printf("Thread count: %d\n", i);
    }
    pthread_exit(NULL);
}

int main(void)
{
    pthread_t hThread;
    int count = 5;

    pthread_create(&hThread, NULL, ChildThread, &count);
    printf("Parent is continuing....\n");

    pthread_exit(NULL);
}
```

Output:

```
hz@hzj: ~/Desktop/Lab9
hz@hzj:~/Desktop/Lab9$ nano p1.c
hz@hzj:~/Desktop/Lab9$ gcc -pthread -o p1 p1.c
hz@hzj:~/Desktop/Lab9$ ./p1
Parent is continuing....
Thread count: 0
Thread count: 1
Thread count: 2
Thread count: 3
Thread count: 4
```

Task 02:

Problem # 1: Modify the above Box #1 program such that the main program passes the count as argument to the child thread function and the child thread function prints that many count print statements.

CODE:

```
GNU nano 4.8 p2.c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

void *ChildThread(void *arg)
{
    int count = *((int *)arg);
    for (int i = 0; i < count; i++)
    {
        printf("Thread count: %d\n", i);
    }
}

int main(void)
{
    pthread_t hThread;
    int count = 5;

    pthread_create(&hThread, NULL, ChildThread, &count);
    printf("Parent is continuing....\n");

    // No pthread_exit(NULL);
    return 0;
}
```

Output:

```
hz@hzj:~/Desktop/Lab9$ nano p2.c
hz@hzj:~/Desktop/Lab9$ gcc -pthread -o p2 p2.c
hz@hzj:~/Desktop/Lab9$ ./p2
Parent is continuing....
```

TASK 03:

Problem # 2: Write a program Box # 2 by removing pthread_exit function from parent thread function and check the output? Is it the same as output of Box # 1? If not the same, Why? Explain?

Code:

```
GNU nano 4.8 p3.c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

void *ChildThread(void *arg)
{
    int id = *((int *)arg);
    printf("Thread ID: %d\n", id);
    pthread_exit(NULL);
}

int main(void)
{
    int N = 5;
    pthread_t threads[N];
    int thread_ids[N];

    for (int i = 0; i < N; i++)
    {
        thread_ids[i] = i;
        pthread_create(&threads[i], NULL, ChildThread, &thread_ids[i]);
    }

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

```
for (int i = 0; i < N; i++)
{
    thread_ids[i] = i;
    pthread_create(&threads[i], NULL, ChildThread, &thread_ids[i]);
}

for (int i = 0; i < N; i++)
{
    pthread_join(threads[i], NULL);
}

return 0;
```

Output:

```
hz@hzj:~/Desktop/Lab9$ nano p3.c
hz@hzj:~/Desktop/Lab9$ gcc -pthread -o p3 p3.c
hz@hzj:~/Desktop/Lab9$ ./p3
Thread ID: 1
Thread ID: 2
Thread ID: 3
Thread ID: 4
Thread ID: 0
```

TASK 04:

CODE:

```
GNU nano 4.8                                p4.c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int a;
    int b;
} ThreadData;

void *ChildThread(void *arg)
{
    ThreadData *data = (ThreadData *)arg;
    printf("Thread Data: a = %d, b = %d\n", data->a, data->b);
    pthread_exit(NULL);
}

int main(void)
{
    int N = 3;
    pthread_t threads[N];
    ThreadData data[N];

    for (int i = 0; i < N; i++)
    {
        data[i].a = i;

    for (int i = 0; i < N; i++)
    {
        data[i].a = i;
        data[i].b = i * 2;
        pthread_create(&threads[i], NULL, ChildThread, &data[i]);
    }

    for (int i = 0; i < N; i++)
    {
        pthread_join(threads[i], NULL);
    }

    return 0;
}
```

OUTPUT:

```
hz@hzj:~/Desktop/Lab9$ nano p4.c
hz@hzj:~/Desktop/Lab9$ gcc -pthread -o p4 p4.c
hz@hzj:~/Desktop/Lab9$ ./p4
Thread Data: a = 2, b = 4
Thread Data: a = 0, b = 0
Thread Data: a = 1, b = 2
```


CSE 302L: Operating Systems Lab

LAB ASSESSMENT RUBRICS

Marking Criteria	Exceeds expectation (2.5)	Meets expectation (1.5)	Does not meet expectation (0)	Score
1. Correctness	Program compiles (no errors and no warnings). Program always works correctly and meets the specification(s). Completed between 81-100% of the requirements.	Program compiles (no errors and some warnings). Some details of the program specification are violated, program functions incorrectly for some inputs. Completed between 41-80% of the requirements.	Program fails to or compile with lots of warnings. Program only functions correctly in very limited cases or not at all. Completed less than 40% of the requirements.	
2. Delivery	Delivered on time, and in correct format (disk, email, hard copy etc.)	Not delivered on time, or slightly incorrect format.	Not delivered on time or not in correct format.	
3. Coding Standards	Proper indentation, whitespace, line length, wrapping, comments and references.	Missing some of whitespace, line length, wrapping, comments or references.	Poor use of whitespace, line length, wrapping, comments and references.	
4. Presentation of document	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting and excellently organized.	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting.	No name, date, or assignment title included. No task titles, no objectives, no output screenshots, poor formatting.	

Instructor:

Name: Engr. Abdullah Hamid

Signature: _____