NATIONAL TEXTILE

UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

SUBMITTED BY:

Eman Faisal 23-NTU-CS-1149

SECTION SE: 5th (A)

Operating Systems-LAB4 hometask

SUBMITTED TO:

Sir Nasir Mahmood

SUBMISSION DATE:

15/10/25

TASK1:

```c
#include <stdio.h>

#include <pthread.h>
typedef struct {
int id;
char* message;
} ThreadData;
void* printData(void* arg) {
ThreadData* data = (ThreadData*)arg;
printf("Thread %d says: %s\n", data->id, data->message);
return NULL;
}
int main() {
pthread_t t1, t2;
ThreadData data1 = {1, "Hello"};
ThreadData data2 = {2, "World"};
pthread_create(&t1, NULL, printData, &data1);
pthread_create(&t2, NULL, printData, &data2);
pthread_join(t1, NULL);
pthread_join(t2, NULL);
printf("All threads done.\n");
return 0;
}
```

OUTPUT:

```
root@DESKTOP-GFUS3VG:/home/emanuser/week4_hometask# gcc task1.c -o out
root@DESKTOP-GFUS3VG:/home/emanuser/week4_hometask# ./out
Thread 1 says: Hello
Thread 2 says: World
All threads done.
root@DESKTOP-GFUS3VG:/home/emanuser/week4_hometask#
```

TASK2:

```c
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* worker_thread(void* arg) {
int thread_num = *(int*)arg;
printf("Thread %d: Starting work...\n", thread_num);
```

```
sleep(1); // Simulate some work
printf("Thread %d: Work completed!\n", thread_num);
return NULL;
}
int main() {
pthread_t threads[5];
int thread_args[5];
// Create 5 threads
for (int i = 0; i < 5; i++) {
thread_args[i] = i + 1;
printf("Main: Creating thread %d\n", i + 1);
pthread_create(&threads[i], NULL, worker_thread, &thread_args[i]);
}
// Wait for all threads to complete
for (int i = 0; i < 5; i++) {
pthread_join(threads[i], NULL);
printf("Main: Thread %d has finished\n", i + 1);
}
printf("All threads completed!\n");
return 0;
}
```

OUTPUT:

```
root@DESKTOP-GFUS3VG:/home/emanuser/week4_hometask# gcc task2.c -o outtt
root@DESKTOP-GFUS3VG:/home/emanuser/week4_hometask# ./outtt
Main: Creating thread 1
Main: Creating thread 2
Thread 1: Starting work...
Main: Creating thread 3
Thread 2: Starting work...
Main: Creating thread 4
Thread 3: Starting work...
Main: Creating thread 5
Thread 4: Starting work...
Thread 5: Starting work...
Thread 1: Work completed!
Thread 4: Work completed!
Thread 3: Work completed!
Thread 2: Work completed!
Thread 5: Work completed!
Main: Thread 1 has finished
Main: Thread 2 has finished
Main: Thread 3 has finished
Main: Thread 4 has finished
Main: Thread 5 has finished
All threads completed!
root@DESKTOP-GFUS3VG:/home/emanuser/week4_hometask#
```

TASK3:

```c
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* calculate_sum(void* arg) {
int n = *(int*)arg;
int* result = malloc(sizeof(int)); // Allocate memory for result
*result = 0;
for (int i = 1; i <= n; i++) {
*result += i;
}
printf("Thread calculated sum of 1 to %d = %d\n", n, *result);
return (void*)result; // Return the result
}
int main() {
pthread_t thread_id;
int n = 100;
void* sum;
pthread_create(&thread_id, NULL, calculate_sum, &n);
// Get the return value from thread
pthread_join(thread_id, &sum);
```

```c
printf("Main received result: %d\n", *(int*)sum);
free(sum); // Don't forget to free allocated memory
return 0;
}
```

OUTPUT: