Course: OPERATING SYSTEMS
Course code: CT-353

LAB 03

EXERCISE:

QUESTION 01: Implement the above code and paste the screen shot of the output.

ANSWER:

CODE:

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

void *print_message_function(void *ptr);
void *func1(void *ptr);
void *func2(void *ptr);

int main() {
    pthread_t thread1, thread2;
    char *message1 = "Thread 1";
    char *message2 = "Thread 2";
    int iret1, iret2;

    /* Create independent threads each of which will execute function */
    iret1 = pthread_create(&thread1, NULL, func1, (void*) message1);
    iret2 = pthread_create(&thread2, NULL, func2, (void*) message2);

    /* Wait till threads are complete before main continues. Unless we */
    /* wait we run the risk of executing an exit which will terminate */
    /* the process and all threads before the threads have completed. */
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);

    printf("Thread 1 returns: %d\n", iret1);
    printf("Thread 2 returns: %d\n", iret2);

    exit(0);
}

void *func1(void *ptr) {
    for (int i = 0; i <= 3; i++) {
        int delay = 1;
        printf("%d\n", i);
    }
    return NULL;
}
```

Name: Hiba Kazmi Roll no: DT-22025
Course: OPERATING SYSTEMS
Course code: CT-353

```
void *func2(void *ptr) {
    for (int i = 0; i <= 3; i++) {
        int delay = 2;
        printf("%d\n", i);
    }
    return NULL;
}

void *print_message_function(void *ptr) {
    char *message;
    message = (char *) ptr;
    printf("%s\n", message);
    return NULL;
}
```

OUTPUT:

```
0
0
1
2
3
1
2
3
Thread 1 returns: 0
Thread 2 returns: 0

--------------------------------
Process exited after 0.1308 seconds with return value 0
Press any key to continue . . .
```

When committed pthread thread1 and thread 2 the output difference is:

```
Thread 1 returns: 0
Thread 2 returns: 0
0
1
0
1
2
2
3
3

--------------------------------
Process exited after 0.1373 seconds with return value 0
Press any key to continue . . .
```

Name: Hiba Kazmi Roll no: DT-22025
Course: OPERATING SYSTEMS
Course code: CT-353

```
    /* Wait till threads are comple
    /* wait we run the risk of exec
    /* the process and all threads
//    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);

    printf("Thread 1 returns: %d\n"
    printf("Thread 2 returns: %d\n"

    exit(0);
}
```

```
0
1
2
3
0
1
2
3
Thread 1 returns: 0
Thread 2 returns: 0
```

```
    /* Wait till threads are complet
    /* wait we run the risk of execu
    /* the process and all threads
    pthread_join(thread1, NULL);
//    pthread_join(thread2, NULL);

    printf("Thread 1 returns: %d\n"
    printf("Thread 2 returns: %d\n"

    exit(0);
}
```

```
0
1
2
3
0
1
2
3
Thread 1 returns: 0
Thread 2 returns: 0
```

QUESTION 02: Describe the following line of code:

iret1 = pthread_create(&thread1, NULL, print_message_function, (void*) message1);

ANSWER:

This line of code creates a new thread (thread1) that executes the print_message_function with the argument message1. The return value, indicating success (0) or error, is stored in iret1.

Name: Hiba Kazmi Roll no: DT-22025