

```

1.)#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

// printWelcomeMessage will be called when the Thread is created in the main function
// which takes string as an argument
void *printWelcomeMessage(void *names) {

    sleep(2);
    char *name = (char *)names;
    printf("\n[THREAD] Hello, Welcome %s.", name);
    pthread_exit(NULL);

}

int main () {

    // thread definition
    pthread_t threads[5];

    // parameter to be passed to the called function - printWelcomeMessage
    char names[10][15] =
{"Amritha", "Praveen", "Saurabh", "Sangeetha", "Lakshmy", "Srinivasan", "Ramaguru"};

    int result;

    for(int i = 0; i < 7; i++ ) {

        printf("\n[MAIN] Creating thread, %d", i);

        // Creating the threading and thus calling the function with parameter passed
to it
        result = pthread_create(&threads[i], NULL, printWelcomeMessage, (void
*)names[i]);

        if (result) {

            printf("Error in creating thread, %d ", result);
            exit(-1);
        }

    }

    // Exit the thread
    pthread_exit(NULL);
}

```

```

sharma@Sharma ~> nano lab.c
sharma@Sharma ~> gcc lab.c
sharma@Sharma ~> ./a.out

[MAIN] Creating thread, 0
[MAIN] Creating thread, 1
[MAIN] Creating thread, 2
[MAIN] Creating thread, 3
[MAIN] Creating thread, 4
[MAIN] Creating thread, 5
[MAIN] Creating thread, 6
[THREAD] Hello, Welcome Saurabh.
[THREAD] Hello, Welcome Srinivasan.
[THREAD] Hello, Welcome Lakshmy.
[THREAD] Hello, Welcome Amritha.
[THREAD] Hello, Welcome Ramaguru.
[THREAD] Hello, Welcome Sangeetha.
[THREAD] Hello, Welcome Praveen.
sharma@Sharma ~>

```

2.)

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

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

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

```
#include <unistd.h>
```

```
// printWelcomeMessage will be called when the Thread is created in the main function
```

```
// which takes string as an argument
```

```
void *printWelcomeMessage(void *names) {
```

```
    sleep(2);
```

```
    char *name = (char *)names;
```

```
    printf("\n[THREAD] Hello, Welcome %s.", name);
```

```
    pthread_exit(NULL);
```

```
}
```

```
int main () {
```

```
// thread defintion

pthread_t threads[5];


// parameter to be passed to the called function - printWelcomeMessage
char names[10][15] =
{"Amritha","Praveen","Saurabh","Sangeetha","Lakshmy","Srinivasan","Ramaguru"};


int result;


for(int i = 0; i < 7; i++ ) {

    printf("\n[MAIN] Creating thread, %d", i);


    // Creating the threading and thus calling the function with parameter passed to it
    result = pthread_create(&threads[i], NULL, printWelcomeMessage, (void *)names[i]);


    if (result) {

        printf("Error in creating thread, %d ", result);
        exit(-1);
    }

}


// Exit the thread
pthread_exit(NULL);
}
```

```
sharma@Sharma ~> gcc lab1.c
sharma@Sharma ~> ./a.out

[MAIN] Creating thread, 0
[MAIN] Creating thread, 1
[MAIN] Creating thread, 2
[MAIN] Creating thread, 3
[MAIN] Creating thread, 4
[MAIN] Creating thread, 5
[MAIN] Creating thread, 6
[THREAD] Hello, Welcome 140732291425976.
[THREAD] Hello, Welcome 140732291425984.
[THREAD] Hello, Welcome 140732291426000.
[THREAD] Hello, Welcome 140732291425952.
[THREAD] Hello, Welcome 140732291425960.
[THREAD] Hello, Welcome 140732291425968.
[THREAD] Hello, Welcome 140732291425992. ↵
sharma@Sharma ~> █
```

3.)

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

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

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

```
#include <unistd.h>
```

```
struct add {
```

```
    int a;
```

```
    int b;
```

```
};
```

```
void *printWelcomeMessage(void * var) {
```

```
    sleep(1);
```

```
    struct add *obj = var;
```

```
int sum = obj->a + obj->b;

printf("\n[THREAD] Hello, Sum is %d.", sum);

pthread_exit(NULL);

}

int main () {

    // thread defintion
    pthread_t threads;

    struct add var;

    var.a = 5;
    var.b = 5;

    int result;

    printf("\n[MAIN] Creating thread");

    // Creating the threading and thus calling the function with parameter passed to it
    result = pthread_create(&threads, NULL, printWelcomeMessage, &var);

    // Exit the thread
    pthread_exit(NULL);
    return 0;
}
```

```
sharma@Sharma ~> gcc lab2.c  
sharma@Sharma ~> ./a.out
```

```
[MAIN] Creating thread  
[THREAD] Hello, Sum is 10.↵  
sharma@Sharma ~>
```