**20CYS281 – Operating Systems Lab**

MultiThreading

Name : Vinayak RRoll no : CB.EN.U4CYS21084

**Sample code -Multithreading:**

/\*

@Author: Ramaguru Radhakrishnan

@Date: 21 - Dec - 2022

@Description: Creation and Execution of a simple thread

\*/

#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);

}

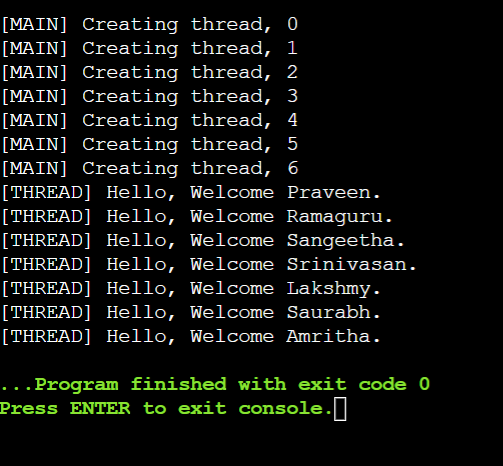
}

// Exit the thread

pthread\_exit(NULL);

}

**Output:**



**Using Thread id:**

#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 \*threads) {

sleep(2);

long \*tid=(long \*)threads;

printf("\n[THREAD] Hello, Welcome %ld.",\*tid);

pthread\_exit(NULL);

}

int main () {

// thread defintion

pthread\_t threads[10];

// 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 \*)&threads[i]);

if (result) {

printf("Error in creating thread, %d ", result);

exit(-1);

}

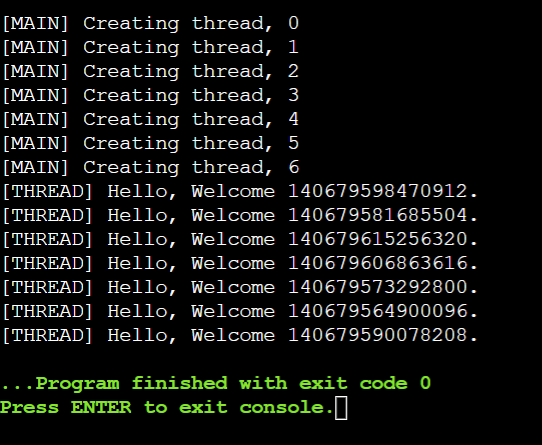
}

// Exit the thread

pthread\_exit(NULL);

}

**Output:**



**Addition with structures:**

#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 () {

pthread\_t threads;

struct add var;

var.a = 5;

var.b = 5;

int result;

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

result = pthread\_create(&threads, NULL, printWelcomeMessage, &var);

pthread\_exit(NULL);

return 0;

}

**Output:**

