**OPERATING SYSTEMS**

**20CYS281**

Name: Aishwarya GS

Roll # CB.EN.U4CYS21004

CODE#01

#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);

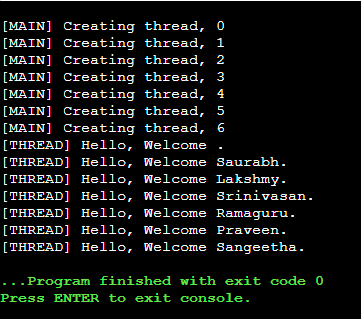
}

}

// Exit the thread

pthread\_exit(NULL);

}



CODE#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[7];

// 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);

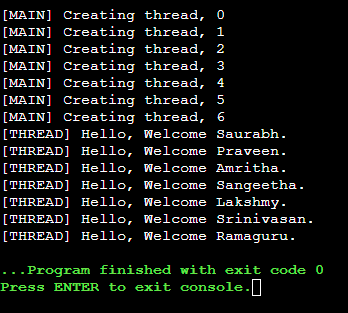
}

}

// Exit the thread

pthread\_exit(NULL);

}



Code#3

#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 \*threadid) {

sleep(2);

long tid = (long)threadid;

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

}

int main () {

// thread defintion

pthread\_t threads[7];

// 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);

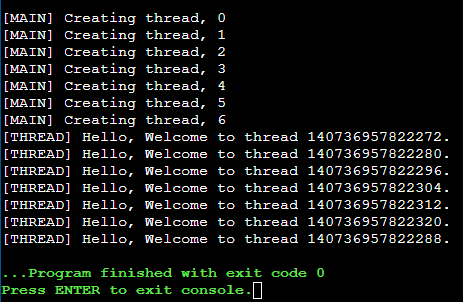
}

}

// Exit the thread

pthread\_exit(NULL);

}



CODE#04

#include <pthread.h>

#include <stdlib.h>

#include <stdio.h>

#include <unistd.h>

struct argfunc{

int a;

int b;

char name[10];

};

void \*addition(void \*arg) {

struct argfunc \*obj = arg;

int c = obj->a + obj->b;

printf("%d",c);

}

int main () {

// thread defintion

pthread\_t threads[5];

// parameter to be passed to the called function - printWelcomeMessage

int result;

struct argfunc mobj;

mobj.a = 5;

mobj.b = 6;

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, addition,&mobj);

if (result) {

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

exit(-1);

}

}

// Exit the thread

pthread\_exit(NULL);

}