Multithreading

within the same program concurrent execution happen means it is called multithreading  
if multitasking is happening in thread basis that is multi threading.

#include<pthread.h> is the header file in c for multithreading.

The lifetime of thread is only within main function(the execution of thread should complete within main function)

Pthread\_create

declare thread:  
pthread\_t variablename;

Pthread\_create(thread\_id, thread\_attributes ,function i,e this is called when thread is successful, function\_parameter)

Pthread\_exit(NULL);

#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);

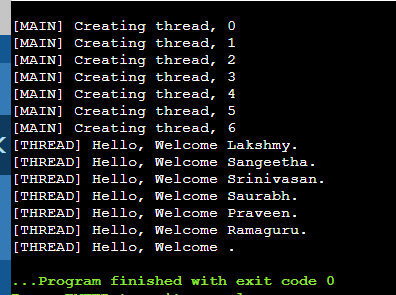
}

}

// Exit the thread

pthread\_exit(NULL);

}



To get all the names:

#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(1);

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);

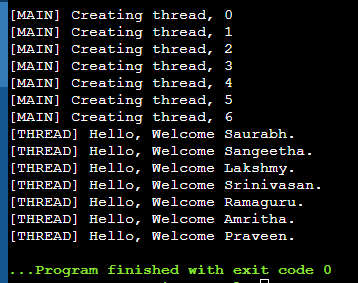
}

}

// Exit the thread

pthread\_exit(NULL);

}



TO PRINT 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 \*threadid) {

sleep(1);

long tid=(long)threadid;

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

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 \*)&threads[i]);

if (result) {

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

exit(-1);

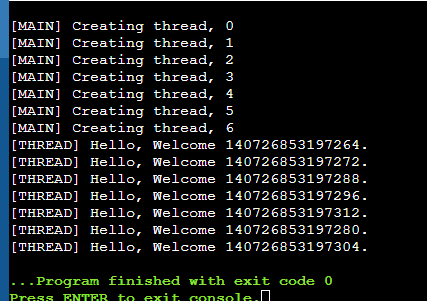
}

}

// Exit the thread

pthread\_exit(NULL);

}



If u want to send multiple parameter to function in thread then use structure.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Struct argfunc  
{  
int a;  
int b;  
};

Void \*addition(void \*arg)  
{  
struct argfunc \*obj=arg;  
int c=obj->a + obj->b;  
printf(“%d”,c);  
}

Int main()  
{  
struct argfunc mobj;  
mobj.a=15;  
mobj.b=10;

Pthread\_create(&thread,NULL,addition,&mobj);  
}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program for addition in 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

struct variabletoadd{

int a;

int b;

};

void \*addition(void \*arg) {

struct variabletoadd \*obj = arg;

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

printf("%d",c);

}

int main () {

// thread defintion

pthread\_t threads;

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

struct variabletoadd mobj;

mobj.a=15;

mobj.b=10;

int result;

// Creating the threading and thus calling the function with parameter passed to it

result = pthread\_create(&threads, NULL, addition,&mobj);

if (result) {

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

exit(-1);

}

// Exit the thread

pthread\_exit(NULL);

}

