JARIWALA SAHIL Y.

SEM: 5

SUB: OS

LAB : 4

AIM: THREAD CREATION

pthread_create():

SYNOPSIS

```
# include <pthread. h>

int pthread_create(pthread_t *thread,
      const pthread_attr_t *attr,
      void *(*start_routine) (void *), void *arg);
```

Description :

we can create thread using pthread_create(), its take thread pointer of typr pthread as first arument , as second argument its take struct pointer of type pthread_attr_t we can pass NULL as second argument which set defualt attribute for thread, as thierd argument its take function pointer which we want to run under new thread, as last argumnet its take void pointer this is for argument which is reqiured by function wich we are passing it can be int pointer, char pointer or struct pointer, array pointer. It can NULL also.

Its return 0 on sucess otherwise it return error no.

NOTE: Compile and link with -pthread.

E . G .

```
pthread_create(&t1,NULL, f1,NULL);
pthread_create(&t1,NULL, f1,(void *)str);
```

pthread_join():

SYNOPSIS

```
# include <pthread. h>

int pthread_join(pthread_t thread, void **retval);
```

Description :

we can use this method on joinalbe thread. Using this method calling thread waits for termination of thread which has passed as argument. When passed thread tterminate calling thread execute. Genrally we use this in main thread becuase main thread should wait untill all other thread terminate otherwise other thread will not execute because main process has terminated.

If multiple threads simultaneously try to join with the same thread, the results are undefined.

E . G .

```
pthread_join(t1, NULL);
```

1. Write a program to create a thread using pthread_create.

CODE:

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

void *f1(){
	printf("Hello from Thread\n");
}

int main(){
	pthread_t t1;
	int s = pthread_create(&t1,NULL, f1,NULL);
	printf("waiting for thread\n");
	pthread_join(t1, NULL);
	printf("thread jioned\n");
}
```

OUTPUT:

```
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ gcc thread.c -pthread
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
waiting for thread
Hello from Thread
thread jioned
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$
```

2.Write a program to create a multithread using pthread_create.

CODE:

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

```
void *f1(){
  printf("Hello from Thread 1\n");
}

void *f2(){
        printf("Hello from Thread 2\n");
}

int main(){
        pthread_t t1, t2;
        pthread_create(&t1,NULL, f1,NULL);
        pthread_create(&t2,NULL, f2,NULL);
        pthread_join(t1, NULL);
        pthread_join(t2, NULL);
        printf("thread jioned\n");
}
```

OUTPUT:

```
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ gcc mul_thread.c -pthread
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
Hello from Thread 1
Hello from Thread 2
thread jioned
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$
```

3. Write a program to pass a character string to the threaded function.

CODE:

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

void *f1(void *str){
        char *s = (char *)str;
        printf("String %s recived\n",s);
}
```

```
int main(){
	pthread_t t1;
	char str[10] = "Hello";

	int s = pthread_create(&t1,NULL, f1,(void *)str);
	printf("waiting for thread\n");
	pthread_join(t1, NULL);
	printf("thread jioned\n");
}
```

OUTPUT:

```
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
waiting for thread
String 'Hello' recived
thread jioned
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$
```

4. Write a program to pass struct as argument to thread and print sum of element of struct from thread

CDOE:

```
#include<pthread.h>
#include<stdio.h>
#include<string.h>
struct data
{
	int a;
	int b;
} ;
typedef struct data data;


void *f1(void *d){
```

```
        int ans;

        data *temp= (data *)d;
        ans = temp->a + temp->b;
        printf("ans is %d\n",ans);
}

int main(){
        pthread_t t1;
        data t,*d;
        t.a = 5;
        t.b = 10;
        d = &t;
        int s = pthread_create(&t1,NULL, f1,(void *)d);
        printf("waiting for thread\n");
        pthread_join(t1, NULL);
        printf("thread jioned\n");
}
```

OUTPUT:

```
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ gcc thread.c -pthread
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
waiting for thread
ans is 15
thread jioned
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$
```

5. IMPLEMENT BASIC CALCULATOR USING THREADS.

CODE:

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

```
struct calculator
{
    int a;
    int b;
} ;

typedef struct calculator calc;


void *sum(void *d){
    int ans;
    calc *temp = (calc *)d;
    ans = temp->a + temp->b;
    printf("Sum is %d\n",ans);
}

void *sub(void *d){
    int ans;
    calc *temp = (calc *)d;
    ans = temp->a - temp->b;
    printf("Sub is %d\n",ans);
}

void *mul(void *d){
    int ans;
    calc *temp = (calc *)d;
    ans = temp->a * temp->b;
    printf("Multiplication is %d\n",ans);
}

void *div(void *d){
    int ans;
    calc *temp = (calc *)d;
    ans = temp->a / temp->b;
    printf("Division is %d\n",ans);
}


int main(){
    pthread_t t1, t2, t3, t4;
    calc t,*c;
```

```
    t.a = 20;
    t.b = 10;
    c = &t;
    pthread_create(&t1,NULL, sum,(void *)c);
    pthread_create(&t2,NULL, sub,(void *)c);
    pthread_create(&t3,NULL, mul,(void *)c);
    pthread_create(&t4,NULL, div,(void *)c);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);
    pthread_join(t4, NULL);
    printf("thread jioned\n");
}
```

OUTPUT:

```
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ gcc calc.c -pthread
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
Sum is 30
Division is 2
Multiplication is 200
Sub is 10
thread jioned
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
Sub is 10
Division is 2
Multiplication is 200
Sum is 30
thread jioned
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$
```

6. WIRTE PROGRAME WHICH PERFORM 2*2 MATRIX MULTIPLICATION USING THREADS.

CODE:

```
#include<pthread.h>
#include<stdio.h>
#include<string.h>
struct matrix
```

```
{
    int a;
    int b;
    int c;
    int d;
} ;
typedef struct matrix matrix;

matrix ans;
matrix m1, m2;

void *particial_mul_a(){
    ans.a = ((m1.a * m2.a) + (m1.b * m2.c));
    printf("Ans.a is calculated\n");
}

void *particial_mul_b(){
    ans.b = ((m1.a * m2.b) + (m1.b * m2.d));
    printf("Ans.b is calculated\n");
}

void *particial_mul_c(){
    ans.c = ((m1.c * m2.a) + (m1.d * m2.c));
    printf("Ans.c is calculated\n");
}

void *particial_mul_d(){
    ans.d = ((m1.c * m2.b) + (m1.d * m2.d));
    printf("Ans.d is calculated\n");
}

int main(){
    pthread_t t1, t2, t3, t4;
    matrix temp;
    temp.a = 1;
    temp.b = 2;
    temp.c = 3;
    temp.d = 4;
    m1 = m2 = temp;
    pthread_create(&t1,NULL, particial_mul_a,NULL);
    pthread_create(&t2,NULL, particial_mul_b,NULL);
```

```
    pthread_create(&t3,NULL, particial_mul_c,NULL);
    pthread_create(&t4,NULL, particial_mul_d,NULL);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);
    pthread_join(t4, NULL);
    printf("Answer matrix is:\n");
    printf("%d\t%d\n", ans.a,ans.b);
    printf("%d\t%d\n", ans.c,ans.d);

}
```

OUTPUT:

```
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ gcc matrix.c -pthread
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$ ./a.out
Ans.a is calculated
Ans.b is calculated
Ans.d is calculated
Ans.c is calculated
Answer matrix is:
7       10
15      22
jariwalasahil@jariwalasahil:~/Desktop/SEM_5/OS/Lab$
```