Program to create Threads in Linux

Leave a Comment / Programs / By Baljit Singh Saini

In this post we discuss the use of ***pthread_create*** function create threads in linux. Various program to create threads in Linux are discussed below that shows how to create threads, how to pass input to thread and how to return value from thread.

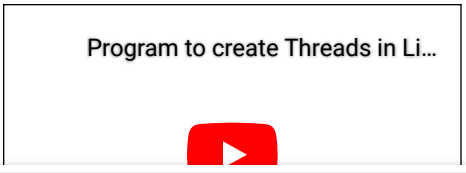
Syntax

```
#include<pthread.h>
int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
                   void *(*start_routine) (void *), void arg);
```

The first parameter is the buffer which will contain the ID of the new thread, if pthread_create is successful. The second parameter specifies the attributes of the thread. This parameter is generally NULL until you want to change the default settings. The third parameter is the name the function which the thread will execute. Hence, everything that you want the thread to do should be defined in this function. Lastly, the fourth parameter is the input to the function in the third parameter. If the function in the third parameter does not take any input then the fourth parameter is NULL.

Program 1: Program to create threads in linux. Thread prints 0-4 while the main process prints 20-24

```
 #include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
void *thread_function(void *arg);
 int i,j;
 int main() {
 pthread_t a_thread;  //thread declaration
```



```
 pthread_create(&a_thread, NULL, thread_function, NULL);
//thread is created
 pthread_join(a_thread, NULL); //process waits for thread to finish . //Comment this line to see the difference
 printf("Inside Main Program\n");
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 }



void *thread_function(void *arg) {
// the work to be done by the thread is defined in this function
 printf("Inside Thread\n");
 for(i=0;i<5;i++)
 {
 printf("%d\n",i);
 sleep(1);
 }
 }
```

Note: To compile any program which involves creation of thread(s) use pthread library (lpthread)
Suppose the above program is named "Thread.c", then to compile write
$gcc Thread.c -lpthread
To run the command remains same
$./a.out

Output

```
0
1
2
3
4
Inside Main Program
20
21
22
23
24
```

How it works?

pthread_create() creates a new thread which starts to execute thread_function. This function creates a loop which prints 0-4. The sleep function makes the thread go to sleep after each digit is printed. pthread_join() makes the main function wait until the newly created thread finishes its execution. So the control returns to the main function only when the thread finishes. Then the main function prints "Inside Main program" and executes the loop from 20-24.

How a thread returns a value to the main process?

Program 2: Program to create a thread. The thread prints numbers from zero to n, where value of n is passed from the main process to the thread. The main process also waits for the thread to finish first and then prints from 20-24.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
#include<string.h>
void *thread_function(void *arg);
 int i,n,j;
 int main() {
 char *m="5";
 pthread_t a_thread;  //thread declaration
 void *result;
 pthread_create(&a_thread, NULL, thread_function, m); //thread is created
 pthread_join(a_thread, &result);
 printf("Thread joined\n");
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 printf("thread returned %s\n",(char *)result);
 }
 void *thread_function(void *arg) {
 int sum=0;
 n=atoi(arg);



for(i=0;i<n;i++)
 {
 printf("%d\n",i);
 sleep(1);
 }
 pthread_exit("Done"); // Thread returns "Done"
 }
```

How it works?

When we call pthread_create() function a value if passed to the thread by passing that value as the fourth parameter of the pthread_create() function.

How to pass multiple values to a thread using structure?

Program 3: Program to create a thread. The thread is passed more than one input from the main process. For passing multiple inputs we need to create structure and include all the variables that are to be passed in this structure.

```
#include<stdio.h
#include<pthread.h>
struct arg_struct {   //structure which contains multiple variables that are to passed as input to the thread
     int arg1;
```

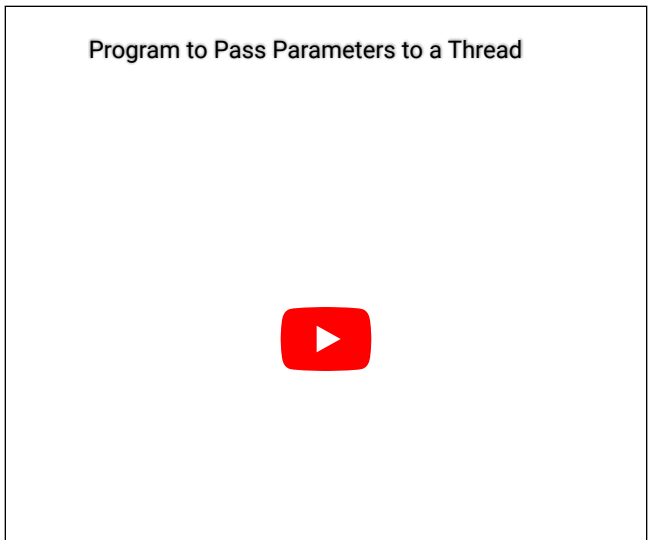
```
    printf("%d\n", args -> arg1);
    printf("%d\n", args -> arg2);
    pthread_exit(NULL);
}



int main()
 {
     pthread_t t;
     struct arg_struct args;
     args.arg1 = 5;
     args.arg2 = 7;
     pthread_create(&t, NULL, arguments, &args);
//structure passed as 4th argument
     pthread_join(t, NULL); /* Wait until thread is finished */
}
```

Note: Just as above we can also pass an array to a thread.

Video on Passing Parameters to Thread



Viva Questions Related to Program to create Threads in Linux

Q1. Why is the second parameter in pthread_create() function generally taken as NULL?

Q2. What is the significance of using pthread_join() function?

Q3. Which function is used to return some value from the child process to the parent process?

Q4. How can we pass multiple values to a thread?

Q5. How is the value returned by a thread saved/stored by the main process?

Practice Programs on Thread Creation

Q1. Write a program to create two Threads T1 and T2. Thread T1 creates a file named **Thread.txt** while T2 writes “Hello its T2” into the Thread.txt

Q2. Write a program to create a thread T1. The main process passes two numbers to T1. T1 calculates the sum of these numbers and returns the sum to the parent

Deadlock Simulation

Race Condition

Process synchronization using mutex locks

← Previous Post

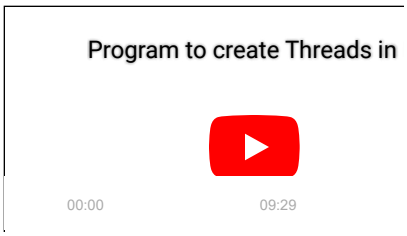
Next Post →

Leave a Comment

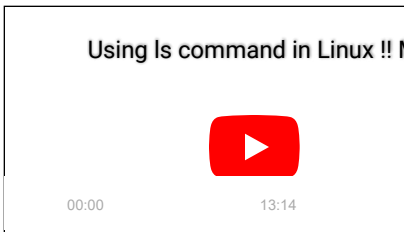
You must be logged in to post a comment.

Search ...

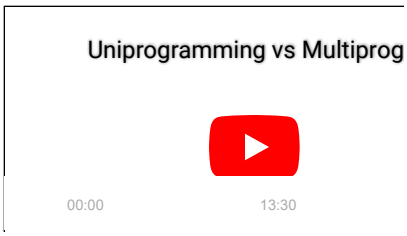
OS Lab



Linux Essentials Series



OS Theory



Categories

Database

Entity-Relationship(ER)

Introduction to Database