

All Online Learning

www.allonlinelearning.com

Thread Example

Program-1:

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

void* routine() {
    printf("Test from threads\n");
}

int main(int argc, char* argv[]) {
    pthread_t p1;
    pthread_create(&p1, NULL, &routine, NULL);
    pthread_join(p1, NULL);
    return 0;
}
```

Output:

```
lm@lm-VirtualBox:~$ gcc threadp1.c -o threadp1 -lpthread
lm@lm-VirtualBox:~$ ./threadp1
Test from threads
```

All Online Learning

www.allonlinelearning.com

Program-2:

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

void* routine() {
    printf("Test from threads\n");
}

int main(int argc, char* argv[]) {
    pthread_t p1, p2;
    pthread_create(&p1, NULL, &routine, NULL);
    pthread_create(&p2, NULL, &routine, NULL);

    pthread_join(p1, NULL);
    pthread_join(p2, NULL);

    return 0;
}
```

Output:

```
lm@lm-VirtualBox:~$ gcc threadp1p2.c -o threadp1p2 -lpthread
```

```
lm@lm-VirtualBox:~$ ./threadp1p2
```

```
Test from threads
```

```
Test from threads
```

All Online Learning

www.allonlinelearning.com

Program-3:

```
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void* routine() {
    printf("Test from threads\n");
    sleep(3);
    printf("Ending thread\n");
}
int main(int argc, char* argv[]) {
    pthread_t p1, p2;
    if (pthread_create(&p1, NULL, &routine, NULL) != 0) {
        return 1;
    }
    if (pthread_create(&p2, NULL, &routine, NULL) != 0) {
        return 2;
    }
    if (pthread_join(p1, NULL) != 0) {
        return 3;
    }
    if (pthread_join(p2, NULL) != 0) {
        return 4;
    }
    return 0;
}
```

Output:

lm@lm-VirtualBox:~\$ gcc threadcondi.c -o threadcondi -lpthread

lm@lm-VirtualBox:~\$./threadcondi

Test from threads

Test from threads

Ending thread

Ending thread

All Online Learning

www.allonlinelearning.com

Program-4:

```
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void *thread_task(void *args)
{
    printf("Hello\n We are learning Thread concepts");
    sleep(30);
}

int main()
{
    pthread_t t_id;
    int ret;
    printf("Process id=%d\n",getpid());
    printf("Before creation of thread\n");
    ret=pthread_create(&t_id,NULL,thread_task,NULL);
    if(ret==0)
    {
        printf("Thread created successfully\n");
    }
    else
    {
        printf("Problem crating thread\n");
    }
    pthread_join(t_id,NULL);
    printf("Thread Terminated \n");
    return 0;
}
```

All Online Learning

www.allonlinelearning.com

Output:

```
lm@lm-VirtualBox:~$ gcc thread1.c -o thread1 -lpthread
```

```
lm@lm-VirtualBox:~$ ./thread1
```

```
Process id=3571
```

```
Before creation of thread
```

```
Thread created successfully
```

```
Hello
```

```
We are learning Thread conceptsThread Terminated
```

Testing of Thread:

Open new terminal and then check

```
lm@lm-VirtualBox:~$ ps -T -p 3571
```

PID	SPID	TTY	TIME	CMD
-----	------	-----	------	-----

3571	3571	pts/0	00:00:00	thread1
------	------	-------	----------	---------

3571	3572	pts/0	00:00:00	thread1
------	------	-------	----------	---------

All Online Learning

www.allonlinelearning.com

Program-5:

```
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
void *thread_function(void *args);
int i,j;
int main()
{
pthread_t t_id;
printf("Process id=%d\n",getpid());
printf("Before creation of thread\n");
pthread_create(&t_id,NULL,thread_function,NULL);
pthread_join(t_id,NULL);
printf("Inside main program\n");
for(j=20;j<=25;j++)
{
printf("%d\n ",j);
sleep(1);
}
//printf("Thread Terminated \n");
return 0;
}
void *thread_function(void *args)
{
printf("Hello We are inside Thread \n");
for(i=1;i<=5;i++)
{
printf("%d\n ",i);
sleep(1);
}
sleep(30);
}
```

All Online Learning

www.allonlinelearning.com

Output:

```
lm@lm-VirtualBox:~$ gcc thread2.c -o thread2 -lpthread
```

```
lm@lm-VirtualBox:~$ ./thread2
```

```
Process id=4059
```

```
Before creation of thread
```

```
Hello We are inside Thread
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
Inside main program
```

```
20
```

```
21
```

```
22
```

```
23
```

```
24
```

```
25
```

If we comment `pthread_join(t_id, NULL);` line

Output:

```
lm@lm-VirtualBox:~$ gcc thread2.c -o thread2 -lpthread
```

```
lm@lm-VirtualBox:~$ ./thread2
```

```
Process id=4072
```

```
Before creation of thread
```

```
Inside main program
```

```
20
```

```
Hello We are inside Thread
```

```
1
```

```
21
```

```
2
```

```
22
```

```
3
```

All Online Learning

www.allonlinelearning.com

23

4

24

5

25