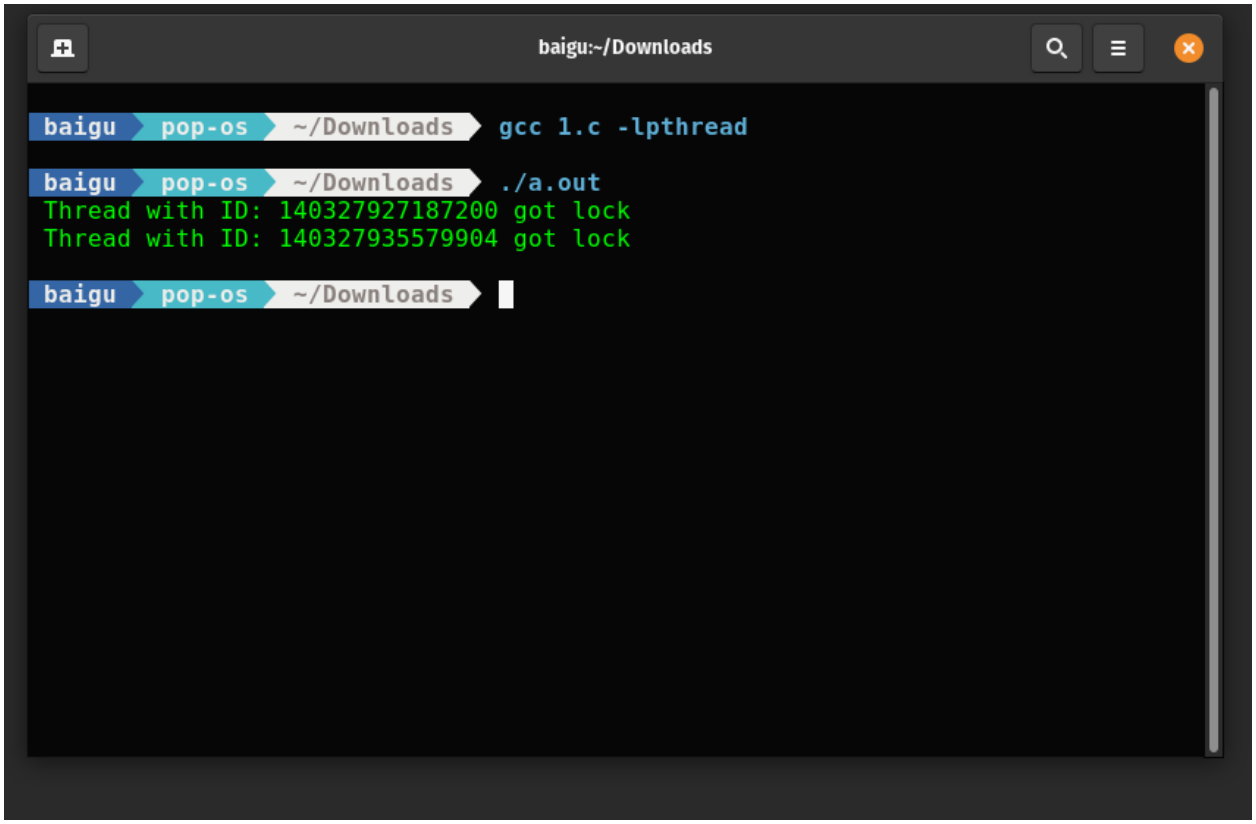
Example 1

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

sem_t sem ;

void *f1 ( void * arg )
{
        sem_wait(&sem ) ;
        printf ( " Thread with ID:% ld got lock\n" , pthread_self() ) ;
        sleep ( 2 ) ;
        sem_post(&sem ) ;
        pthread_exit (NULL) ;
}

int main ( )
{
        pthread_t tid1 , tid2 ;
        sem_init (&sem , 0 , 1 ) ;
        pthread_create (& tid1 ,NULL, f1 ,NULL) ;
        pthread_create (& tid2 ,NULL, f1 ,NULL) ;
        pthread_exit (NULL) ;
}
```



Example 2

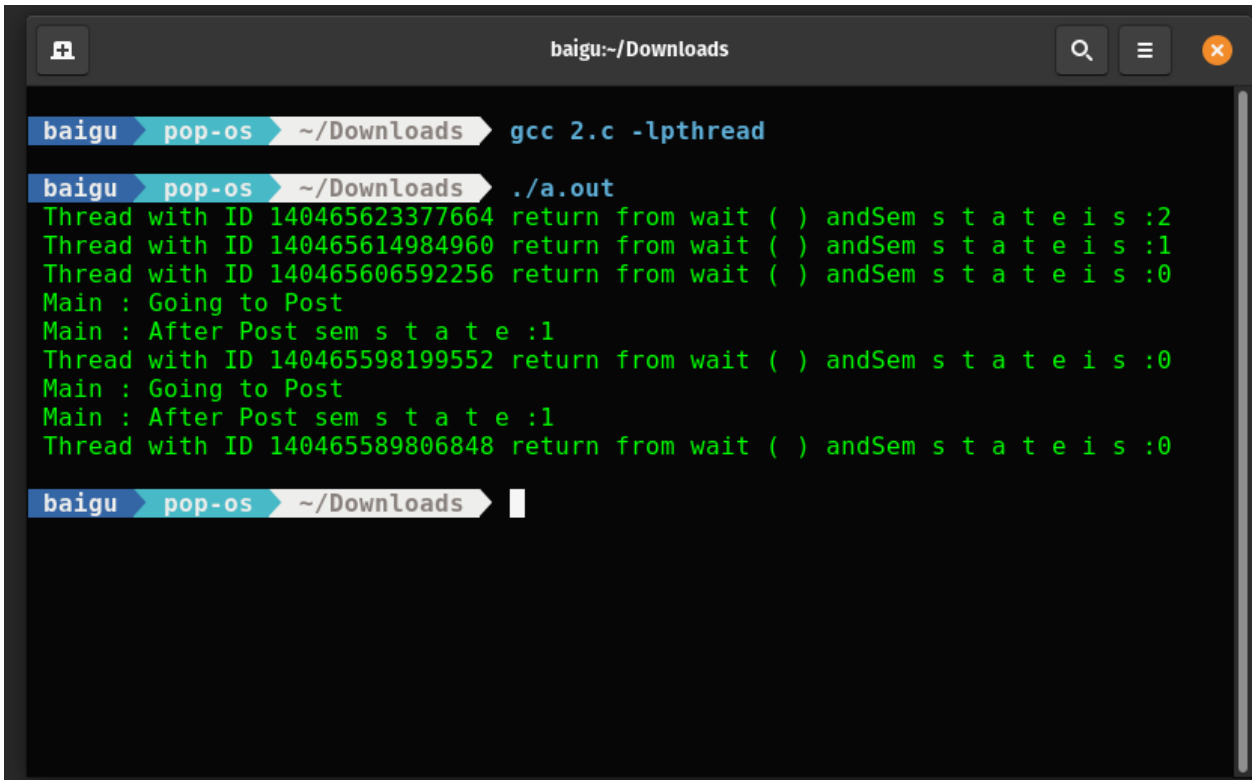
```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

sem_t sem;

void *f1(void *arg)
{
        int i = 1, value;
        sem_wait(&sem);
        sem_getvalue(&sem, &value);
        printf(" Thread with ID %ld return from wait ( ) andSem s t a t e i s :%d\n", pthread_self(),
value);

        pthread_exit(0);
}
```

```
int main()
{
        pthread_t tid[5];
        int value, i;
        sem_init(&sem, 0, 3);
        for (i = 0; i < 5; i++){
        pthread_create(&tid[i], NULL, f1, NULL);
        }
        sleep(3);
        printf(" Main : Going to Post\n");
        sem_post(&sem);
        sem_getvalue(&sem, &value);
        printf(" Main : After Post sem s t a t e :%d \n", value);
        sleep(3);
        printf(" Main : Going to Post\n");
        sem_post(&sem);
        sem_getvalue(&sem, &value);
        printf(" Main : After Post sem s t a t e :%d \n", value);
        pthread_exit(0);
}
```



Example 3

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

sem_t sem;

void *f1(void *arg)
{
        printf ( " Child :  begin\n");
        sem_post(&sem ) ;
        printf ( " Child :  end\n");
        pthread_exit (NULL) ;
}

int main()
{
        pthread_t tid;
        sem_init(&sem, 0, 0);
        printf(" parent : begin\n");
        pthread_create(&tid, NULL, f1, NULL);
        sem_wait(&sem);
        printf(" parent : end\n");
        pthread_exit(NULL);
}
```


baigu:~/Downloads
baigu pop-os ~/Downloads gcc 3.c -lpthread
baigu pop-os ~/Downloads ./a.out
parent : begin
Child : begin
Child : end
parent : end
baigu pop-os ~/Downloads