

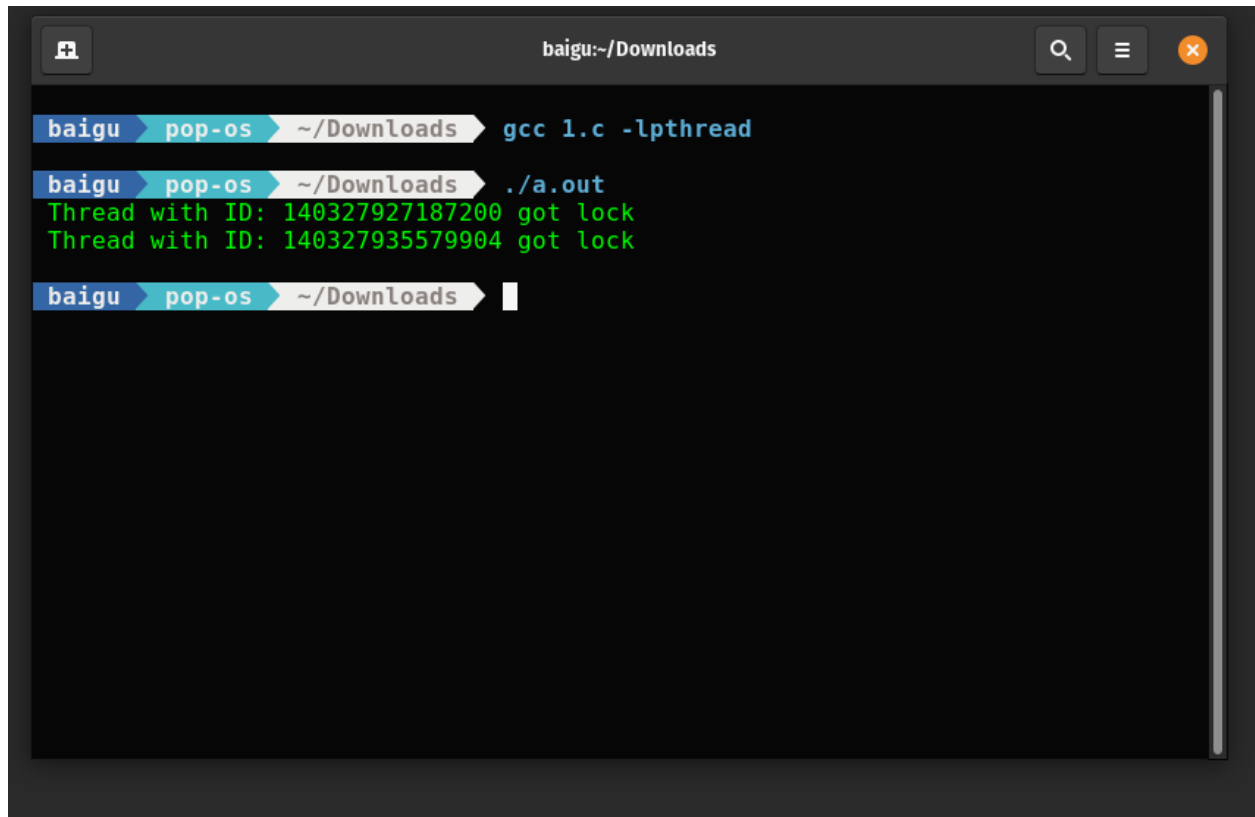
Example 1

```
#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:%d 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) ;
}
```

A terminal window titled 'baigu:~/Downloads' with search, menu, and close buttons in the title bar. The terminal shows three command prompts. The first prompt is 'baigu pop-os ~/Downloads' followed by the command 'gcc 1.c -lpthread'. The second prompt is 'baigu pop-os ~/Downloads' followed by the command './a.out', which produces two lines of green output: 'Thread with ID: 140327927187200 got lock' and 'Thread with ID: 140327935579904 got lock'. The third prompt is 'baigu pop-os ~/Downloads' followed by a cursor, indicating the terminal is ready for the next command.

```
baigu:~/Downloads
baigu pop-os ~/Downloads gcc 1.c -lpthread
baigu pop-os ~/Downloads ./a.out
Thread with ID: 140327927187200 got lock
Thread with ID: 140327935579904 got lock
baigu pop-os ~/Downloads
```

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);
}

```

```

baigu ~/Downloads
baigu pop-os ~/Downloads gcc 2.c -lpthread
baigu pop-os ~/Downloads ./a.out
Thread with ID 140465623377664 return from wait ( ) andSem s t a t e i s :2
Thread with ID 140465614984960 return from wait ( ) andSem s t a t e i s :1
Thread with ID 140465606592256 return from wait ( ) andSem s t a t e i s :0
Main : Going to Post
Main : After Post sem s t a t e :1
Thread with ID 140465598199552 return from wait ( ) andSem s t a t e i s :0
Main : Going to Post
Main : After Post sem s t a t e :1
Thread with ID 140465589806848 return from wait ( ) andSem s t a t e i s :0
baigu pop-os ~/Downloads

```

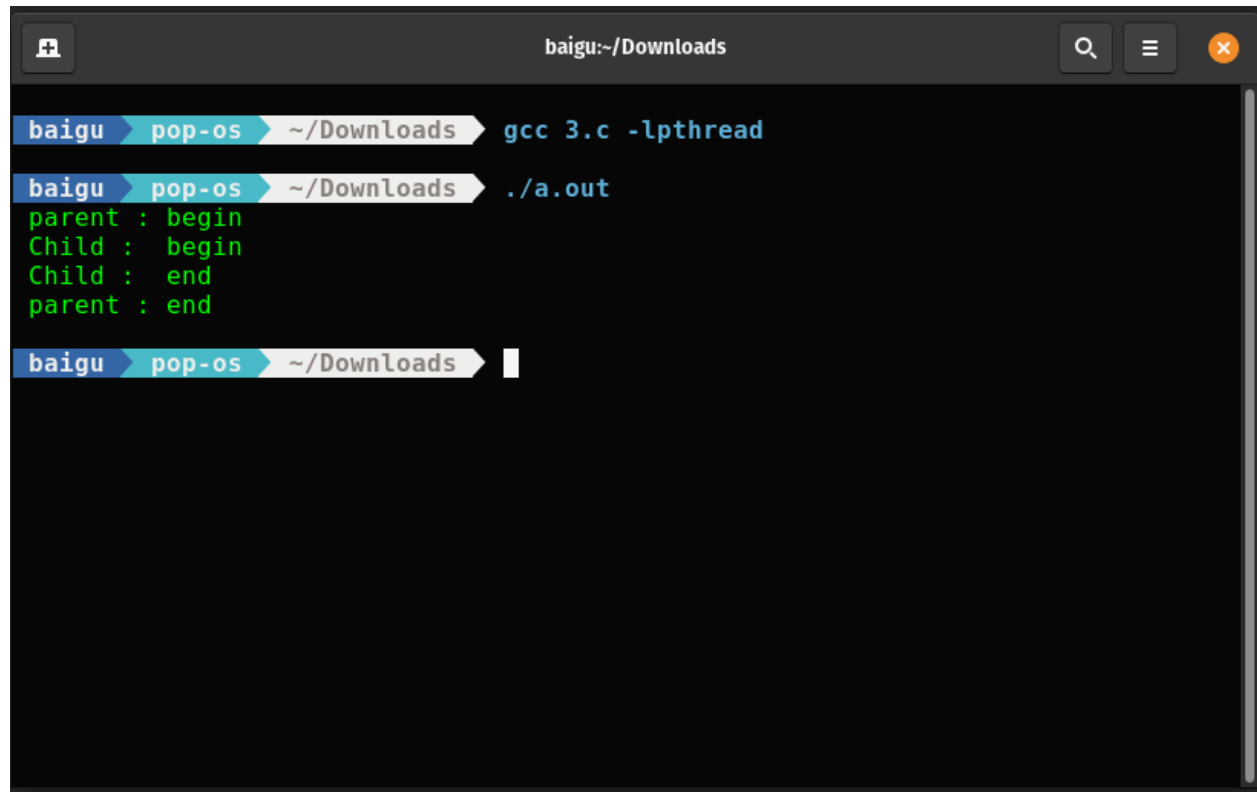
Example 3

```
#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);
}
```



A terminal window titled "baigu:~/Downloads" with standard window controls (minimize, maximize, close) on the right. The terminal shows a user named "baigu" in a "pop-os" environment at the directory "~/Downloads". The user runs the command "gcc 3.c -lpthread", which compiles a file named "3.c" into an executable named "a.out". Then, the user runs "./a.out", which produces the following output: "parent : begin", "Child : begin", "Child : end", and "parent : end". The output indicates that a parent process and a child process (created via threads) both begin and end their execution. The terminal has a dark background with light blue text for prompts and commands, and green text for the program's output. A vertical scrollbar is visible on the right side of the terminal window.

```
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 ➤
```