互斥锁是一种简单的加锁的方法来控制对共享资源的访问，互斥锁只有两种状态,即上锁( lock )和解锁( unlock )。

互斥锁的数据类型是： **pthread\_mutex\_t**。

# 互斥锁基本操作



以下函数需要的头文件：

#include <pthread.h>

**初始化互斥锁**

int pthread\_mutex\_init(pthread\_mutex\_t \*mutex, const pthread\_mutexattr\_t \*attr);

功能：

初始化一个互斥锁。

参数：

**mutex**：互斥锁地址。类型是 pthread\_mutex\_t 。  
 **attr**：设置互斥量的属性，通常可采用默认属性，即可将 attr 设为 NULL。

返回值：

成功：0，成功申请的锁默认是打开的。

失败：非 0 错误码

**上锁**

int pthread\_mutex\_lock(pthread\_mutex\_t \*mutex);

功能：

对互斥锁上锁，若互斥锁已经上锁，则调用者一直阻塞，直到互斥锁解锁后再上锁。

参数：

**mutex**：互斥锁地址。

返回值：

成功：0

失败：非 0 错误码

**试图上锁**

int pthread\_mutex\_trylock(pthread\_mutex\_t \*mutex);

调用该函数时，若互斥锁未加锁，则上锁，返回 0；若互斥锁已加锁，则函数直接返回失败，即 EBUSY。

**解锁**

int pthread\_mutex\_unlock(pthread\_mutex\_t \* mutex);

功能：

对指定的互斥锁解锁。

参数：

**mutex**：互斥锁地址。

返回值：

成功：0

失败：非 0 错误码

**销毁互斥锁**

int pthread\_mutex\_destroy(pthread\_mutex\_t \*mutex);

功能：

销毁指定的一个互斥锁

参数：

**mutex**：互斥锁地址。

返回值：

成功：0

失败：非 0 错误码

# 互斥锁应用实例

示例代码如下：

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

pthread\_mutex\_t mutex; //互斥锁

// 打印机

void printer(char \*str)

{

    pthread\_mutex\_lock(&mutex); //上锁

    while(\*str!='\0')

    {

        putchar(\*str);

        fflush(stdout);

        str++;

        sleep(1);

    }

    printf("\n");

    pthread\_mutex\_unlock(&mutex); //解锁

}

// 线程一

void \*thread\_fun\_1(void \*arg)

{

    char \*str = "hello";

    printer(str); //打印

}

// 线程二

void \*thread\_fun\_2(void \*arg)

{

    char \*str = "world";

    printer(str); //打印

}

int main(void)

{

    pthread\_t tid1, tid2;

    pthread\_mutex\_init(&mutex, NULL); //初始化互斥锁

    // 创建 2 个线程

    pthread\_create(&tid1, NULL, thread\_fun\_1, NULL);

    pthread\_create(&tid2, NULL, thread\_fun\_2, NULL);

    // 等待线程结束，回收其资源

    pthread\_join(tid1, NULL);

    pthread\_join(tid2, NULL);

    pthread\_mutex\_destroy(&mutex); //销毁互斥锁

    return 0;

}

运行结果如下：

