

## 五级项目 4：线程实现读者写者问题

### 1 学时

- 1 学时

### 2 实验目的

- 理解、掌握、应用 Linux 下使用线程间同步方式实现具体问题。

### 3 实验内容

- “读者-写者”问题描述：有 100 个读线程（reader）和 100 个写线程（writer）共同读、写文件。要求：

- （1）允许多个 reader 同时读一个文件；
- （2）当有一个 reader 在读文件时，不允许 writer 写文件；
- （3）当有一个 writer 在写文件时，不允许 reader 读文件，也不允许其它 writer 写文件。

（4）可以用读写锁解决“读者-写者”问题。在此对文件的读、写采用标准的输入和输出进行模拟，读、写文件锁用的的时间开销，通过让线程 sleep 一个随机的时间（0~1 秒）来实现，程序运行后首先创建 100 个读线程和 100 个写线程。

### 4 实验原理

#### （1）线程基本操作

关于线程基本操作的函数如下几个表所示。

表 Linux 系统关于线程的基本操作

名称	说明
pthread_create	创建新线程
pthread_self	获取线程 ID
pthread_exit	线程退出
pthread_join	线程等待

表 线程创建函数 pthread\_create

项目	描述
头文件	#include <pthread.h>
原型	int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine)(void*), void *arg);
功能	创建新线程
参数	thread: 当线程创建成功时，返回新线程 ID attr: 线程属性，NULL 表示默认属性

	start_routine: 线程创建后调用的函数, 也称线程函数 arg: 传递给线程函数的参数
返回值	成功返回 0, 失败返回错误代码

表 获取线程 ID 函数 pthread\_self

项目	描述
头文件	#include <pthread.h>
原型	pthread_t pthread_self(void);
功能	获取线程 ID
参数	void
返回值	返回线程的 ID

表 线程退出函数 pthread\_exit

项目	描述
头文件	#include <pthread.h>
原型	void pthread_exit(void *retval);
功能	线程退出
参数	retval: 线程结束时的返回值, 由 pthread_join()接收
返回值	无返回值

表 8.5 线程等待函数 pthread\_join 语法

项目	描述
头文件	#include <pthread.h>
原型	int pthread_join(pthread_t thread, void **retval);
功能	线程等待
参数	thread: 等待线程的 ID retval: 不为 NULL 时, retval 为被等待线程结束时的返回值
返回值	成功返回 0, 失败返回错误代码

## (2)线程间通信

线程间共享进程的地址空间, 因此线程间通信的难点在于对共享资源访问时的同步和互斥。线程等待函数 pthread\_join 使线程挂起, 等待另一线程结束后再运行, 可以解决简单的线程同步。除此之外, Linux 系统还提供了多种方式处理线程间的同步、互斥问题, 其中比较常用的有互斥锁、条件变量、信号量和读写锁等。

互斥锁的相关操作主要有互斥锁的初始化、销毁、加锁和解锁等, 相关函数如表所示。

表 互斥锁的相关函数

名称	说明
pthread_mutex_init	初始化互斥锁

pthread_mutex_destroy	销毁互斥锁
pthread_mutex_lock	对互斥锁加锁
pthread_mutex_unlock	对互斥锁解锁
pthread_mutex_trylock	尝试加锁，若不成功（已经上锁），则立刻返回

表 创建互斥锁函数 pthread\_mutex\_init

项目	描述
头文件	#include <pthread.h>
原型	int pthread_mutex_init(pthread_mutex_t *restrict mutex, const pthread_mutexattr_t *restrict attr);
功能	初始化互斥锁
参数	mutex: 互斥锁 attr: 互斥锁属性， NULL 表示默认属性
返回值	成功返回 0，失败返回错误代码

注意：互斥锁使用前应先初始化，可以使用上述函数 pthread\_mutex\_init 动态初始化，也可以静态初始化：

```
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
初始化后就可以使用加锁、解锁函数。
```

表 互斥锁加锁函数 pthread\_mutex\_lock

项目	描述
头文件	#include <pthread.h>
原型	int pthread_mutex_lock(pthread_mutex_t *mutex);
功能	对互斥锁加锁
参数	mutex: 互斥锁
返回值	成功返回 0，失败返回错误代码

表 互斥锁解锁函数 pthread\_mutex\_unlock

项目	描述
头文件	#include <pthread.h>
原型	int pthread_mutex_unlock(pthread_mutex_t *mutex);
功能	对互斥锁解锁
参数	mutex: 互斥锁
返回值	成功返回 0，失败返回错误代码

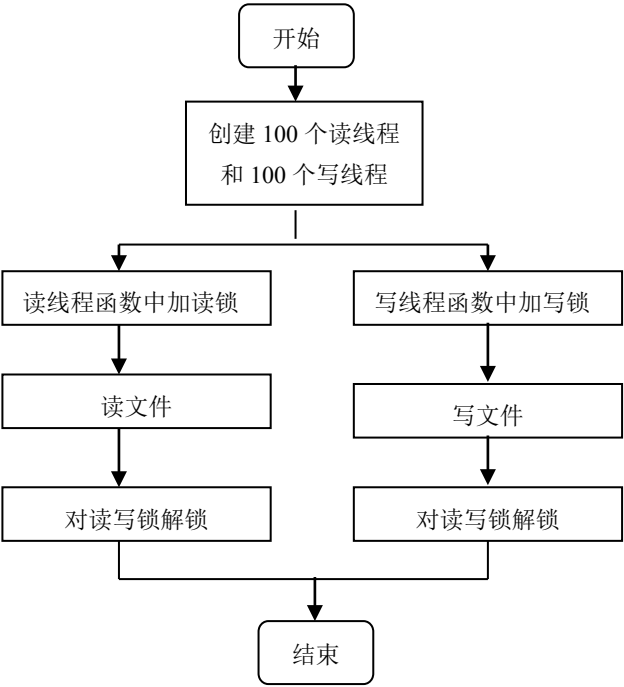
表 互斥锁测试加锁函数 pthread\_mutex\_trylock

项目	描述
头文件	#include <pthread.h>
原型	int pthread_mutex_unlock(pthread_mutex_t *mutex);
功能	尝试对互斥锁解锁，若失败则立刻返回
参数	mutex: 互斥锁
返回值	成功返回 0，失败返回错误代码

表 互斥锁销毁函数 pthread\_mutex\_destroy

项目	描述
头文件	#include <pthread.h>
原型	int pthread_mutex_unlock(pthread_mutex_t *mutex);
功能	注销互斥锁
参数	mutex: 互斥锁
返回值	成功返回 0，失败返回错误代码

(3)项目流程图如下图所示：



## 5 预习要求和技术准备工作

- 掌握 Linux 基本操作
- 掌握 C 语言开发工具的使用
- 掌握线程基本操作，线程间通信的方式的使用。

## 6 实验环境

- PC 机
- 在 Windows 环境中的 VMware 虚拟机上运行 Ubuntu 操作系统或者独立的 Ubuntu 操作系统
- 基于 Linux 的 vi 编辑器和 gcc 编译器

## 7 实验设计及操作步骤

7.1 以 root 身份登录系统，在/home 目录中创建目录 exp54

```
cd /home
```

```
mkdir exp54
```

7.2 进入刚创建的目录

```
cd exp54
```

7.3 使用 vi 编辑文件，文件名是 thread\_rdwr.c。

7.4 编写程序，实现要求的功能

7.5 编译可执行文件。

```
gcc -o thread_rdwr thread_rdwr.c -lpthread
```

7.6 运行程序

```
./thread_rdwr
```

结果演示（部分结果，仅供参考）如图所示。

```
Reader ID: 9; Reader Num: 1; Writer Num 0
Reader ID: 18; Reader Num: 2; Writer Num 0
Reader ID: 80; Reader Num: 3; Writer Num 0
Reader ID: 7; Reader Num: 4; Writer Num 0
Reader ID: 20; Reader Num: 5; Writer Num 0
Reader ID: 64; Reader Num: 6; Writer Num 0
Reader ID: 0; Reader Num: 7; Writer Num 0
Reader ID: 56; Reader Num: 8; Writer Num 0
Reader ID: 84; Reader Num: 9; Writer Num 0
Reader ID: 66; Reader Num: 10; Writer Num 0
Reader ID: 91; Reader Num: 11; Writer Num 0
Reader ID: 31; Reader Num: 12; Writer Num 0
Reader ID: 47; Reader Num: 13; Writer Num 0
Reader ID: 85; Reader Num: 14; Writer Num 0
Reader ID: 49; Reader Num: 14; Writer Num 0
Reader ID: 14; Reader Num: 15; Writer Num 0
```

.....

```
Writer ID: 41; Reader Num: 0; Writer Num 1
Writer ID: 22; Reader Num: 0; Writer Num 1
Writer ID: 62; Reader Num: 0; Writer Num 1
Writer ID: 39; Reader Num: 0; Writer Num 1
Writer ID: 51; Reader Num: 0; Writer Num 1
Writer ID: 72; Reader Num: 0; Writer Num 1
Writer ID: 13; Reader Num: 0; Writer Num 1
Writer ID: 76; Reader Num: 0; Writer Num 1
Writer ID: 35; Reader Num: 0; Writer Num 1
Writer ID: 57; Reader Num: 0; Writer Num 1
Writer ID: 85; Reader Num: 0; Writer Num 1
Writer ID: 73; Reader Num: 0; Writer Num 1
```

图 运行结果

## 8 实验报告提交要求:

将实验操作每个步骤中的命令、源程序以及截图写入实验报告，实验报告命名为“学号姓名-实验 54.doc”，交给指定人员。

## 9 项目思考

比较读写锁、互斥锁、信号量的特点和区别。