# 



**课 程 实 验 报 告**

**课程名称： 操作系统原理**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期：**

**计算机科学与技术学院**

**目录**

[实验一 熟悉Linux下的编程环境 1](#_Toc509858049)

[**1.1实验目的** 1](#_Toc509858050)

[**1.2 实验内容** 1](#_Toc509858051)

[**1.3 实验过程** 1](#_Toc509858052)

[1.3.1 编程思路 2](#_Toc509858053)

[1.3.2 遇到的问题及解决方式 3](#_Toc509858054)

[1.3.3 实验测试与结果分析 3](#_Toc509858055)

[**1.4 实验总结** 4](#_Toc509858056)

[实验二 掌握添加系统调用的方法 5](#_Toc509858057)

[**2.1 实验目的** 5](#_Toc509858058)

[**2.2 实验内容** 5](#_Toc509858059)

[**2.3 实验过程** 5](#_Toc509858060)

[2.3.1 编程思路 5](#_Toc509858061)

[2.3.2 遇到的问题及解决方式 5](#_Toc509858062)

[2.3.3 实验测试与结果分析 5](#_Toc509858063)

[**2.4 实验总结** 5](#_Toc509858064)

[实验三 共享内存与进程同步 6](#_Toc509858065)

[**3.1 实验目的** 6](#_Toc509858066)

[**3.2 实验内容** 6](#_Toc509858067)

[**3.3 实验过程** 6](#_Toc509858068)

[3.3.1 编程思路 6](#_Toc509858069)

[3.3.2 遇到的问题及解决方式 6](#_Toc509858070)

[3.3.3 实验测试与结果分析 6](#_Toc509858071)

[**3.4 实验总结** 6](#_Toc509858072)

[实验四 Linux文件目录 7](#_Toc509858073)

[**4.1 实验目的** 7](#_Toc509858074)

[**4.2 实验内容** 7](#_Toc509858075)

[**4.3 实验过程** 7](#_Toc509858076)

[4.3.1 编程思路 7](#_Toc509858077)

[4.3.2 遇到的问题及解决方式 7](#_Toc509858078)

[4.3.3 实验测试与结果分析 7](#_Toc509858079)

[**4.4 实验总结** 7](#_Toc509858080)

# 实验一 熟悉Linux下的编程环境

## **1.1实验目的**

1、掌握Linux操作系统的使用方法，包括键盘命令、系统调用；

2、熟悉Linux下的编程环境。

## **1.2 实验内容**

1．编一个C程序，其内容为实现文件拷贝的功能(使用系统调用open/read/write...）；

2．编一个C程序，其内容为分窗口同时显示三个并发进程的运行结果。要求用到Linux下的图形库 (gtk/Qt)。 如三个进程誊抄演示。

## **1.3 实验过程**

1. 首先安装编程环境，从ubuntu官网下载ubuntu16.04.3系统镜像，通过虚拟机软件VMware安装虚拟机。在ubuntu虚拟机下编写一个c程序，用系统调用(open/writed等)而不是库函数(fwrite/fopen等)来实现文件誊抄，编写完成后用ctrl+alt+t快捷键打开终端，进入主目录/home/lumos，使用gcc -o proj1\_copy proj1\_copy.c命令来进行编译，成功生成proj1\_copy执行文件，在终端中用./proj1\_copy命令，执行结果如图1.1所示，成功生成FileWrite文件。

图1.1 拷贝程序执行结果

2. 第二个实验，先安装gtk+3.0的编程环境，通过以下四步完成：

(1)、安装gcc/g++/gdb/make等基本编程工具：sudo apt-get install build-essential

(2)、安装GTK+3.0：sudo apt-get install libgtk-3-dev

(3)、安装pkg-config：sudo apt-get install pkg-config

(4)、安装帮助文件，方便查看帮助：sudo apt-get install devhelp

安装完成后，确认一下GTK+版本 pkg-config –modversion gtk+-3.0，结果如图1.2所示，GTK+版本为。

图1.2 GTK+版本

随后编写三个并发进程的程序：一个程序显示当前时间，一个显示从1加到100的累加求和，一个显示从100到1的倒计时，刷新周期都为1秒。这里把累加求和作为主进程，另外两个窗口作为子进程。

编写完成后通过指令gcc `pkg-config --cflags gtk+-3.0` -o proj1\_Three proj1\_Three.c `pkg-config --libs gtk+-3.0` 进行编译，编译结果如图1.3所示，成功生成proj1\_Three执行文件。

图1.3 三个窗口显示三个并发进程

### 1.3.1 编程思路

1. 文件拷贝程序：通过系统调用open("/home/lumos/FileRead.txt",O\_RDONLY);用只读方式打开FileRead文件，再次使用open("/home/lumos/FileWrite.txt",O\_WRONLY|O\_CREAT|O\_TRUNC,S\_IRUSR|S\_IWUSR) 用读写方式打开一个文件FileWrite，O\_TRUNC参数的作用是当文件FileWrite已经存在时，把这个文件清空，再进行写入操作，S\_IRUSR|S\_IWUSR则是文件权限的管理。打开两个文件之后，在一个循环中，通过read(r\_fp,&buffer,sizeof(char)); 把文件FileRead读取到缓冲区buffer中，逐个字符读取，然后逐个字符地写入文件FileWrite中，直到读取完成后跳出循环。

2. 三个窗口显示三个并发进程：首先通过fork()创建两个子进程，在每个子进程内并行调用对应的窗口初始化函数。三个进程的窗口初始化程序类似，以累计求和窗口的初始化函数init\_add为例：

先进行环境初始化gtk\_init(&argc,&argv); 在程序使用到GTK+工具库之前，必须对它进行初始化；然后通过gtk\_window\_new(GTK\_WINDOW\_TOPLEVEL) 创建一个窗口；再对这个窗口的属性进行设置，如标题、窗口宽度、窗口大小；这些窗口的基本属性设置完成后把关闭窗口信号绑定到一个关闭事件，当发出关闭信号时，进入回调函数"destroy"，使程序结束；然后把一个标签绑定到这个窗口上，用于显示对应信息，注意这个标签一定要使用label=gtk\_label\_new(NULL)进行实例化，通过g\_timeout\_add(1000,getAdd,(void \*)label)来控制每隔1000ms，即一秒钟刷新一次数据，最后显示所有窗口，并用gtk\_main()作为主循环，至此窗口程序结束。

三个窗口区别主要在于获取的数据不同：getAdd函数中使用一个全局变量count进行累加，并把这个数据写入到字符缓冲buf中，然后通过gtk\_label\_set\_markup(GTK\_LABEL(data), buf)把缓冲区里的数据写入到标签里；倒计时程序与累加求和类似，不赘述；getTime函数用于获取系统当前时间，用到两个结构体time\_t和tm，前者用来存储从1970年到现在经过了多少秒，后者存储的是年月日这样的具体时间结构，通过stuct tm\* localtime(const time\_t \*timep) 函数获取经过时区转换的当前时间，再通过gchar \*time\_dis = g\_strdup\_printf("%2d:%2d:%2d",(time\_tm->tm\_hour), (time\_tm->tm\_min), (time\_tm->tm\_sec))来把需要用到的时间信息输出到字符缓冲区中，最后把字符缓冲区的输入到标签里，即完成了把当前时间信息显示在标签上的功能。

### 1.3.2 遇到的问题及解决方式

1. 文件拷贝程序：最初在打开目标文件时，使用的是open("/home/lumos/FileWrite.txt",O\_WRONLY|O\_CREAT|O\_TRUNC); 并没有声明文件的权限，导致生成的文件显示成加锁的模式，无法访问。解决方法是加上两个参数控制访存权限。S\_IRUSR的功能是允许文件的所有者阅读它，S\_IWUSR的功能是允许文件所有者写它。

2. 三个并发进程窗口程序：最初使用gcc -o proj1\_Three proj1\_Three.c `pkg-config --libs gtk+-3.0`指令进行编译，结果显示ld return-1，不能成功生成执行文件，编译错误如图1.4所示，后来查询相关资料，改用gcc `pkg-config --cflags gtk+-3.0` -o proj1\_Three proj1\_Three.c `pkg-config --libs gtk+-3.0` 进行编译，可以成功生成执行文件。

图1.4 新版GTK3编译错误

### 1.3.3 实验测试与结果分析

1. 文件拷贝程序：进行两项测试

(1).目标文件不存在时，执行拷贝程序后，观察能否生成正确的目标文件，要求内容与原文件相同。

先把FileWrite文件删除，执行一次拷贝程序，如图1.5所示，成功生成FileWrite文件，且内容与FileRead文件内容一致，程序正确运行。

图1.5 文件拷贝程序测试1

(2). 目标文件已经存在时，在其中输入与源文件中不同的数据，观察执行拷贝程序后目标文件的内容。

修改FileWrite文件内容为aaaaaa，执行一次拷贝程序，如图1.6所示， FileWrite文件内容与FileRead文件内容一致，程序正确运行。

图1.6 文件拷贝程序测试2

2. 三个窗口显示三个并发进程：直接执行程序，程序的执行结果如图1.7所示，程序正确运行。

图1.7 三个窗口显示三个并发进程测试

## **1.4 实验总结**

通过本次实验，熟悉了linux下的终端命令操作，学会了GTK的安装方法和基本用法，学会如何通过系统调用控制文件的读写及权限管理，了解了time\_t和tm两种时间结构体的内容，学会了如何获取当前时间，多进程的窗口直观显示也让我对并行进程有了更深的理解。

# 实验二 掌握添加系统调用的方法

## **2.1 实验目的**

1、掌握系统调用的实现过程；

2、通过编译内核方法，增加一个新的系统调用；

3、另编写一个应用程序，使用新增加的系统调用。

## **2.2 实验内容**

1. 内核编译、生成，用新内核启动；

2. 新增系统调用实现：文件拷贝或P、V操作。

## **2.3 实验过程**

1. 在[https://www.kernel.org/](https://www.kernel.org/。网站下载linux4.24.16) 网站下载linux4.24.16.tar.xz内核代码，进入linux系统，通过su指令进入管理员模式，把系统内核压缩包拷贝到目录/usr/src下，然后使用xz -d linux4.24.16.tar.xz和tar -xvf linux4.24.16.tar两条命令解压缩，安装一些基本的工具软件之后，进入linux4.24.16目录下，依次执行以下三条命令：

sudo make mrproper 作用是清楚编译过程中产生的所有中间文件；

sudo make clean 作用是清除上一次产生的编译中间文件；

sudo make menuconfig 出现选择的图形化界面之后，直接选择exit退出，弹出的提示里选择save保存，实现内核的默认配置。

随后执行sudo make -j4进行四线程编译，大概两小时之后，编译完成，在终端输入sudo make modules\_install来安装内核模块，安装完成后再输入sudo make install来安装内核。内核安装结束之后就重新启动系统，在虚拟机出现vmware字样时就要迅速的按住shift键，进入内核选择加载界面，选择linux4.24.16进入系统，就完成了使用新内核启动的功能。

2. 新增系统调用，实现文件拷贝功能，与实验一基本一致，只是在内核中进行打开，写文件等操作时需要使用get\_fs, set\_fs对他们进行保护。

### 2.3.1 编程思路

### 2.3.2 遇到的问题及解决方式

编译内核函数时应该使用sys\_open而不是open

Sysopen的参数在编译内核时设置成3个sys\_open(const char \_\_user \*filename, int flags, int mode)

### 2.3.3 实验测试与结果分析

## **2.4 实验总结**

# 实验三 共享内存与进程同步

## **3.1 实验目的**

1、掌握Linux下共享内存的概念与使用方法；

2、掌握环形缓冲的结构与使用方法；

3、掌握Linux下进程同步与通信的主要机制。

## **3.2 实验内容**

利用多个共享内存（有限空间）构成的环形缓冲，将源文件复制到目标文件，实现两个进程的誊抄。

## **3.3 实验过程**

### 3.3.1 编程思路

### 3.3.2 遇到的问题及解决方式

### 3.3.3 实验测试与结果分析

## **3.4 实验总结**

# 实验四 Linux文件目录

## **4.1 实验目的**

1、了解Linux文件系统与目录操作；

2、了解Linux文件系统目录结构；

3、掌握文件和目录的程序设计方法。

## **4.2 实验内容**

1. 功能类似ls -lR；

2. 查询指定目录下的文件及子目录信息；

3. 显示文件的类型、大小、时间等信息；

4. 递归显示子目录中的所有文件信息。

## **4.3 实验过程**

### 4.3.1 编程思路

### 4.3.2 遇到的问题及解决方式

### 4.3.3 实验测试与结果分析

## **4.4 实验总结**