**《Linux程序设计》试题A卷**

*2014-2015学年度第二学期（回忆版）*

**一、填空题（4分\*5=20分）**

1. 对于umask=022，生成文件的默认权限是\_\_~~755\_~~644\_\_\_。
2. Linux中可以调用文件操作函数分\_\_\_系统调用函数\_\_\_和C库函数。
3. 将文件按页打印的命令是\_\_more\_\_\_\_。
4. chmod的功能是\_\_\_修改文件权限\_\_\_。
5. Linux中文件描述符的数据类型是\_\_\_非负整数\_\_\_。

**二、简答题（40分）**

1. Linux中文件描述符和文件指针FILE \*的区别是什么？（9分）

文件描述符：在linux系统中打开文件就会获得文件描述符，它是个很小的正整数。每个进程在PCB（Process Control Block）中保存着一份文件描述符表，文件描述符就是这个表的索引，每个表项都有一个指向已打开文件的指针。

文件指针：C语言中使用文件指针做为I/O的句柄。文件指针指向进程用户区中的一个被称为FILE结构的数据结构。FILE结构包括一个缓冲区和一个文件描述符。而文件描述符是文件描述符表的一个索引，因此从某种意义上说文件指针就是句柄的句柄（在Windows系统上，文件描述符被称作文件句柄）。

1. 什么是操作系统内核？内核的主要功能是什么？（12分）

操作系统是一系列程序的集合，其中最重要的部分构成了内核

单内核/微内核

单内核是一个很大的进程，内部可以分为若干模块，运行时是一个独立的二进制文件，模块间通讯通过直接调用函数实现

微内核中大部分内核作为独立的进程在特权下运行，通过消息传递进行通讯

1. 比较软链接和硬链接的不同之处（至少三点），写出shell和应用程序中创建软硬链接的命令。（15分）

硬链接：

a.不同的文件名对应同一个inode

b.不能跨越文件系统

c.对应系统调用link

软链接：

a.存储被链接文件的文件名(而不是inode)实现链接

b.可跨越文件系统

c.对应系统调用symlink

shell:

ln –s [文件名] [文件名] 创建软连接

ls [文件名] [文件名] 创建硬链接

应用程序：

创建硬链接：

#include <unistd.h>

int link(const char \*oldpath, const char \*newpath);

(Return: 0 if success; -1 if failure)

创建软链接：

#include <unistd.h>

int symlink(const char \*oldpath, const char \*newpath);

(Return: 0 if success; -1 if failure)

1. 比较字符设备和块设备的区别，并举实例。（9分）

字符设备：**提供连续的数据流**，应用程序可以**顺序读取**，**通常不支持随机存取**。相反，此类设备支持**按字节/字符**来读写数据。举例来说，调制解调器是典型的字符设备。

 块设备： 应用程序可以**随机访问设备数据**，程序可**自行确定读取数据的位置**。硬盘是典型的块设备，应用程序可以寻址磁盘上的任何位置，并由此读取数据。此外，数据的读写只能以**块**(通常是512B)的倍数进行。与字符设备不同，块设备并不支持基于字符的寻址。

**三、论述编程题（40分）**

1. 重定向是Linux中的重要机制。试分析其使用方法、应用实例，并简述实现机制。（15分）

使用方法：linux中可是用shell的重定向符号

>为输出重定向

<为输入重定向

使用方法：

输出重定向: echo “linux”>1.txt //标准输出流重定向到1.txt

输入重定向：cat<1.txt //标准输入流重定向到1.txt

实现机制：

主要通过使用dup2系统调用通过明确指定目标描述符来把一个文件描述符复制为另外一个

举个例子：

int fd=-1;

char buffer[100];

int len=0;

fd=open("4.txt",O\_CREAT|O\_RDWR);

dup2(fd,STDOUT\_FILENO);

printf("hello world");

close(fd);

该段代码实现了标准输出的重定向，printf的打印的hello world写入4.txt文件

1. 使用C的库函数，编写一个函数void bindiff(char \*file1,char \*file2,char \*fileo)，将文件从file1、file2对应的路径中读取并逐字节比对，将相同的字节输出到fileo对应的文件中。（25分）

#include<stdio.h>

void bindiff(char \*file1, char \*file2, char \*fileo);

int main()

{

bindiff("1.txt","2.txt","3.txt");

}

void bindiff(char \*file1, char \*file2, char \*fileo)

{

FILE \* fp1 = 0,\*fp2=0,\*fpo=0;

char ch1, ch2;

fp1 = fopen(file1,"r");

fp2 = fopen(file2,"r");

fpo = fopen(fileo, "w");

while (1)

{

ch1 = (char)fgetc(fp1);

ch2 = (char)fgetc(fp2);

if (feof(fp1) || feof(fp2))

{

break;

}

if (ch1 == ch2)

{

fputc(ch2,fpo);

}

}

fclose(fp1);

fclose(fp2);

fclose(fpo);

}