

浙江大学 2004—2005 学年 冬 学期期终考试

《操作系统分析与实验》课程试卷

考试时间： 120 分钟 开课学院：_____ 专业：_____

姓名：_____ 学号：_____ 任课教师：_____

题序	一	二	三	四	五	总分
评分						
评阅人						

一、选择题。对于每一个小题，各选择一个正确的答案，填入“()”中（共40分，每题2分）

- 1.() 2.() 3.() 4.() 5.()
 6.() 7.() 8.() 9.() 10.()
 11.() 12.() 13.() 14.() 15.()
 16.() 17.() 18.() 19.() 20.()

- Linux内核的发布者是_____。
 A. Linus Torvards B. <http://www.kernel.org> C. RedHat D. <http://www.gnu.org>
- 目前Linux内核稳定的发行版本编号是_____。
 A. 9.0 B. 2.6.10 C. 2.4.18 D. 2.6.10-rc3-bk15
- 在bash shell提示符后，键入set并且按下<Enter>键。系统显示_____。
 A. 一个帮助界面，列出了shell常用命令；
 B. set命令应该随带的参数名称及其意义；
 C. 环境变量和它们当前设置的列表；
 D. shell的出错提示信息，表示set命令不存在。
- Linux继承了UNIX的传统，也提供在线帮助系统。当你使用man passwd时，系统不会显示_____。
 A. 与passwd命令相关的命令；
 B. passwd命令的格式；
 C. passwd命令的参数意义；
 D. passwd命令的历史。
- shell是Linux的命令解释器。以下关于shell的说法中，_____是错误的。
 A. command.com是一种LINUX shell；
 B. C shell是一种LINUX shell；
 C. TC shell是一种LINUX shell；
 D. Korn shell是一种LINUX shell。

6. 在Linux文件系统中，有一个目录叫“/dev”。该目录下的文件_____。
A. 是某个设备的逻辑标识； B. 也是包含一串二进制数的通用文件；
C. 是某类设备的逻辑标识； D. 以上全不对。
7. Linux的一个目录项(directory entry)_____。
A. 描述了该目录下，一个文件的信息； B. 是一种描述目录的数据结构；
C. 保存了某一子目录的全部信息； D. 保存了某一子目录或某一文件的全部信息。
8. 使用_____, 不能显示你的家目录的绝对路径。
A. 显示/etc/passwd的内容； B. cd
C. 查看环境变量\$PWD； D. echo \$HOME
9. 命令_____显示当前目录下文件名满足如下条件的所有文件的类型：文件名以chapter开始，后面跟着一个数字1、2、6、8或9，以.eps结尾。
A. ls chapter[12689].eps B. ls chapter[1,2,6,8,9].eps
C. file chapter[1,2,6,8,9].eps D. file chapter[12689].eps
10. 命令chmod =r sample实现命令_____的效果。
A. chmod 004 sample B. chmod 444 sample
C. chmod 040 sample D. chmod 400 sample
11. 进程的作业号为3，pid号为10974，并且忽略2号signal和15号signal。尽管如此，命令_____将终止它。
A. kill -9 %3 B. kill -2 %3
C. kill -9 %10974 D. kill -2 %10974
12. CPU调度算法_____没有在LINUX的2.4.18版中使用。
A. FCFS B. SJF
C. ROUND ROBIN D. priority
13. 命令_____编译power.c并把生成的可执行代码保存到XpowerY文件。
A. gcc -c XpowerY power.c B. gcc -g XpowerY power.c
C. gcc -o XpowerY power.c D. 以上命令都不能
14. SHELL终端中，输入“!vi”，将_____。
A. 不带参数执行vi命令 B. 执行以“vi”开头的一条命令
C. 无条件执行vi命令 D. 执行最近刚执行过的以“vi”开头的命令
15. 在一个磁盘分区中创建一个与Windows兼容的文件系统，可以用_____。
A. mknod B. mkdir
C. mkfs D. mkfs.ext2
16. Linux为每一个文件系统设置了magic number，其作用是_____。
A. 描述文件系统的类型 B. 记录了该文件系统中文件、目录总数
C. 表达了共享该文件系统的进程数 D. 唯一表示该文件系统的id
17. Linux的系统调用mount(), 属于_____。
A. 设备管理的操作 B. 文件系统的操作


```

        read_unlock(&current->fs->lock); break;
    }
    read_unlock(&current->fs->lock);
    spin_lock(&dcache_lock);
    if (nd->dentry != nd->mnt->mnt_root) {
        dentry = dget(nd->dentry->d_parent);
        spin_unlock(&dcache_lock);
        dput(nd->dentry);
        nd->dentry = dentry;
        break;
    }
    parent=nd->mnt->mnt_parent;
    if (parent == nd->mnt) {
        spin_unlock(&dcache_lock);
        break;
    }
    mntget(parent);
    dentry=dget(nd->mnt->mnt_mountpoint);
    spin_unlock(&dcache_lock);
    dput(nd->dentry);
    nd->dentry = dentry;
    mntput(nd->mnt);
    nd->mnt = parent;
}
}

struct nameidata {
    struct dentry *dentry; struct vfsmount *mnt; struct qstr last;
    unsigned int flags; int last_type;
};
struct vfsmount
{
    struct list_head mnt_hash;
    struct vfsmount *mnt_parent; /* fs we are mounted on */
    struct dentry *mnt_mountpoint; /* dentry of mountpoint */
    struct dentry *mnt_root; /* root of the mounted tree */
    struct super_block *mnt_sb; /* pointer to superblock */
    struct list_head mnt_mounts; /* list of children, anchored here */
    struct list_head mnt_child; /* and going through their mnt_child */
}

```

```

    atomic_t mnt_count; int mnt_flags;
    char *mnt_devname;      /* Name of device e.g. /dev/dsk/hda1 */
    struct list_head mnt_list;
};
struct dentry {
    atomic_t d_count;    unsigned int d_flags;
    struct inode * d_inode;    /* Where the name belongs to - NULL is negative */
    struct dentry * d_parent;    /* parent directory */
    struct list_head d_hash;    /* lookup hash list */
    struct list_head d_lru;    /* d_count = 0 LRU list */
    struct list_head d_child;    /* child of parent list */
    struct list_head d_subdirs; /* our children */
    struct list_head d_alias;    /* inode alias list */
    int d_mounted; struct qstr d_name;
    unsigned long d_time;    /* used by d_revalidate */
    struct dentry_operations *d_op;
    struct super_block * d_sb; /* The root of the dentry tree */
    unsigned long d_vfs_flags;
    void * d_fsdata;    /* fs-specific data */
    unsigned char d_iname[DNAME_INLINE_LEN]; /* small names */
};
struct fs_struct {
    atomic_t count; rwlock_t lock; int umask;
    struct dentry * root, * pwd, * alroot;
    struct vfsmount * rootmnt, * pwdmnt, * alrootmnt;
};

```

四、编程序，显示系统的缺页中断次数（20分）

(1) 请叙述Linux的缺页中断调用机理。

(2) 请给出你的显示方案。