《计算机操作系统》实验一报告

姓名： 学号： 实验日期：

一、实验环境：Linux系统

二、实验目的：

**1.** 了解Linux运行环境，熟悉交互式分时系统、多用户环境的的运行机制。

**2.** 练习Linux系统命令接口的使用，学会Linux基本命令、后台命令、管道命令等命 令的操作要点。

三、实验准备：

1.复习操作系统中相关的用户接口概念。

2.查阅Linux中Shell的资料，它既是一个命令解释程序，又是一个程序设计语言。

3.熟悉本《实验指导》第二部分，从中你可以学会Shell的一般命令

四、实验内容：

1.查看系统信息：

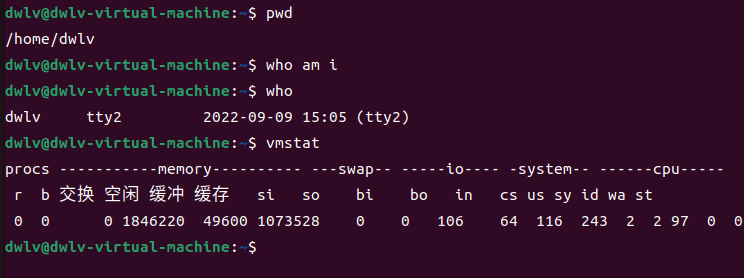
操作：

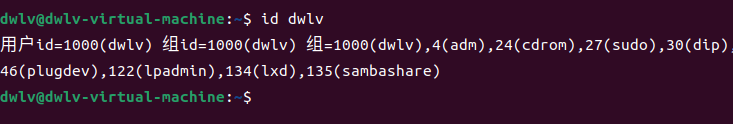
① 执行pwd查看当前目录。

② 用who am i看看当前用户信息。

③ 通过who看看有谁在系统中。

④ 用vmstat显示系统状态

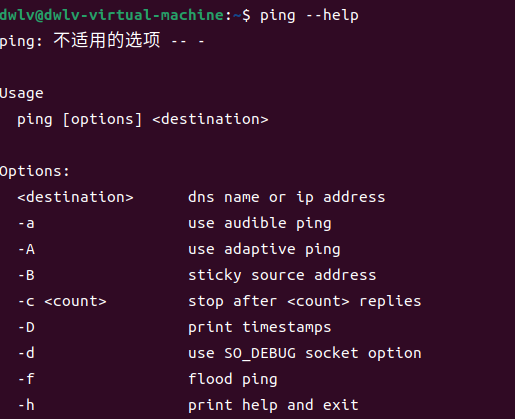


实验结果：输入相关指令后，终端内显示对应的系统信息，发现当前目录为/home/dwlv，当前只有一个用户dwlv，于15:05登录系统，同时也显示了体统的空闲、缓存等信息。

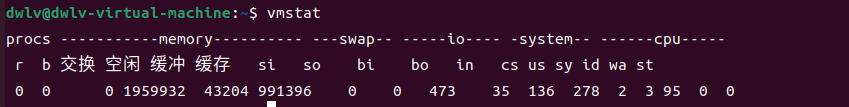
**思考：你的用户名、用户标识、组名、组标识是什么？当前你处在系统的哪个位置中？现在有哪些用户和你一块儿共享系统 ？**

答：我的用户名为dwlv，用户标识为1000，组名为dwlv，组标识为1000.当前位于/home/dwlv目录下，当前暂无用户和我共享系统。

**其他命令试验截图：**







1. 文件操作命令

操作：

① 执行

cat > mytext.txt

② 执行

cat mytext.txt

显示文件内容。

③ 执行`

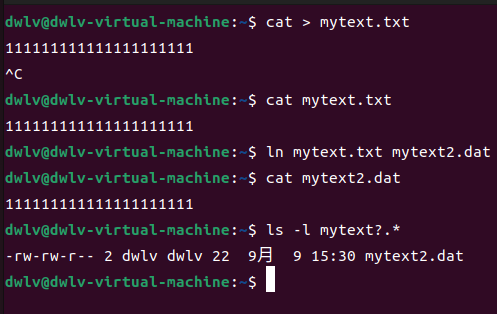
ln mytext.txt mytext2.dat

cat mytext2.dat

④ 执行

ls –l mytext?.\*

显示文件目录，注意i节点号，链接计数。

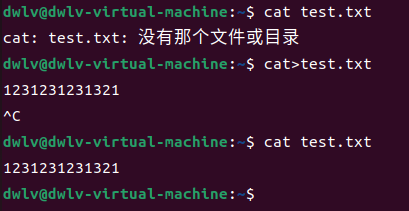


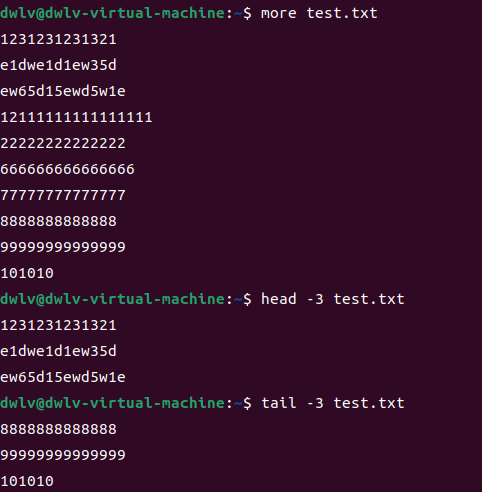
实验结果：创建了一个文件名为mytext.txt的文本文件，其中写入了“11111111111111111111”内容，然后他通过文件链接，创建了一个新的mytext2.dat文件，内容与mytext.txt中的一致。

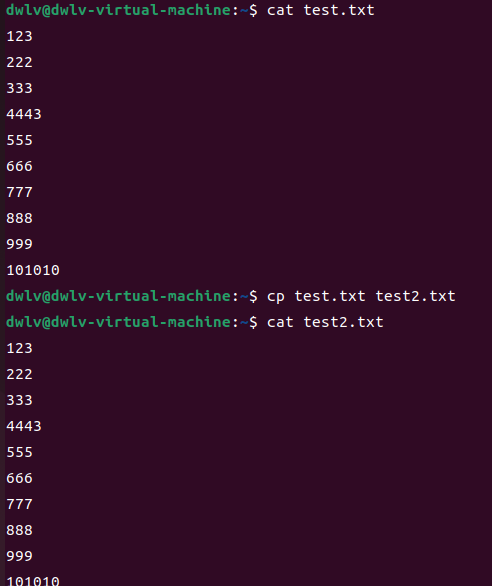
**思考：文件链接是什么意思？有什么作用？**

**答：文件链接相当于给系统内的某个文件制定了另一个新的访问它的名称，通过文件链接创造出的mytext2.dat文件其实就是mytext.txt文件的别名，任何一个文件被修改了，两个文件都会被同步修改。这样就可以共享文件内容。就算删除了链接，也不会破坏原有内容。**

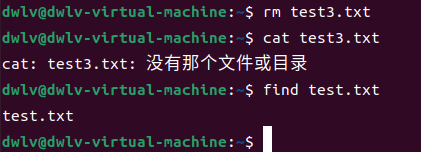
**其他命令试验：**











1. 目录操作

操作：

① 执行

ls –

l

看看当前目录的内容，请特别注意文件类型、文件的存取控制权限、i节点号、文件

属主、文件属组、文件大小、建立日期等信息。

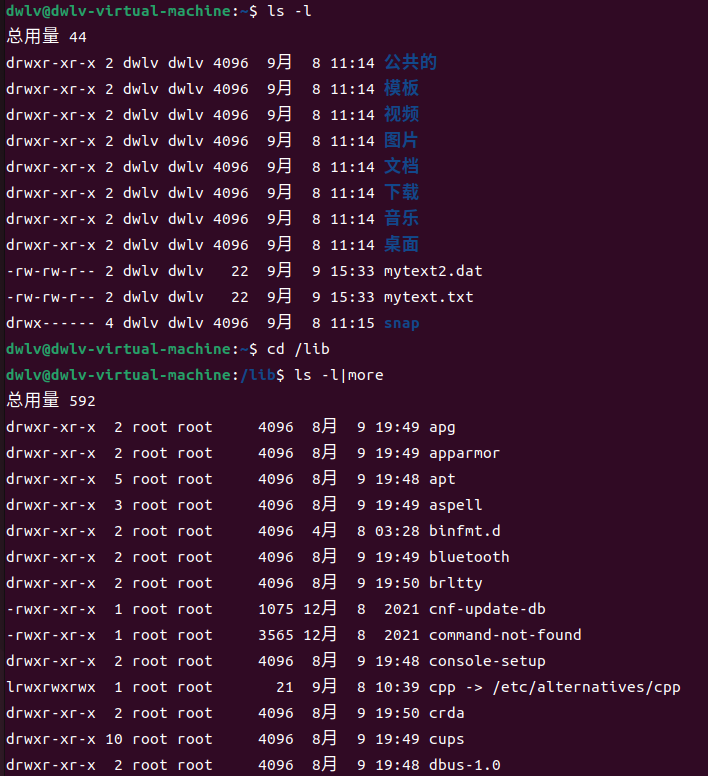
② 执行

cd /lib

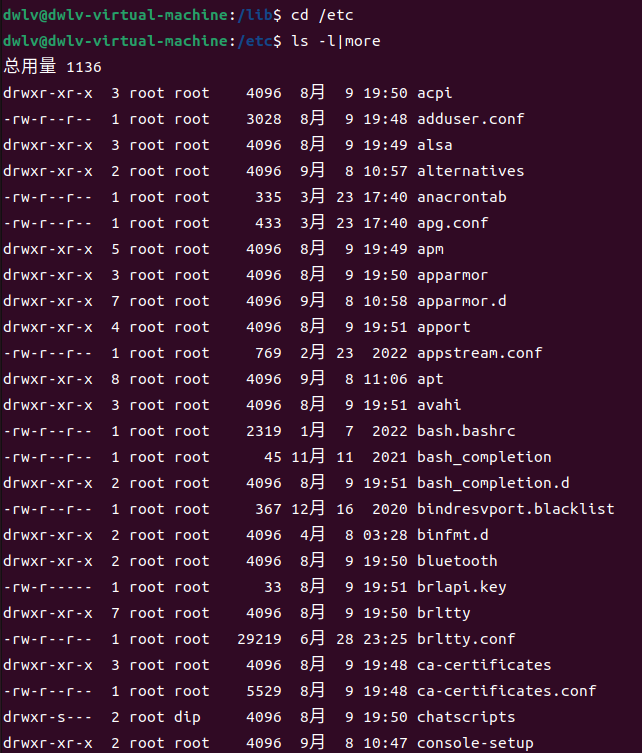
ls –l|more

看看/lib目录的内容，这里都是系统函数。再看看/etc，这里都是系统配置用的数据

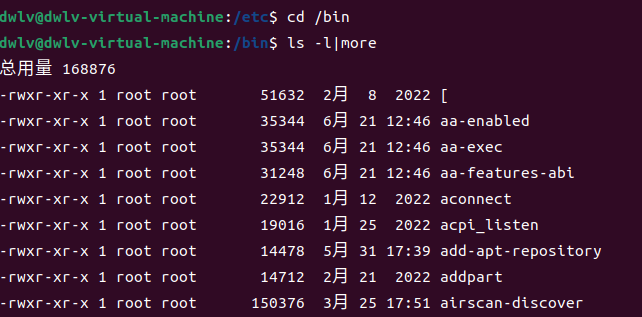
文件；/bin中是可执行程序；/home下包括了每个用户主目录。



实验结果：lib目录下共有592个文件，其中文件的类型、日期、权限、大小都被明确的标明。



实验结果：ect目录下共有1136个文件

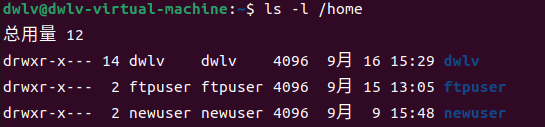


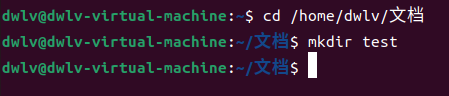
实验结果：bin目录下文件数量最多，为1668876个。

**思考：Linux文件类型有哪几种？文件的存取控制模式如何描述？**

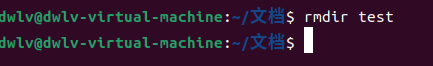
**答：文件类型有：普通文件、目录文件、符号连接、字符设备文件、块设备文件、命名管道FIFO、套接口socket。存取控制模式分为可读、可写、可执行三种。**

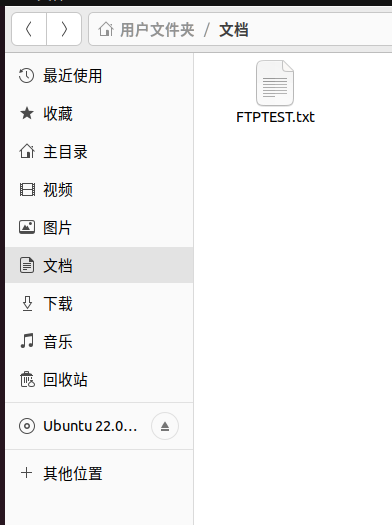
**其他命令试验:**











1. 修改文件属性

操作：

① 执行

chmod 751 mytext.txt

ls –l mytext.txt

② 执行

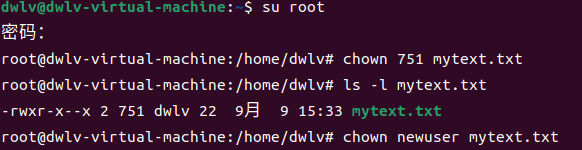
chown stud090 mytext.txt

（存取控制模式的表示可用八进制或字符表示，参见本

《实验指导》第二部分介绍的目录操作的内容。）

（查看文件mytext.txt的存取控制权限。）

（修改文件所有者为stud090）



实验结果：成功将mytext.txt文件的权限设置为751，并将文件所有者转移为newuser用户，再次使用cat命令修改改文件时提示没有权限，不能修改。

**思考：执行了上述操作后，若想再修该文件，看能不能执行。为什么？**

**答：不能修改改文件，因为改文件的所有者从原本的dwlv变成了newuser用户，原有用户没有修改该文件的权限，无法执行修改操作。**

1. 进程管理命令

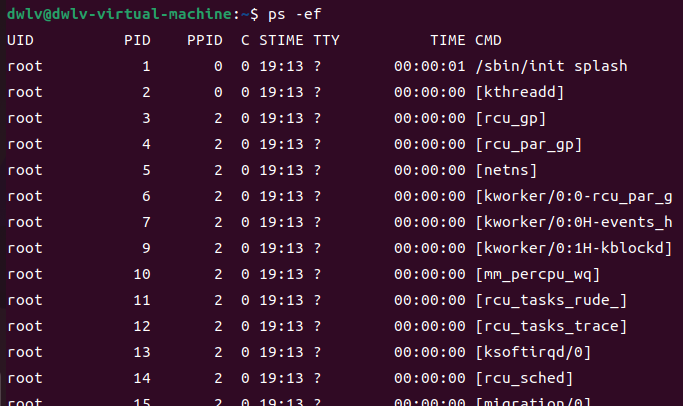
操作：

① 执行

ps –ef

根据本《实验指导》第二部分介绍的进程管理命令选项，查看当前系统中各个

进程的信息。特别注意进程号、父进程号、属主等内容。

② 执行本《实验指导》第二部分介绍的wait和sleep命令。 

实验结果：观察到列出的各个进程，其中，UID: 该进程执行的用户id，PID: 进程id，PPID: 该进程的父级进程id，C: cpu的占用率，形式是百分数（%），STIME: 进程的启动时间，TTY: 发起该进程的设备识别符号，TIME: 进程的执行时间，CMD: 该进程的名称或对应的路径。Wait [n]指令能够等待进程n完成并报告该进程的终止状态。Sleep n：command指令能够将指定指令挂起n秒，结束后执行指令。

五． 讨论

1. Linux系统命令很多，在手头资料不全时，如何查看命令格式？

答：使用man或help指令查看相关命令格式。

2. Linux系统用什么方式管理多个用户操作？如何管理用户文件，隔离用户空间？用命

令及结果举例说明。

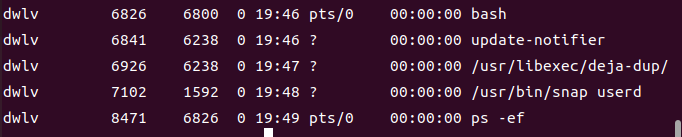
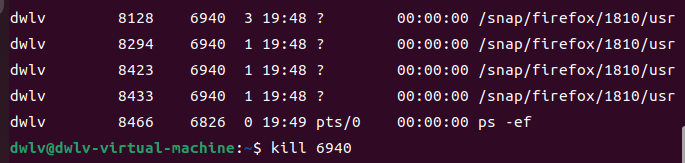
答：通过用户组的方式管理用户，通过给文件设定文件类型和权限以及所有者的方式管理用户文件，隔离用户空间。

例如：你用cat > mytext.txt指令创建了一个txt文本文件，然后你又使用chown newuser mytext.txt指令将文件的所有者修改为了newuser用户，这样，你便不能修改该文件了，因为该文件的所有者以及变成了newuser，你只有查看文件的权限。

3.用什么方式查看你的进程的管理参数？这些参数怎样体现父子关系？当结束一个父

进程后其子进程如何处理？用命令及结果举例说明。

答：用ps -ef查看所有金铖以及它的管理参数，ppid标明的数字是该进程的父进程的进程号，父子间存在一对多的关系。当结束一个父进程后，其所有的子进程都会被关闭。



该图表明：在19:48分时创建了进程firefox，产生了8128、8294等进程的子进程，当我使用kill指令将父进程6940关闭后，其所有的子进程都不见了。

1. Linux 系统“文件”的含义是什么？它的文件有几种类型？如何标识的？

答：Linux的文件是指数据的集合，文件一共有7种类型：普通文件、目录文件、符号连接、字符设备文件、块设备文件、命名管道FIFO、套接口socket。通过第一个字符来标识：-、d、b、c、s、p、l。

1. Linux 系统的可执行命令主要放在什么地方？找出你的计算机中所有存放系统的可执行命令的目录位置。

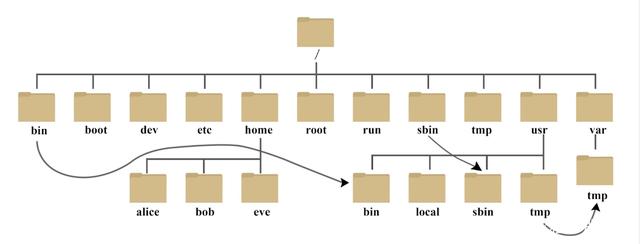
答：主要存放在bin目录下，我的计算机中存放系统的可执行命令的目录位置为：/bin、/user/bin。

1. Linux 系统得设备是如何管理的？在什么地方可以找到描述设备的信息？

答： linux中系统的设备都是以文件的形式存放的，管理文件就能管理相应设备。利用uname -a、head -n 1 /etc/issue、ispci -tv等指令便能够查看对应的设备信息。

7.画出Linux 根文件系统的框架结构。描述各目录的主要作用。你的用户主目录在哪

里？



/:是Linux系统的根目录

/bin:存放用户经常使用的命令

/boot:启动加载程序的静态文件

/dev:设备文件目录，不能单独分区

/etc:系统配置文件目录

/home:普通用户的家目录

/root:系统管理员的家目录

/run:进程的运行数据存放的目录

/sbin:存放系统管理员用户使用的命令

/tmp:临时文件目录；任何人都可以访问，存放周期10天

/usr:存放程序文件，库文件，共享文件，各种文档等

/var:动态数据文件目录；日志文件log，数据库，缓存目录等

/lib:存放程序的库文件

用户主目录在/root和/home中。

8.Linux 系统的Shell是什么？请查找这方面的资料，说明不同版本的Shell的特点。

1.ash

ash shell 是Knneth Almquist 編寫的，只包含24個內部命令，使用起來不是很方便，但是linux 中佔用系統資源最少的shell.

2.bash

bash 是linux 默認的Shell ，它由Brian Fox 和 Chet Ramey 共同完成，一共有40個內部命令。具有如下特點：

（1）可以使用上下方向鍵查閱和修改命令，類似於DOS裏doskey功能；

（2）查找自動匹配以某個字串開頭的命令；

（3）可以使用help命令。

3.ksh

ksh是 Korn Shell 的縮寫，共有42條內部命令。和商業版的 ksh 幾乎完全相容。

4.esh

esh 是linux比較大的內核，有52個內部命令。該shell 是指向/bin/tcsh 的shell,也即csh就是tcsh。

5.zch

zch 是linux最大的Shell之一，共有84個內部命令。一般用途，無需安裝該Shell。

9.下面每一项说明的是哪类文件。

(1）-rwxrw-r-- 第一位-表示普通文件。（如果是d表示文件夹也就是目录）， 后三位表示属主拥有rwx（可读、可写、可执行）权限， 再往后三位表示用户组拥有 r（可读)权限 ， 最后三位表示其它用户拥有 rw(可读、可写）权限。

(5）/etc/passwd 系统用户配置文件

(2) /bin 二进制文件

(6) crw-rw-rw 字符设备文件

(3) ttyx3 终端设备文件

(7) /usr/lib 目标库文件

(4) brw-rw-rw- 块设备(Block Device)文件

《计算机操作系统》实验二报告

姓名： 学号： 实验日期：

一、实验环境：Linux系统

二、实验目的：

**1.了掌握图形化用户界面和字符界面下使用 Shell 命令的方法。**

**2.掌握 ls、cd 等 Shell 命令的功能。**

**3.掌握重定向、管道、通配符、历史记录等的使用方法。**

**4.掌握手工启动图形化用户界面的设置方法。**

三、实验准备：

1.

1. 实验内容：

【操作要求 1】显示系统时间，并将系统时间修改为 2011 年 9 月 17 日零点。

【操作步骤】

（1） 启动计算机，以超级用户身份登录图形化用户界面。

（2） 依次单击顶部面板的「应用程序」菜单=>「附件」=>「终端」，打开桌面环境下的终端工具。

（3） 输入命令“date”，显示系统的当前日期和时间。

（4） 输入命令“date 091700002011”，屏幕显示新修改的系统时间。在桌面环境的终端中执行时显示中文提示信息。

实验结果：



成功修改系统时间为2011年9月17日。

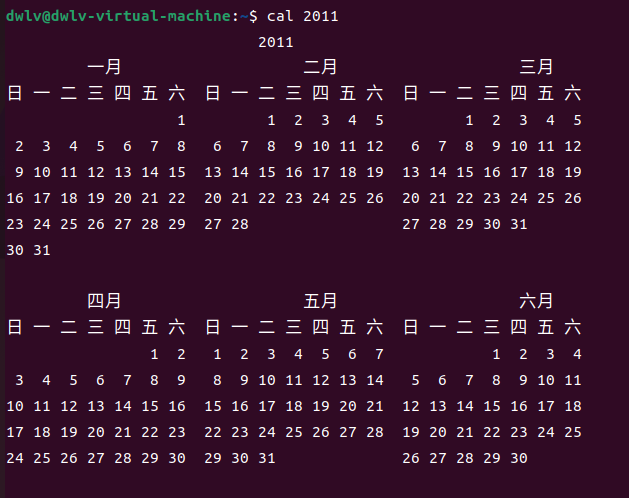
【操作要求 2】切换为普通用户，查看 2011 年 9 月 17 日是星期几。

【操作步骤】

（1） 前一操作是以超级用户身份进行的，但通常情况下只有在必须使用超级用户权限的时候，才以超级用户身份操作。为提高操作安全性，输入“su - helen”命令切换为普通用户 helen。

（2） 输入命令“cal 2011”，屏幕上显示出 2011 年的日历，由此可知 2011 年 9 月 17 日是星期日。

实验结果：



【操作要求 3】查看 ls 命令的-s 选项的帮助信息

【操作步骤】

方法一：

（1） 输入 “man ls” 命令，屏幕显示出手册页中 ls 命令相关帮助信息的第一页，介绍ls 命令的含义、语法结构以及-a、-A、-b 和-B 等选项的意义。

（2） 使用 PgDn 键、PgUp 键以及上、下方向键找到-s 选项的说明信息。

（3） 由此可知，ls 命令的-s 选项等同于--size 选项，以文件块为单位显示文件和目录的大小。

（4） 在屏幕上的“：”后输入“q”，退 出 ls 命令的手册页帮助信息。

方法二：

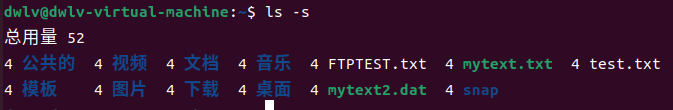
（1） 输入命令“ls --help”，屏幕显示中文的帮助信息。

（2） 拖动滚动条，找到-s 选项的说明信息，由此可知 ls 命令的-s 选项等同于--size 选项，以文件块为单位列出所有文件的大小，如图 2-3 所示。

（3） 在屏幕上的“：”后输入“q”，退 出 ls 命令的手册页帮助信息

**实验结果：**





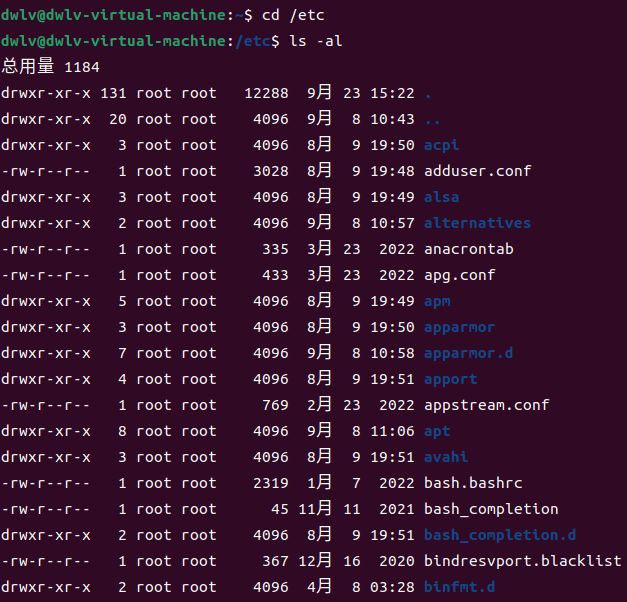
【操作要求 4】查看/etc 目录下所有文件和子目录的详细信息

【操作步骤】

（1） 输入命令“cd /etc”，切换到/etc 目录。

（2） 输入命令“ls -al”，显示/etc 目录下所有文件和子目录的详细信息。

实验结果：



【操作要求 1】查看当前目录。

【操作步骤】

（1） 启动计算机后默认会启动图形化用户界面，按下 CTRL+ALT+F1 键切换到第 1 个虚拟终端。

（2） 输入一个普通用户的用户名（helen）和口令，登录系统。

字符界面下输入口令时，屏幕上不会出现类似“

\*”的信息，提高了口令的安全性。

（3） 输入命令“pwd”，显示当前目录，相关操作参见如下内容。

Red Hat Enterprise Linux Server release 5 (Tikanga)

Kernel 2.6.18-8.el5 on an i386

localhost login : helen

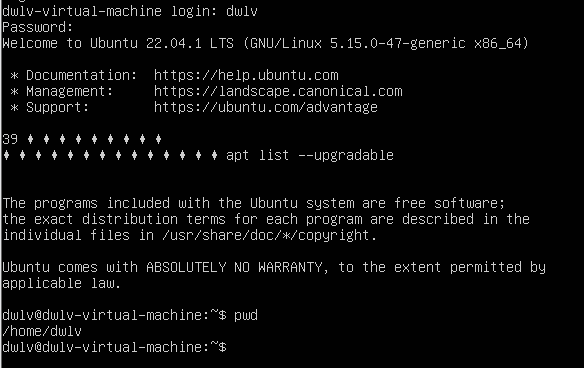
Password :

Last login: Tue Nov 20 09:28:42 on tty1

[helen@localhost ~]$ pwd

/home/helen

实验结果：



【操作要求 2】用 cat 命令在用户主目录下创建一名为 f1 的文本文件，内容为：

Linux is useful for us all.

You can never imgaine how great it is.

【操作步骤】

（1） 输入命令“cat >f1”，屏幕上输入点光标闪烁，依次输入上述内容。

使用 cat 命令进行输入时，不能使用左右上下方向键，只能用退格键（Backspace）来删除光标前一位置的字符。并且一旦按下回车键，该行输入的字符就不可修改。

（2） 上述内容输入后，按 Enter 键，让光标处于输入内容的下一行，按 CTRL+D 键结束输入。

（3） 要查看文件是否生成，输入命令“ls”即可。

（4） 输入命令“cat f1”，查 看 f1 文件的内容，相关操作参见如下内容。

[helen@localhost ~]$ cat >f1

Linux is useful for us all.

You can never imagine how great it is.

[helen@localhost ~]$ ls

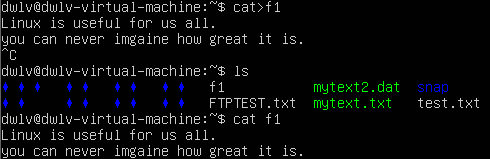
Desktop f1

[helen@localhost ~]$ cat f1

Linux is useful for us all.

You can never imagine how great it is.

实验结果：



【操作要求 3】向 f1 文件增加以下内容：Why not have a try?

【操作步骤】

（1） 输入命令“cat >>f1”，屏幕上输入点光标闪烁。

（2） 输入上述内容后，按 Enter 键，让光标处于输入内容的下一行，按 CTRL+D键结束输入。

（3） 输入“cat f1”命令，查看 f1 文件的内容，会发现 f1 文件增加了一行。

[helen@localhost ~]$ cat >>f1

Why not have a try?

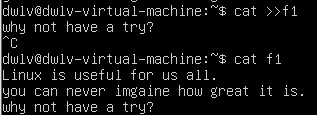
[helen@localhost ~]$ cat f1

Linux is useful for us all.

You can never imagine how great it is.

Why not have a try?

实验结果：



【操作要求 4】统计 f1 文件的行数，单词数和字符数，并将统计结果存放在 countf1 文件。

【操作步骤】

（1） 输入命令“wc <f1> countf1”，屏幕上不显示任何信息。

（2） 输入命令“cat countf1”，查看 countf1 文件的内容，其内容是 f1 文件的行数、单词数和字符数信息，即 f1 文件共有 3 行，19 个词和 87 个字符,相关操作参见如下内容。

[helen@localhost ~]$ wc <f1> countf1

[helen@localhost ~]$ cat countf1

3 19 87

实验结果：



【操作要求 5】将 f1 和 countf1 文件的合并为 f 文件

【操作步骤】

（1） 输入命令“cat f1 countf1 >f”，将两个文件合并为一个文件。

（2） 输入命令“cat f”，查看 f 文件的内容，如下所示。

[helen@localhost ~]$ cat f1 countf1 >f

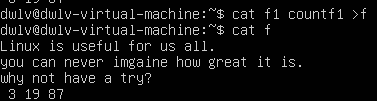
[helen@localhost ~]$ cat f

Linux is useful for us all.

You can never imagine how great it is.

Why not have a try?3 19 87

实验结果：



【操作要求 6】分页显示/etc 目录中所有文件和子目录的信息

【操作步骤】

（1） 输入命令“ls /etc|more”，屏幕显示出“ls /etc”命令输出结果的第一页，屏幕的最后一行上还出现“--More--”字样，按空格键可查看下一页信息，按 Enter 键可查看下 一行信息。

（2） 浏览过程中按“q”键，可结束分页显示。 管道符号“|”用于连接多个命令，前一命令的输出结果是后一命令的输入。

实验结果：



【操作要求 7】仅显示/etc 目录中前 5 个文件和子目录。

【操作步骤】

输入命令“ls /etc |head -n 5”, 屏幕显示出“ls /etc”命令输出结果的前面 5 行,相关操作参见如下内容。

[helen@localhost ~]$ ls /etc|head –n 5

a2ps.cfg

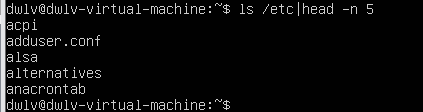
a2ps-site.cfg

acpi

adjtime

aliases

实验结果：



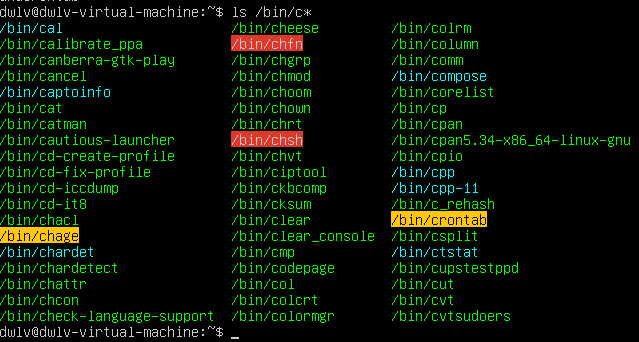
【操作要求 1】显示/bin/目录中所有以 c 为首字母的文件和目录

【操作步骤】

输入命令“ls /bin/c\*”，屏幕将显示/bin 目录中以 c 开头的所有文件和目录,

[helen@localhost ~]$ ls /bin/c\*

实验结果：



【操作要求 2】显示/bin/目录中所有以 c 为首字母,文件名只有 3 个字符的文件和目录

【操作步骤】

（1） 按向上方向键，Shell 命令提示符后出现上一步操作时输入的命令“ls /bin/c\*”。

（2） 将其修改为“ls /bin/c??”，按 下 Enter 键，屏幕显示/bin 目录中以 c 为首字母,文件名只有 3 个字符的文件和目录,相关操作参见如下内容。

[helen@localhost ~]$ ls /bin/c??

/bin/cat

/bin/csh

/bin/cut

实验结果：



【操作要求 3】显示/bin 目录中所有的首字母为 c 或 s 或 h 的文件和目录。

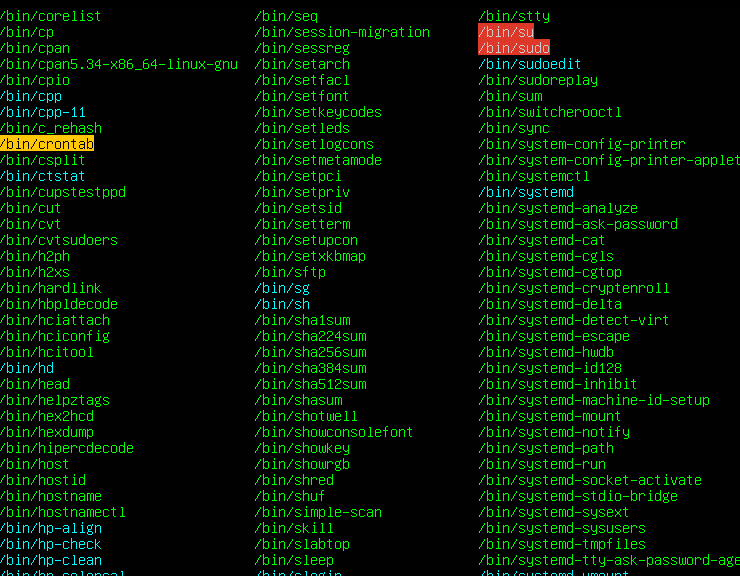
【操作步骤】

输入命令“ls /bin/[csh]\*”，屏幕显示/bin 目录中首字母为 c 或 s 或 h 的文件和目录,

相关操作参见如下内容。

[helen@localhost ~]$ ls /bin/[csh]\*

实验结果：



【操作要求 4】显示/bin/目录中所有的首字母是 v、w、x、y、z 的文件和目录。

【操作步骤】

输入命令“ls /bin/[!a-u]\*”，屏幕显示/bin 目录中首字母是 v~z 的文件和目录,

[helen@localhost ~]$ ls /bin/[!a-u]\*

/bin/vi /bin/view /bin/ypdomainname /bin/zcat





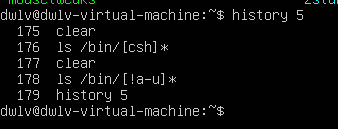
【操作要求 6】查看刚执行过的 5 个命令

【操作步骤】

输入命令“history 5”，显示最近执行过的 5 个命令,相关操作参见如下内容。命令编号可能不同。

[helen@localhost ~]$ history 5

实验结果：



【操作要求 7】修改开机启动字符界面

由于教材比较旧，因此书中的方法不适用，此处采取另一种方式进行设置。

《计算机操作系统》实验三报告

姓名： 学号： 实验日期：

一、实验环境：Linux系统

二、实验目的：

1.利用Linux提供的系统调用设计程序，加深对进程概念的理解。

2.体会系统进程调度的方法和效果。

3.了解进程之间的通信方式以及各种通信方式的使用。

三、实验准备：

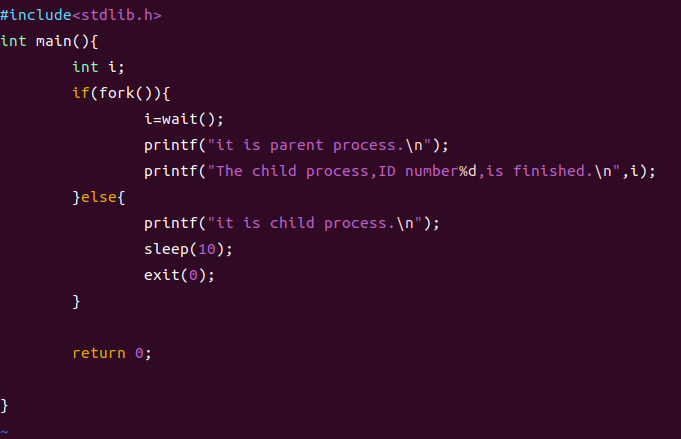
复习操作系统课程中有关进程、进程控制的概念以及进程通信等内容（包括软中断通信、管道、消息队列、共享内存通信及信号量概念）。熟悉本《实验指导》第五部分有关进程控制、进程通信的系统调用。它会引导你学会怎样掌握进程控制。阅读例程中的程序段。

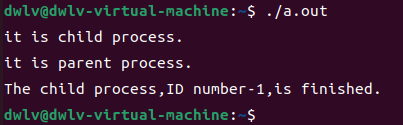
1. 实验内容：

【操作要求 1】

编写程序。实现父进程创建一个子进程。体会子进程与父进程分别获得不同返回值，进而执行不同的程序段的方法。

实验结果：





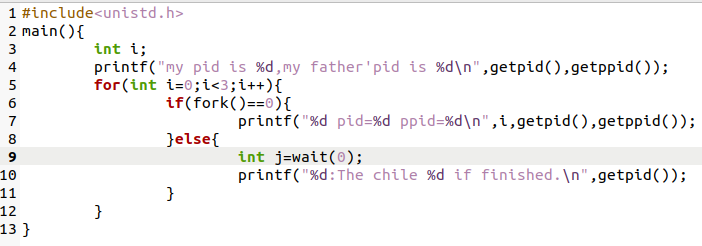
思考：子进程是如何产生的？ 又是如何结束的？子进程被创建后它的运行环境是怎

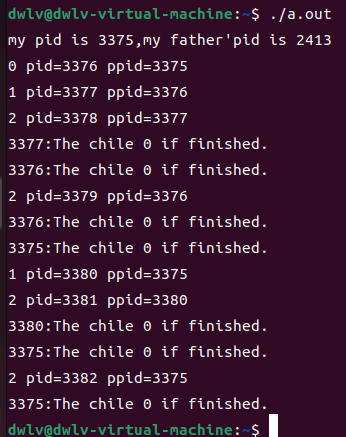
样建立的？

答：父进程使用fork()函数创建形成子进程，同时通过exit函数结束，子进程被创建后会被分配一个进程表项和进程标识符，间差同时运行的进程数目并拷贝进程表项的数据，由子进程继承父进程的所有文件。

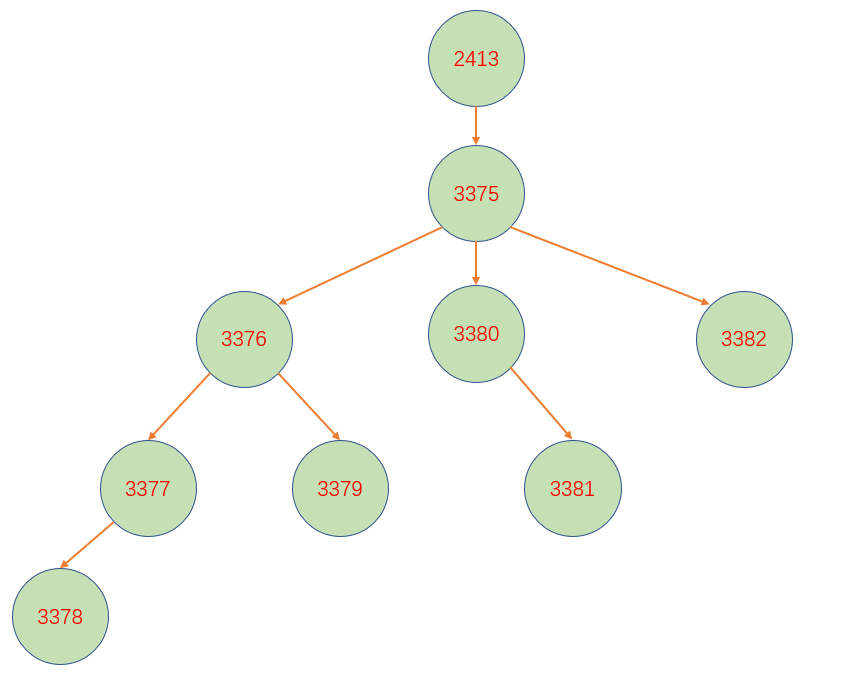
.【操作要求 2】

编写程序。父进程通过循环语句创建若干子进程。探讨进程的家族树以及子进程继承父进程的资源的关系。



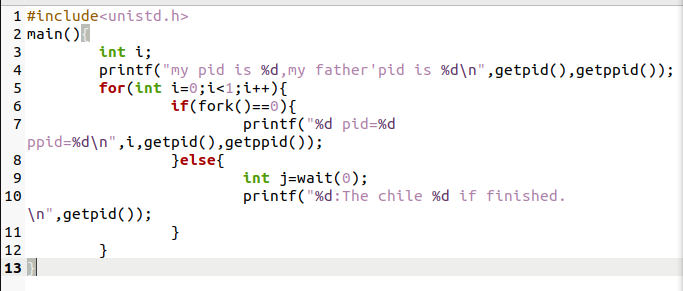


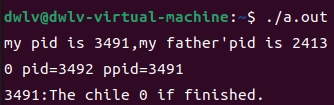
思考：① 画出进程的家族树。子进程的运行环境是怎样建立的？反复运行此程序看会有什么情况？解释一下。



反复运行程序会发现每次运行的家族树的形状都是相同的，但进程号每次都不一样，这是因为运行的时候系统是随机分配进程号，而且每次运行调用的进程都一样。

② 修改程序，使运行结果呈单分支结构，即每个父进程只产生一个子进程。画出进程树，解释该程序



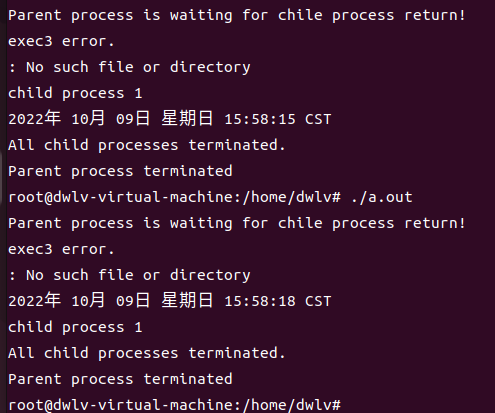




每个父进程只产生一个子进程，导致产生的家族树为链式结构。

.【操作要求 3】

参考例程3 编程，使用fork( )和exec( )等系统调用创建三个子进程。子进程分别启动不同程序，并结束。反复执行该程序，观察运行结果，结束的先后，看是否有不同次序。



思考：子进程运行其它程序后，进程运行环境怎样变化的？反复运行此程序看会有

什么情况？解释一下。

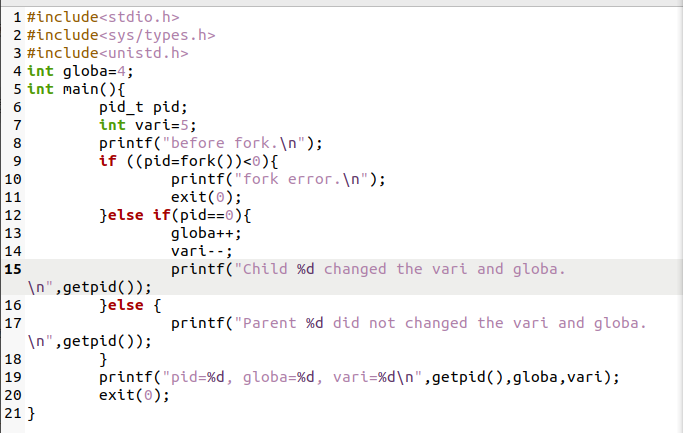
答：当子程序运行其他的程序后，其他程序就会覆盖子程序的进程，由于是覆盖，进程的标识号不会发生改变，但其代码段、数据段和栈段均发生了改变。

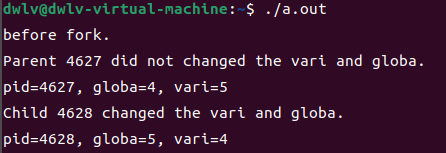
反复运行此程序可以发现，在运行过程中会有部分运行结果不同，例如上图的日期和child processes 1这两句便发生了错位。这是因为当每个子进程运行其他程序时，只有当其他程序结束后他才会结束，因此会有不同的结束时间和结束次序。

.

【操作要求 4】

参考例程4 编程，验证子进程继承父进程的程序、数据等资源。如用父、子进程修改公共变量和私有变量的处理结果；父、子进程的程序区和数据区的位置。



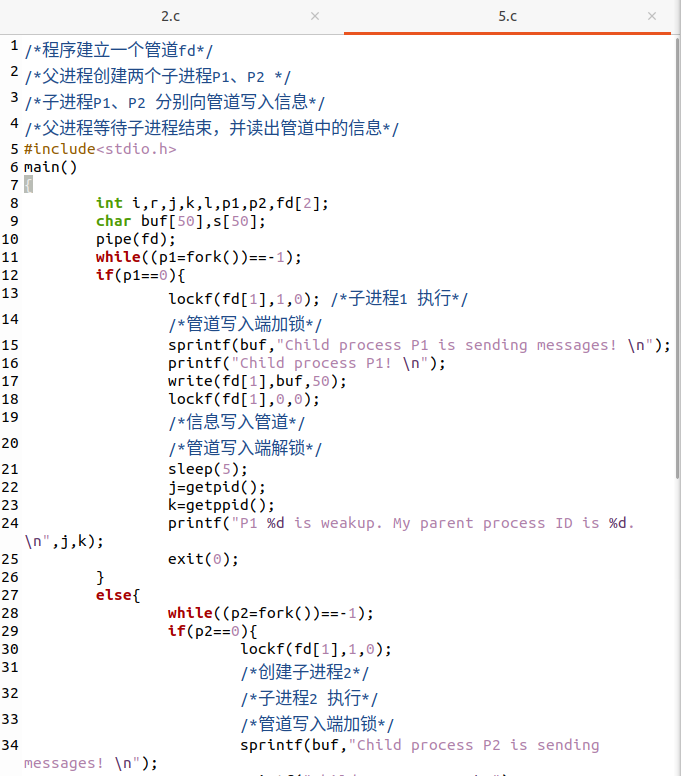


.思考：子进程被创建后，对父进程的运行环境有影响吗？解释一下。

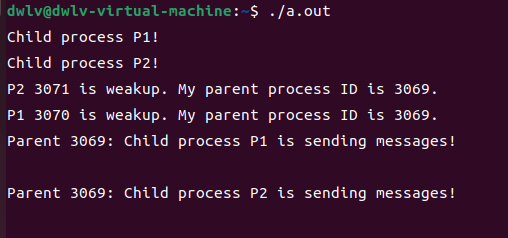
答：子进程被创建后对父进程的运行环境没有影响，从上图中可以看到，父进程虽然创建了子进程，但其全部变量和私有变量均未发生改变，子进程中的改变并没有影响到父进程，他们之间并不共享数据。

【操作要求 5】

参照《实验指导》第五部分中“管道操作的系统调用”。复习管道通信概念，参考例程5，编写一个程序。父进程创建两个子进程，父子进程之间利用管道进行通信。要求能显示父进程、子进程各自的信息，体现通信效果。





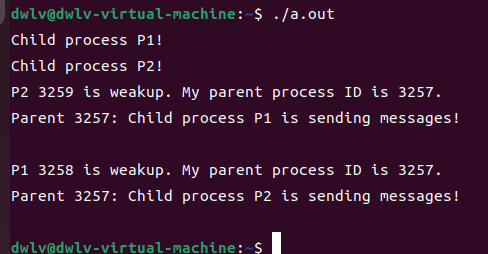


思考：①什么是管道？进程如何利用它进行通信的？解释一下实现方法。

答：管道本质上是一个开辟在内核中的内存空间，可以类比为共享文件，它能够使进程间实现通信，写进程从管道的输入端将数据写入管道，而读进程则从管道的输出端读出数据进行通信。

实现方法：一个管道由一个进程创建，然后该进程调用fork（），此后，父子进程便可以对该管道进行操作，从管道中读入读出数据，实现进程间的通话。

②修改睡眠时机、睡眠长度，看看会有什么变化。请解释。



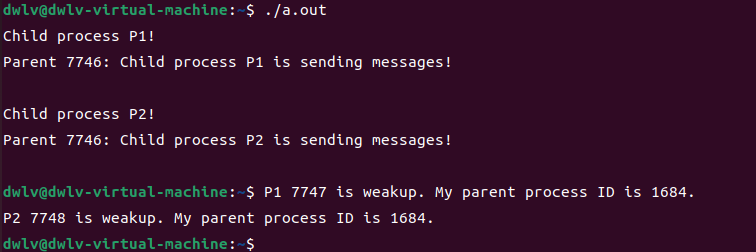
答：修改睡眠时间后，会引起进程被唤醒的时间发生变化，导致最后输出结果的顺序也发生变化。此处将进程1的睡眠时间稍微调长了1s，导致p1程序结束的时间延后，导致当p2结束后先唤醒p2，然后父进程利用wait（0）等待p2进程结束，输出接收到管道中长度为50的消息，利用wait（0）等待p1进程结束，输出接收到管道中长度为50的消息。

③加锁、解锁起什么作用？不用它行吗？

答：加锁解锁能够解决临界资源占用的问题，如果不加锁，共享资源fd就可能会同时被更改，出现问题。

【操作要求 6】

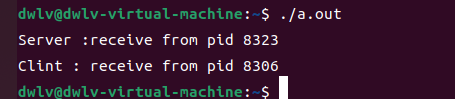
编程验证：实现父子进程通过管道进行通信。进一步编程，验证子进程结束，由父进程执行撤消进程的操作。测试父进程先于子进程结束时，系统如何处理“孤儿进程”的？



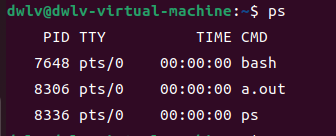
将操作5的代码中的父进程的wait()函数去除，则父进程会比子进程更快的结束，运行代码后发现，父进程的pid为7746，原本其子进程的ppid应为7746，但当其结束后孤儿进程的父进程却为1684，被系统的init进程所领养。

【操作要求 7】

编写两个程序一个是服务者程序，一个是客户程序。执行两个进程之间通过消息机制通信。消息标识MSGKEY 可用常量定义，以便双方都可以利用。客户将自己的进程标识（pid）通过消息机制发送给服务者进程。服务者进程收到消息后，将自己的进程号和父进程号发送给客户，然后返回。客户收到后显示服务者的pid 和ppid，结束。以下例程6 基本实现以上功能。这部分内容涉及《实验指导》第五部分中“IPC系统调用”。先熟悉一下，再调试程序。



服务器收到客户端发送的pid为8323，客户端收到服务器发送的消息（自己的pid）为8306。



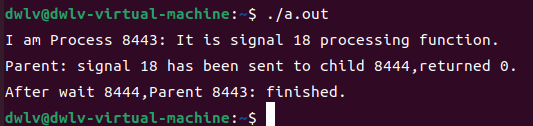
服务器进程号为8306，在后台运行。

思考：想一下服务者程序和客户程序的通信还有什么方法可以实现？解释一下你的设想。

答：还可以通过信号量机的方式实现，通过一个mutex来控制多个进程对一个共享资源的使用，每个进程使用就将mutex减一，使用完后将其加一，当mutex=1时当前进程能够使用。也就是在服务器发送消息的时候mutex减一，使得客户端不能发送信息，只有当服务器使用完毕后，客户端才能够使用。

【操作要求 8】

这部分内容涉及《实验指导》第五部分中“有关信号处理的系统调用”。编程实现软中断信号通信。父进程设定软中断信号处理程序，向子进程发软中断信号。子进程收到信号后执行相应处理程序。



思考：这就是软中断信号处理，有点儿明白了吧？讨论一下它与硬中断有什么区别？看来还挺管用，好好利用它。

答：硬中断是通过硬件执行的，而软终端是通过CPU的某条指令或软件对标志寄存器的某个标志位的设置而产生的的。

五. 研究并讨论

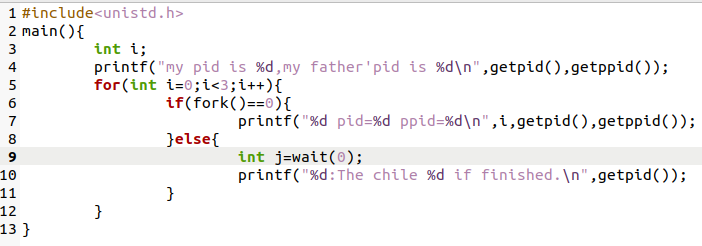
1. 讨论Linux 系统进程运行的机制和特点，系统通过什么来管理进程？

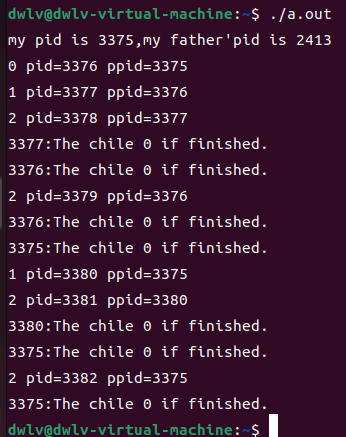
答：在linux中，每个进程在创建时都会被分配一个PCB，使得金铖从就绪态变为执行态。内核通过PID来标识每一个进程，能够循环使用。此外，每个进程都有一个独立的8k的堆栈空间和一个进程描述符结构。

Linux系统通过使用fork、wait等系统调用来实现对进程的管理。

1. C 语言中是如何使用Linux 提供的功能的？用程序及运行结果举例说明。

答：通过添加头文件的方式调用相关函数，通过getpid和getppid的方式来查看每个进程的进程号以及其父亲的进程号。





1. 什么是进程？如何产生的？举例说明。

答：进程是操作系统进行资源分配的最小单位。一个进程是一个程序的一次执行过程。每启动一个进程，操作系统就会为它分配一块独立的[内存](https://so.csdn.net/so/search?q=%E5%86%85%E5%AD%98&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/m0_50786325/article/details/_blank)空间，用于存储PCB、数据段、程序段等资源。每个进程占有一块独立的内存空间。

进程由fork函数调用所产生，通过申请PCB，分配资源而产生。

举例：分时系统中的用户登录

1. 进程控制如何实现的？举例说明。

通过原语实现。

例如往某个进程进入堵塞队列时，通过wakeup()原语来唤醒进程，使之进入到就绪队列中去。

1. 进程通信方式各有什么特点？用程序及运行结果举例说明。
2. 管道

特点：分为全双工和半双工，在同一时间只能进行读或写操作中的其中之一。

2.信号

特点:

1.简单2.不能携带大量信息3.满足某个特定的条件发送4.优先级高

3.信号量

主要用于进程间以及同一进程不同线程之间的同步手段

特点

1.本质是一个计数器，内存中有多少个临界资源，信号量的数字就是多少

2.信号量基于操作系统的pv操作，程序对信号量的操作都是原子操作

3.信号量用于进程间同步，如果要进程间传递数据需要结合共享内存

4.共享内存

共享内存是效率最高的通信方式，允许两个或者多个进程共享物理内存的同一块区域.由于一个共享内存段会成为一个进程用户空间的一部分，所以这种机制无需内核介入

5.消息队列

解耦:只要保证消息格式不变，发送发和接收方可以不受对方的影响

异步:非核心流程异步化，提高系统响应性能。

削峰:限制用户数量，将接受的用户请求写入消息队列中，然后消息队列长度超过最大数量，直接抛弃用户请求或跳转到错误节目

6.Socket通信

网络中不同主机上的应用进程之间双向通信的端口的抽象。Linux下本质为内核借助缓冲区形成的伪文件。可以用文件描述符来引用套接字。

管道的程序运行结果如实验操作5所示，此前已实现过。

1. 管道通信如何实现？该通信方式可以用在何处？

通过写进程向管道内义字符的形式写入数据，后通过读进程从管道种读取数据。

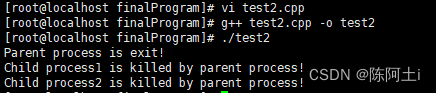
可以用于数据传输、资源共享和通知。

1. 什么是软中断？软中断信号通信如何实现？

“软中断是对硬中断的一种模拟,发送软中断就是向接收进程的 proc 构造中的相应项发送一个特定意义的信号 ，从而实现软件上的模拟“中断”。软中断必须等到接收进程执行时才能生效 。

如何实现：通过signal和kill来实现。

使用系统调用fork()创建两个子进程，再用系统调用signal()让父进程捕捉用函数设置时钟的时间段终止时产生的信号，当捕捉到中断信号后，父进程用系统调用kill()向两个进程发出信号，子进程捕捉到信号后，分别输出子进程被杀死信息后终止： 父进程等待两个子进程终止后，输出父进程被杀死信息终止。



分析：当父进程收到时间中断信号后，执行killChildren（）函数，分别杀死子进程1和子进程2（发送信号SIGUSR1和SIGUSR2），然后输出“Parent process is exit!”，父进程执行结束，随后，由进程调度程序执行子进程1，子进程1收到SIGUSR1信号，执行killProcess1（）函数，输出“Child process1 is killed by parent process!”，子进程1执行完毕，然后执行子进程2，子进程2收到SIGUSR2信号，执行killProcess2（）函数，输出“Child process2 is killed by parent process!”，子进程2执行完毕。

《计算机操作系统》实验四报告

姓名： 学号： 实验日期：

一、实验环境：Linux系统

二、实验目的：

1.体会熟练掌握手工启动前后台作业的方法。

2.熟练掌握进程与作业管理的相关 Shell 命令。

3.掌握 at 调度和 cron 调度的设置方法。

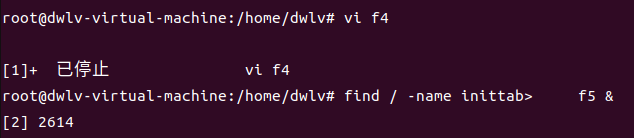
4.了解进行系统性能监视的基本方法。

三、实验内容：

【操作要求 1】

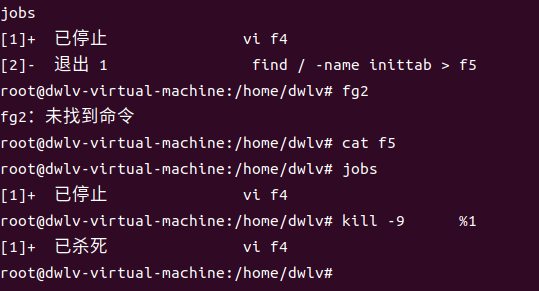
先在前台启动 vi 编辑器并打开 f4 文件，然后挂起，最后在后台启动一个查找 inittab 文件的 find 作业，find 的查找结果保存到 f5。

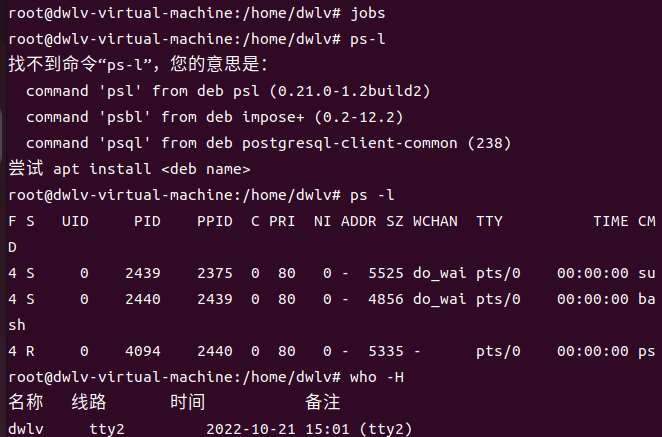
实验结果：



【操作要求 2】

查看当前作业、进程和用户信息，并对作业进行前后台切换。

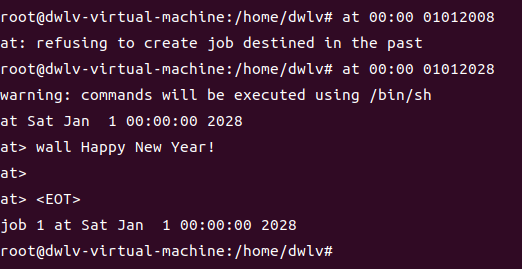




【操作要求 3】

设置一个调度，要求在 2008 年 1 月 1 日 0 时，向所有用户发送新年快乐的问候

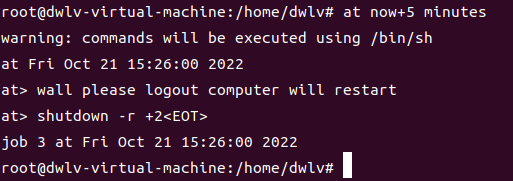
实验结果：（由于不能够在过去设置时间，这里这只为在2028年进行发送）



【操作要求 4】

设置一个调度，要求 5 分钟后向所有用户发送系统即将重启的消息，并在 2分钟后重新启动计算机

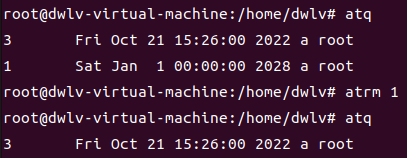
实验结果：



【操作要求 5】

查看所有的 at 调度，并删除 08 年 1 月 1 日执行的调度任务（这里删除2028年的调度任务）

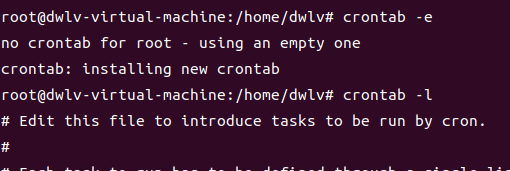
实验结果：

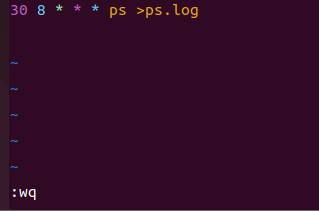


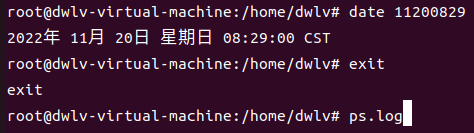
删除了作业号为1的任务

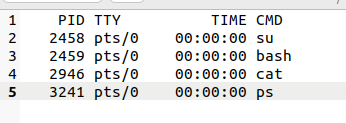
【操作要求 6】

helen 用户设置 crontab 调度，要求每天上午 8 点 30 份查看系统的进程状态，并将查看结果保存于 ps.log 文件。







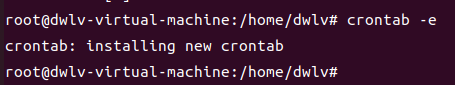


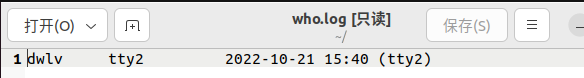
【操作要求 7】

helen 用户添加设置 crontab 调度，要求每三个月的 1 号零时查看正在使用的用户列表

实验结果：



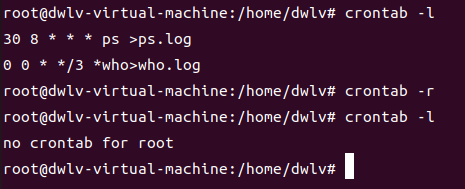




【操作要求 8】

查看 cron 调度内容，最后删除此调度。

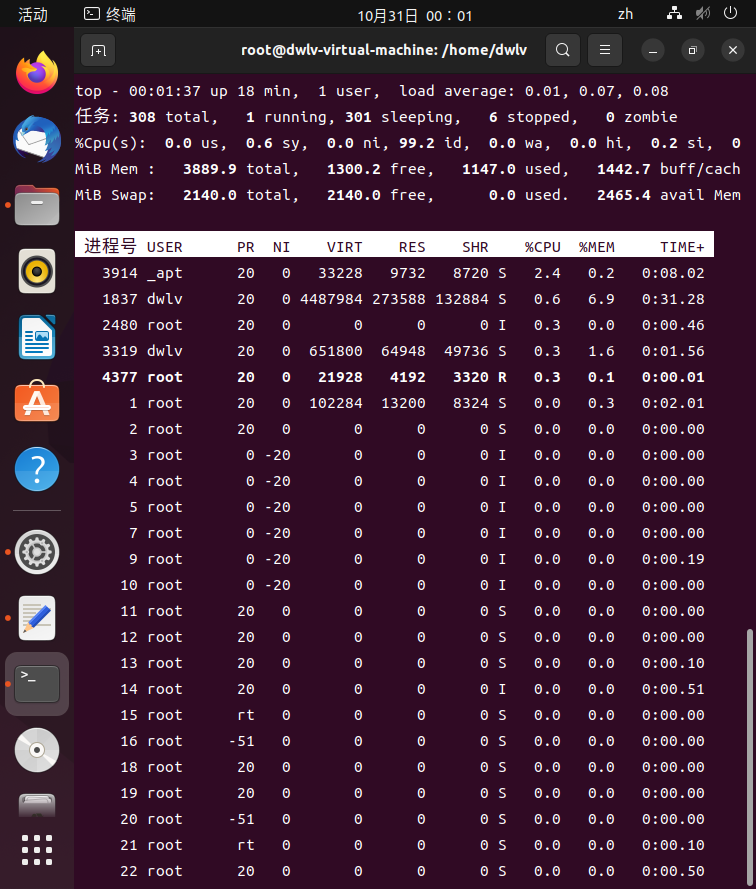
实验结果：



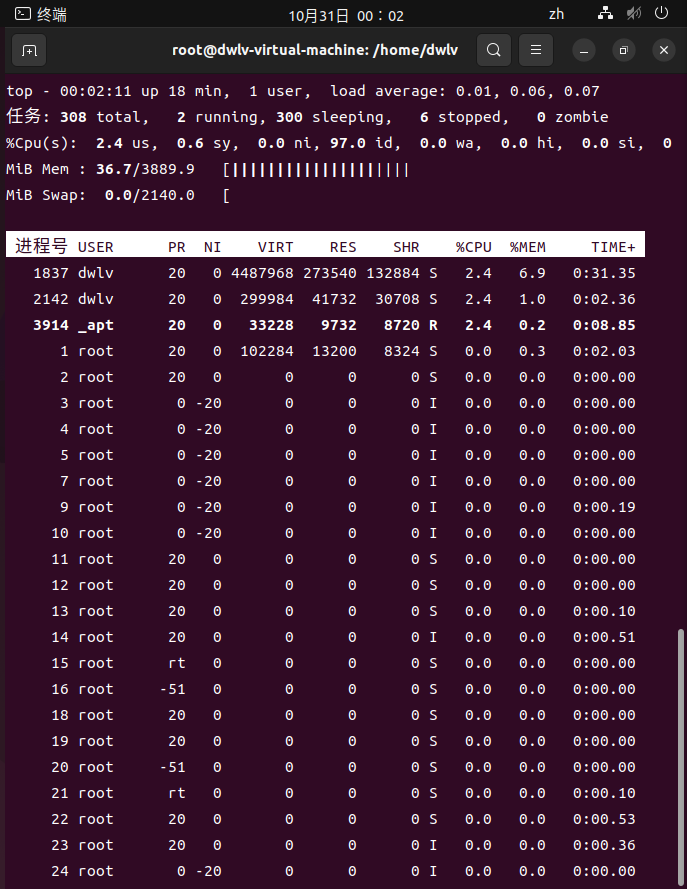
【操作要求 9】

利用 Shell 命令监视系统性能

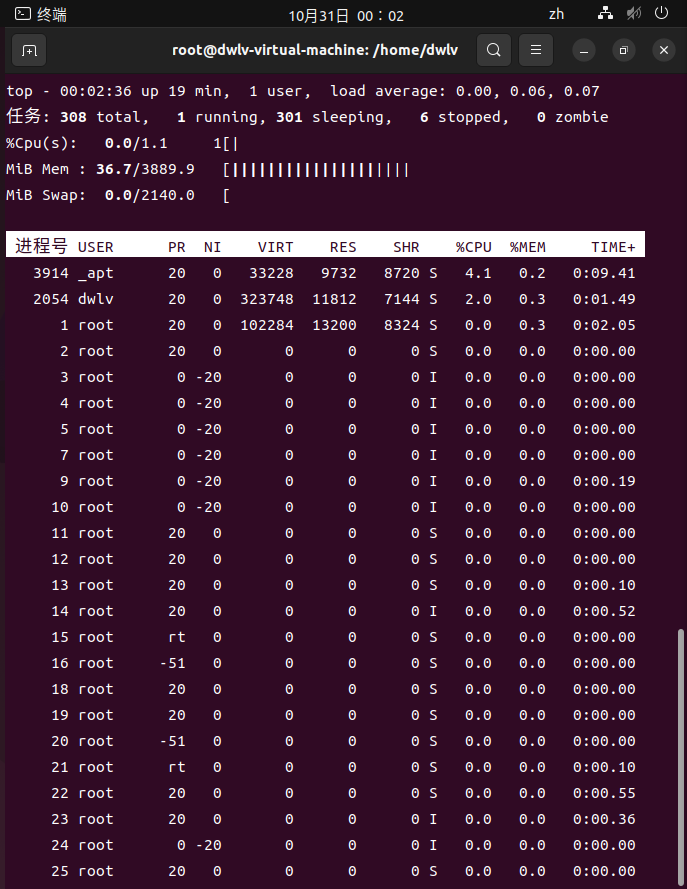
实验结果：



**按下m键后按照资源使用率进行排列**



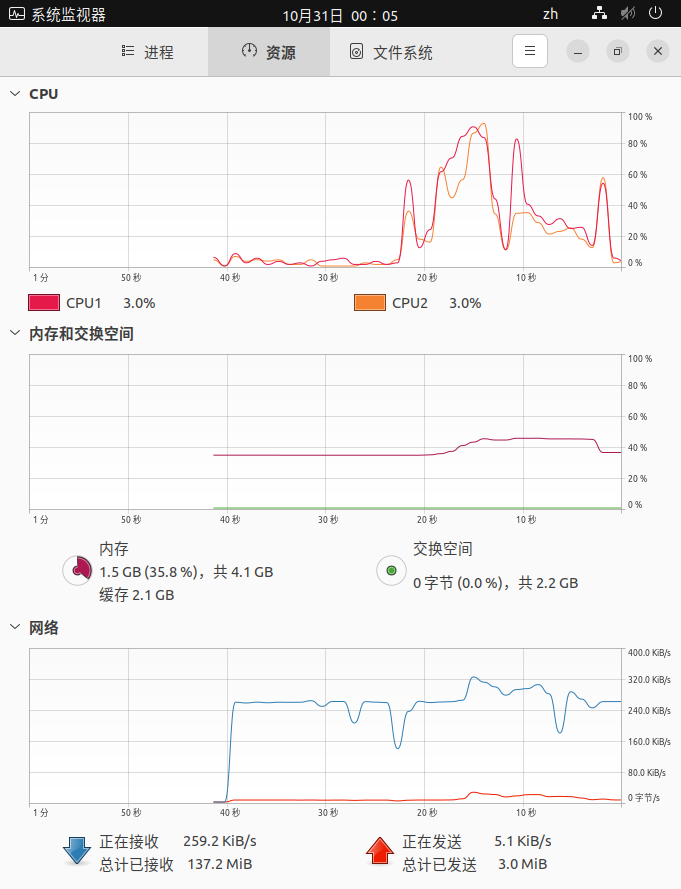
按下T键按照时间排列



【操作要求 10】

利用「系统监视器」工具监视 CPU 使用情况。

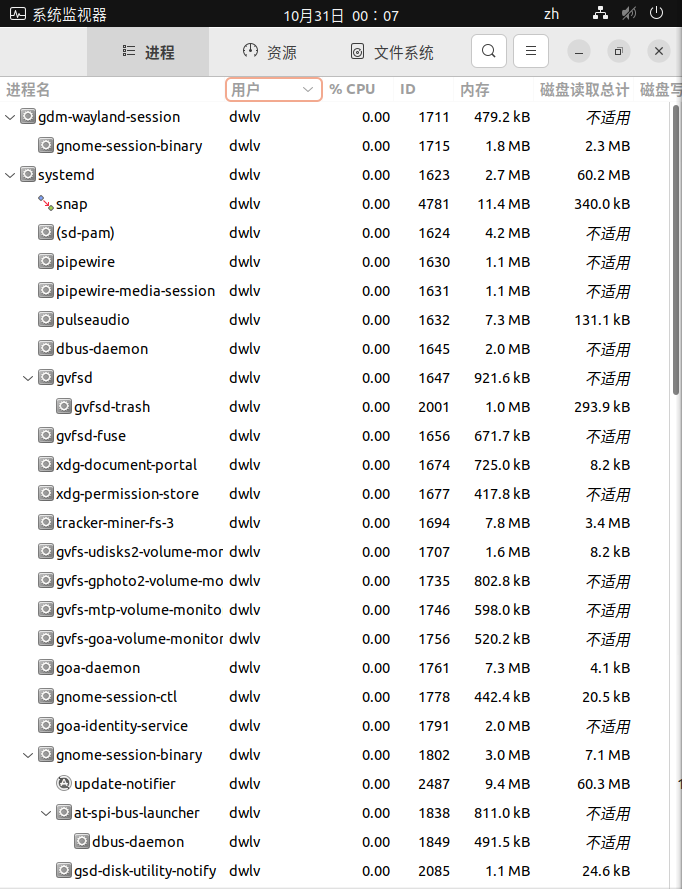
实验结果：

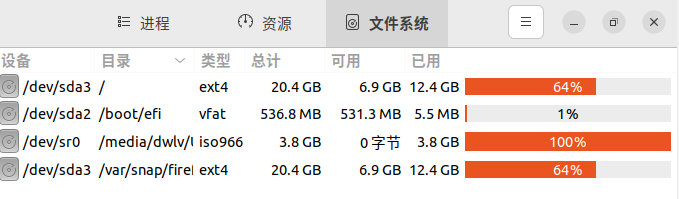


【操作要求 10】

利用「系统监视器」查看当前所有的进程，要求显示出启动进程的用户。

实验结果：

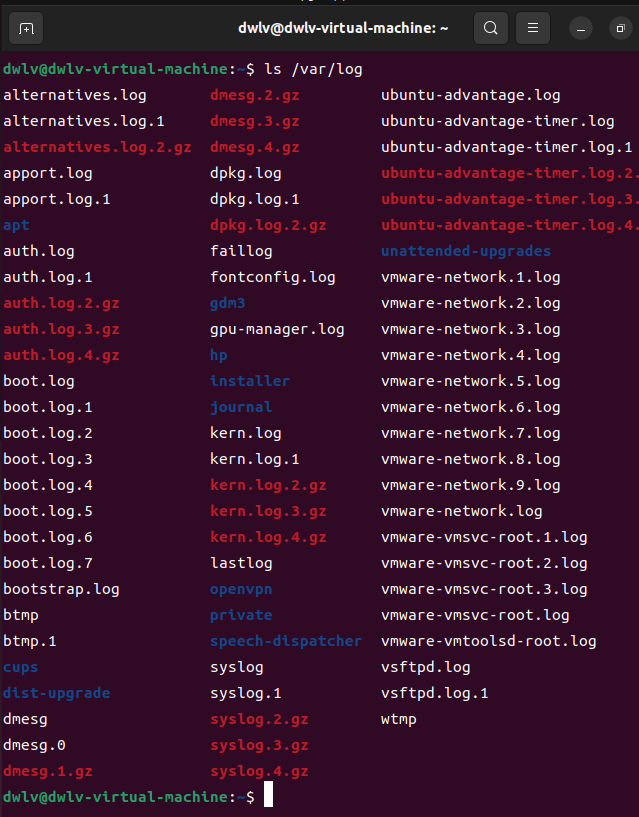




【操作要求 11】

利用「系统日志」工具查看系统日志

实验结果：





《计算机操作系统》实验五报告

姓名： 学号： 实验日期：

一、实验环境：Linux系统

二、实验目的：

1、理解/etc/passwd 和/etc/group 文件的含义。

2、掌握桌面环境下管理用户与组群的方法。

3、掌握利用 Shell 命令管理用户与组群的方法。

4、掌握批量新建用户帐号的步骤和方法。

三、实验内容：

【操作要求 1】

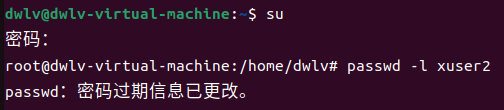
新建两个用户帐号，其用户名为 xuser1 和 xuser2，口令为“e12ut59er”和

“wfu1t28er”。



【操作要求 2】

锁定 xuser2 用户帐号





【操作要求 3】

删除 xuser2 用户



【操作要求 4】

新建两个组群，分别是 myusers 和 temp。

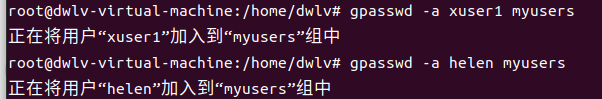
实验结果：



【操作要求 5】

修改 myusers 组群属性，将 xuser1 和 helen 用户加入 myusers 组群。

实验结果：



【操作要求 6】

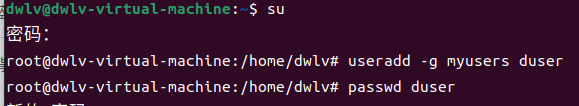
删除 temp 组群

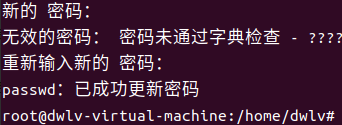


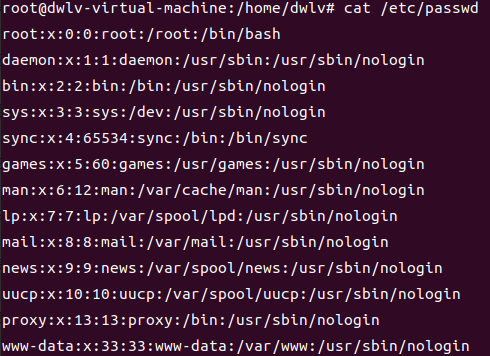
【操作要求 7】

新建一名为 duser 的用户，其口令是“tdd63u2”，主要组群为 myusers。

实验结果：



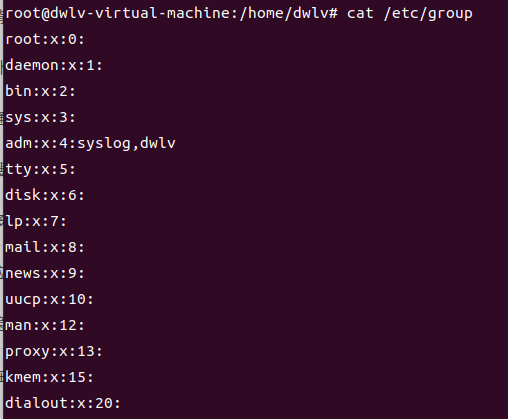




发现存在新的信息数据



查看组群

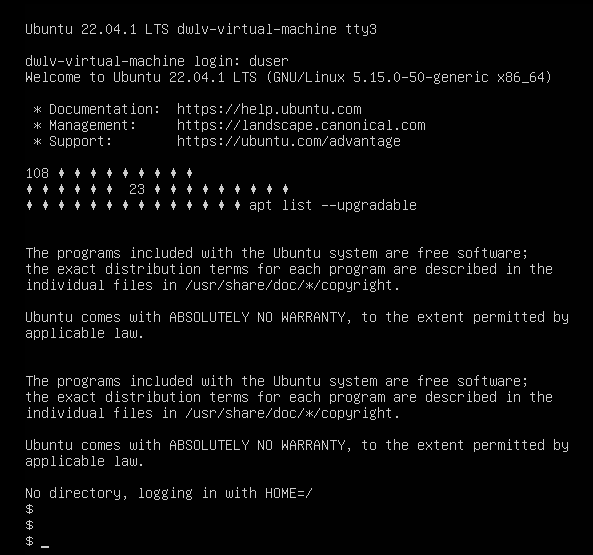


【操作要求 8】

将 duser 用户设置为不需口令就能登录。

实验结果：





【操作要求 9】

查看 duser 用户的相关信息

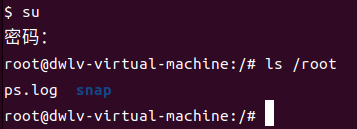
实验结果：



【操作要求 10】

从普通用户 duser 切换为超级用户

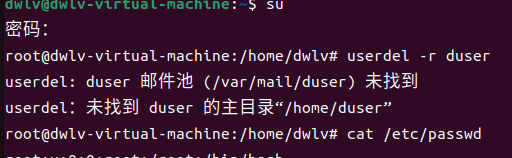
实验结果：



【操作要求 10】

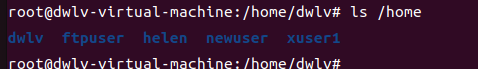
一次性删除 duser 用户及其工作目录

实验结果：





Passwd文件中无duser用户密码



Home文件下也没有duser用户

【操作要求 11】

将 mygroup 组群改名为 newgroup

实验结果：





【操作要求 12】

删除 newgroup 组群

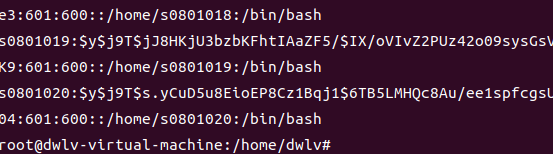
实验结果：



【操作要求 13】

为全班同学 20 位同学创建用户帐号，用户名为“s”+学号的组合，其中班级名册中第一位同学的学号为 080101。所有同学都属于 class0801 组群。所有同学的初始口令为 111111。

实验结果：



**成功创建20个学生账号**

《计算机操作系统》实验六报告

姓名： 学号： 实验日期：

一、实验环境：Linux系统

二、实验目的：

1、掌握vi 的三种工作方式，熟悉vi 编辑程序的使用。

2、学习Shell 程序设计方法。掌握编程要领。

三、实验内容：

【操作要求 1】

按本《实验指导》第三部分的内容。熟悉vi 的三种工作方式。熟悉使用各种编辑功能

思考：试一试vi 的三种工作方式各用在何时？用什么命令进入插入方式？怎样退出插

入方式？文件怎样存盘？注意存盘后的提示信息。

答：

1、一般指令模式是在刚进入vi时的默认模式，其他模式时通过按下ESC键即可切换到该模式。

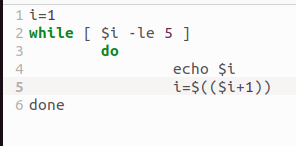
【1】控制光标移动：↑，↓，j  
 【2】删除当前行：dd  
 【3】查找：/字符  
 【4】进入编辑模式：i o a  
 【5】进入底行模式：:  
 【6】撤销操作 u

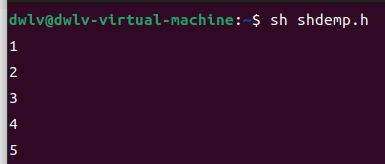
1. 编辑模式通过在一般命令模式下按下i键进入，可以移动光标后在指定位置进行插入和删除
2. 命令行模式通过在一般命令模式下输入“：”按钮进入，该模式下能够进行搜索、保存、离开等操作。

在一般模式下按下i键即可进入插入模式，按下ESC键即可退出插入模式，在一般模式下按下“：”键并输入对应指令如（:wq）即可存盘。

【操作要求 2】

创建和执行Shell 程序

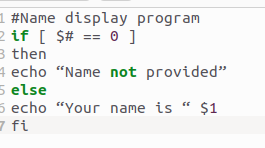


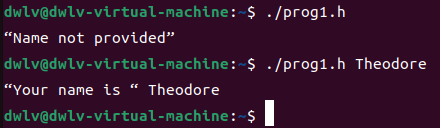


【操作要求 3】

用vi 编写《实验指导》“第四部分Shell 程序设计”中的例1（假设文件名为prog1.h），

练习内部变量和位置参数的用法。

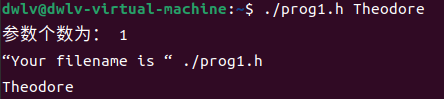




【操作要求 4】

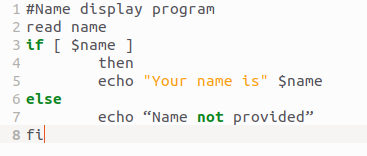
进一步修改程序prog1.h，要求显示参数个数、程序名字，并逐个显示参数。

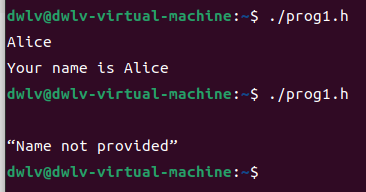




【操作要求 5】

修改例1程序（即上面的 prog1.h），用read命令接受键盘输入。若没有输入显示第一种提示，否则第二种提示。

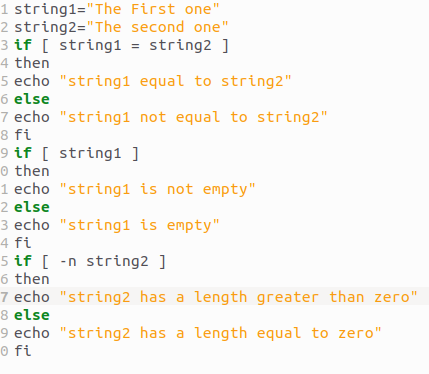


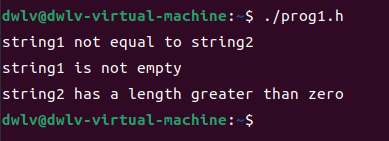


【操作要求 6】

用vi 编写《实验指导》“第四部分 Shell 程序设计”中的例2、例3，练习字符串比较运算符、数据比较运算符和文件运算符的用法，观察运行结果。

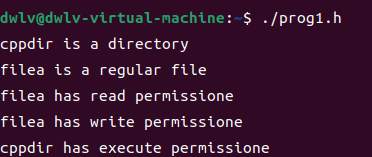
例二：





例三：

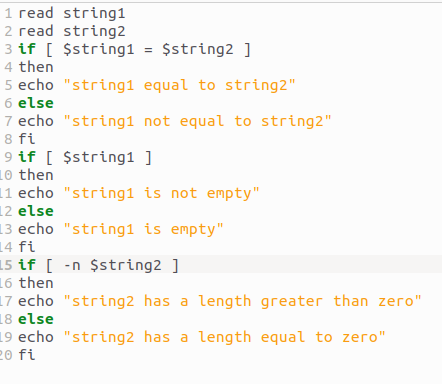


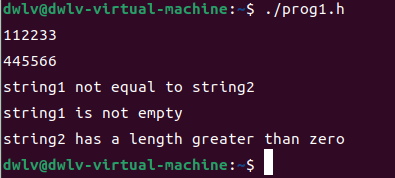


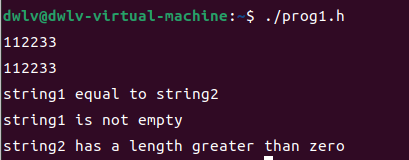
【操作要求 7】

修改例2程序，使在程序运行中能随机输入字符串，然后进行字符串比较。

实验结果：



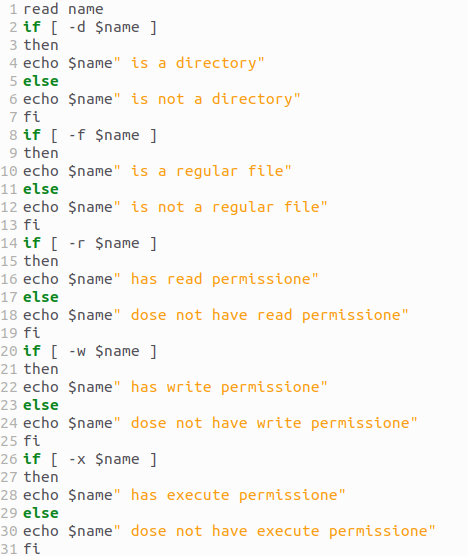


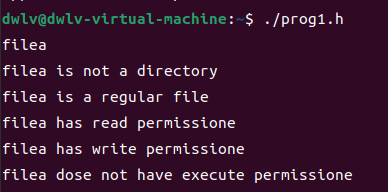


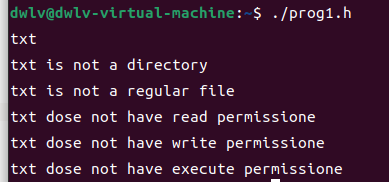
【操作要求 8】

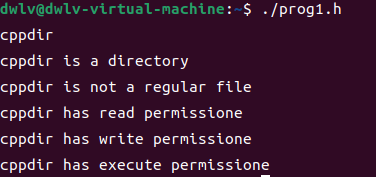
修改例3程序，使在程序运行中能随机输入文件名，然后进行文件属性判断。

实验结果：







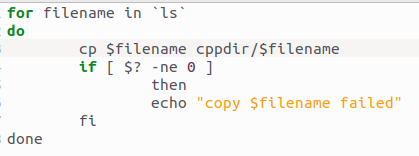


【操作要求 9】

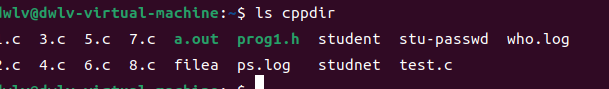
用vi 编写《实验指导》“第四部分 Shell 程序设计”中的例4、例5、例6、例7，掌握控制语句的用法，观察运行结果。

实验结果：

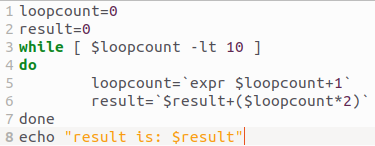
例4：





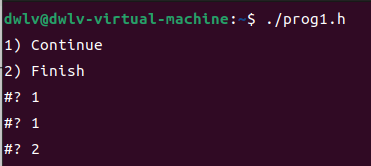
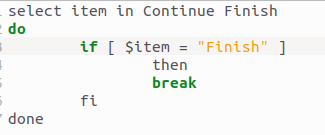


例5：





例6：

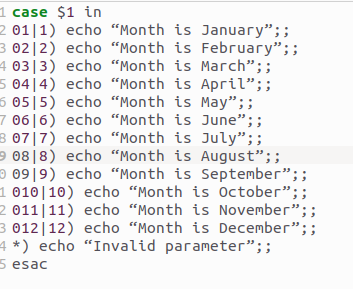


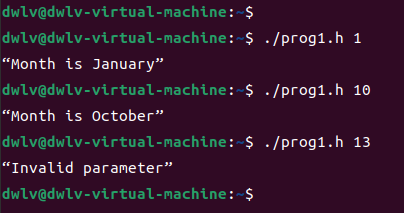
【操作要求 10】

用vi 编写《实验指导》“第四部分 Shell 程序设计”中的例8 及例9掌握条件语句的用法，函数的用法，观察运行结果。

实验结果：

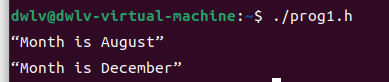
例8：





例9：



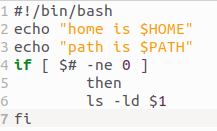


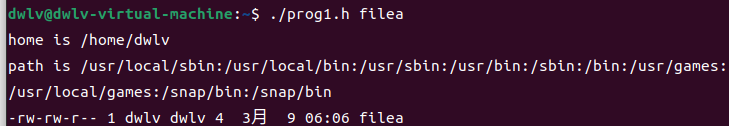
【操作要求 10】

编程，在屏幕上显示用户主目录名（HOME）、命令搜索路径（PATH），并显示由位

置参数指定的文件的类型和操作权限。

实验结果：





## 讨论：

1. Linux 的Shell 有什么特点？

答：他有循环结构以及分支结构，与其他的应用程序具有相同的效果，是系统的用户界面，提供了用户与内核进行交互操作的一个接口。

1. 怎样进行Shell编程？如何运行？有什么条件？

答：shell程序是一个包含了若干行linux命令的文件，和其他的高级程序语言相同，编写一个shell程序需要一个文本编辑器如vi或是vim，根据shell的语法规则，利用chmod的命令即可将其设置为可执行模式，输入文件名即可运行。

3. 编写一个具有以下功能的Shell程序。

(1) 把当前目录下的文件目录信息输出到文件 filedir.txt 中；







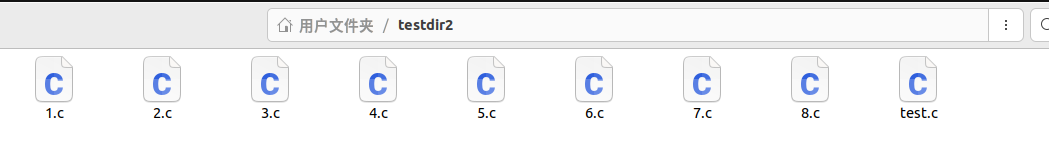
1. 在当前目录下建立一个子目录，目录名为 testdir2 ；





1. 把当前目录下的所有扩展名为 c 的文件以原文件名复制到子目录testdir2中；





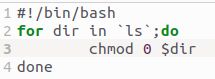
(4) 把子目录中的所有文件的存取权限改为不可读。（提示：用 for 循环控制语句实

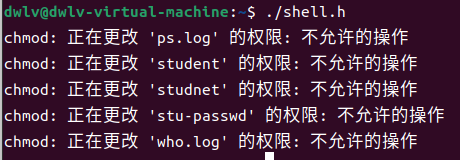
现，循环的控制列表用 ’ls’ 产生。）

原文件的权限



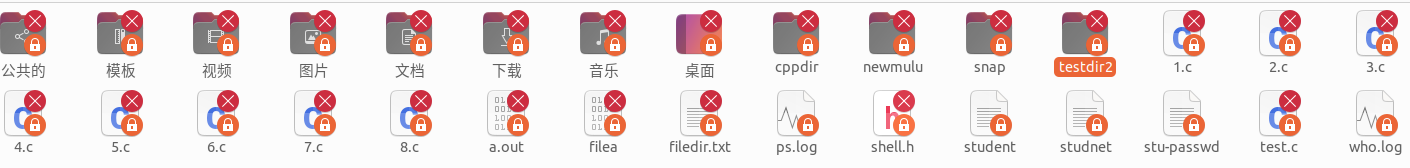
Shell文件中的代码





除了非该用户的文件，所有文件变成不可读不可写不可执行





（5）把你的用户信息追加到文件 filedir.txt 中；



