**中南林业科技大学**



**实验报告**

课程名称： Linux系统及应用实验

姓 名： 刘军 学 号： 20212753

专业班级： 2021计算机科学与技术 3 班

指导老师： 李洁

学 院： 计算机与信息工程学院

|  |  |
| --- | --- |
| **评分项** | **评分** |
| 报告封面、目录（4分）、名称正确无误（8分）、是否按时完成（10分）、版面整洁、条理清晰（8分） |  |
| 目的清晰明确（10分）、内容及步骤完整（30分） |  |
| 总结（30分） |  |
| **总分 100分**  **（抄袭则直接0分）** |  |

# 实验一 Linux操作系统入门

## 实验目的

1. 掌握Linux操作系统的安装

2. 熟悉Linux桌面环境、图形界面基本操作及常用软件

3. 掌握系统界面切换方法

4. 熟悉Linux系统在线帮助工具的使用

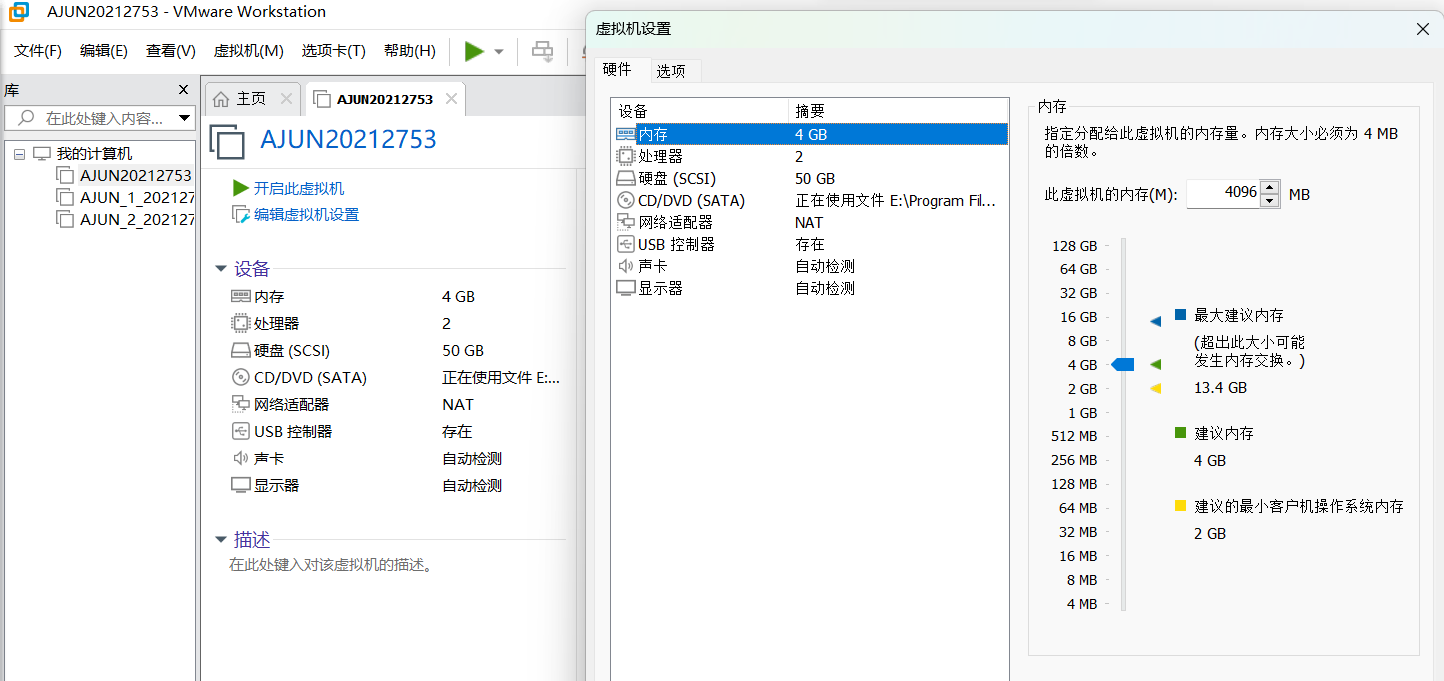
## 实验准备（仅实验一有此项）

（1）准备Ubuntu ISO镜像文件（也可自行下载其他linux系统镜像文件）。例如Ubuntu21.04-desktop（下载地址http://releases.ubuntu.com/21.04/）。

（2）安装VMware Workstation或者Virtualbox软件，创建一台Linux虚拟机。内存建议2GB以上，硬盘和网络模式等其他选项都可以选择默认值。

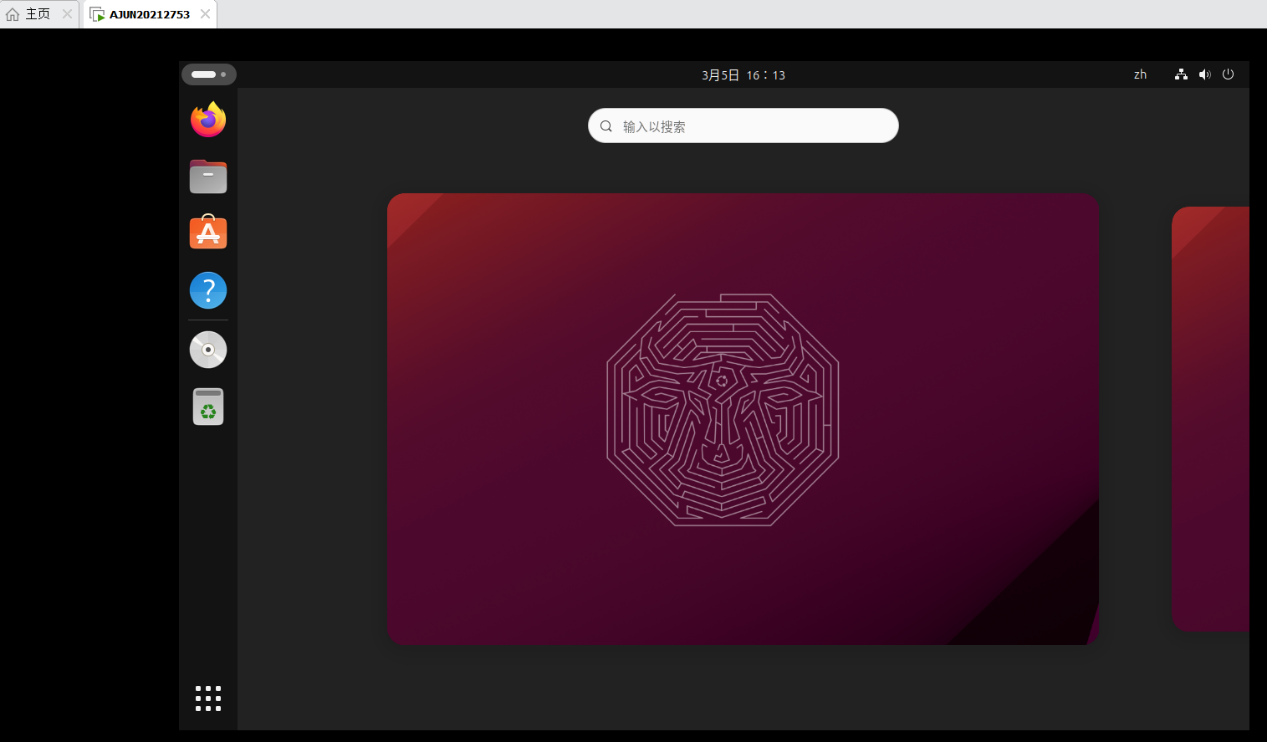
## 实验内容

1. 安装Linux操作系统

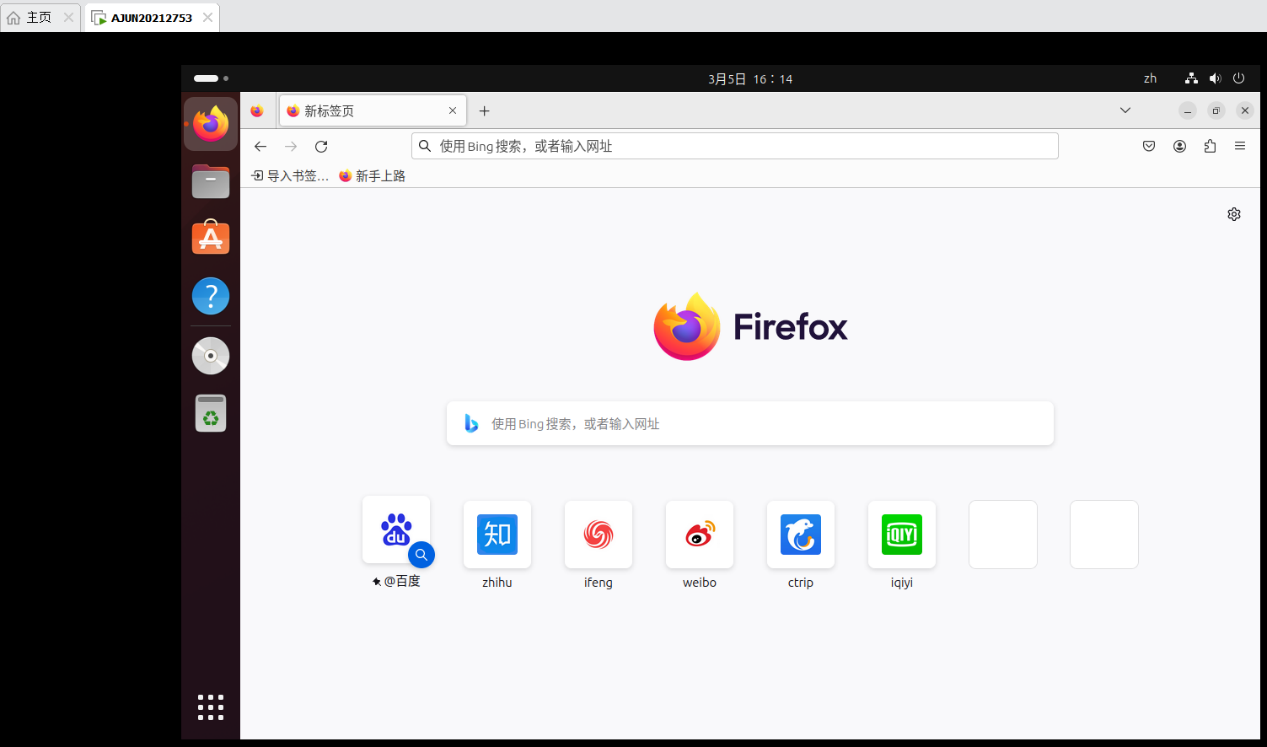


2. 熟悉Linux桌面环境、图形界面基本操作及常用软件

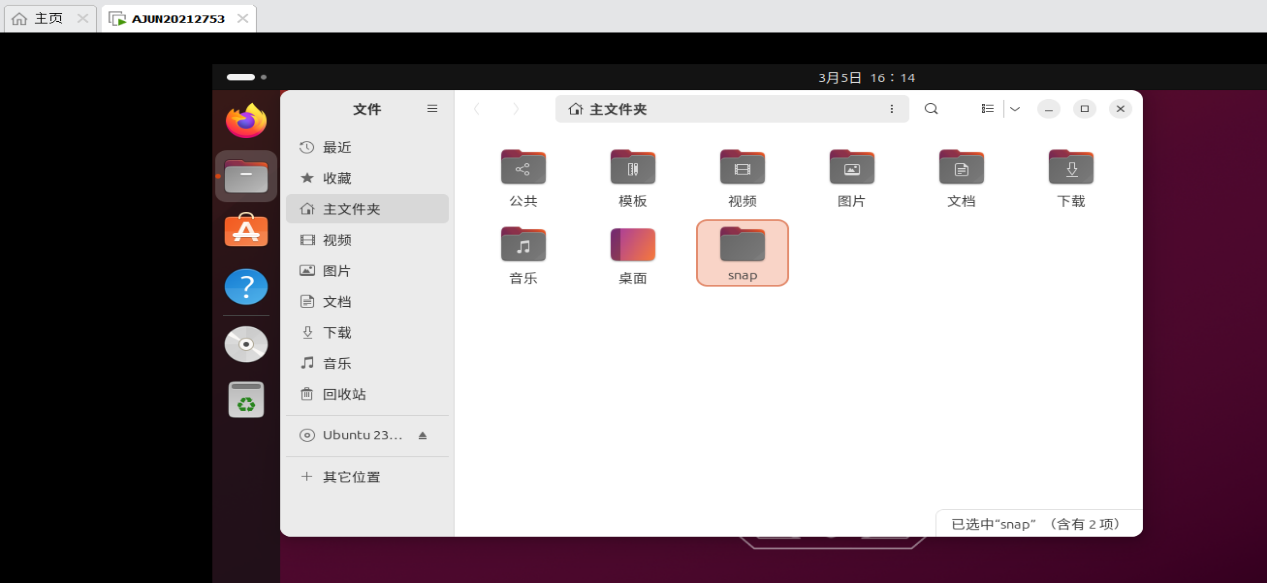
（1）使用活动概览视图。屏幕左上角“活动”（activities）按钮，或者键盘上的windows键。



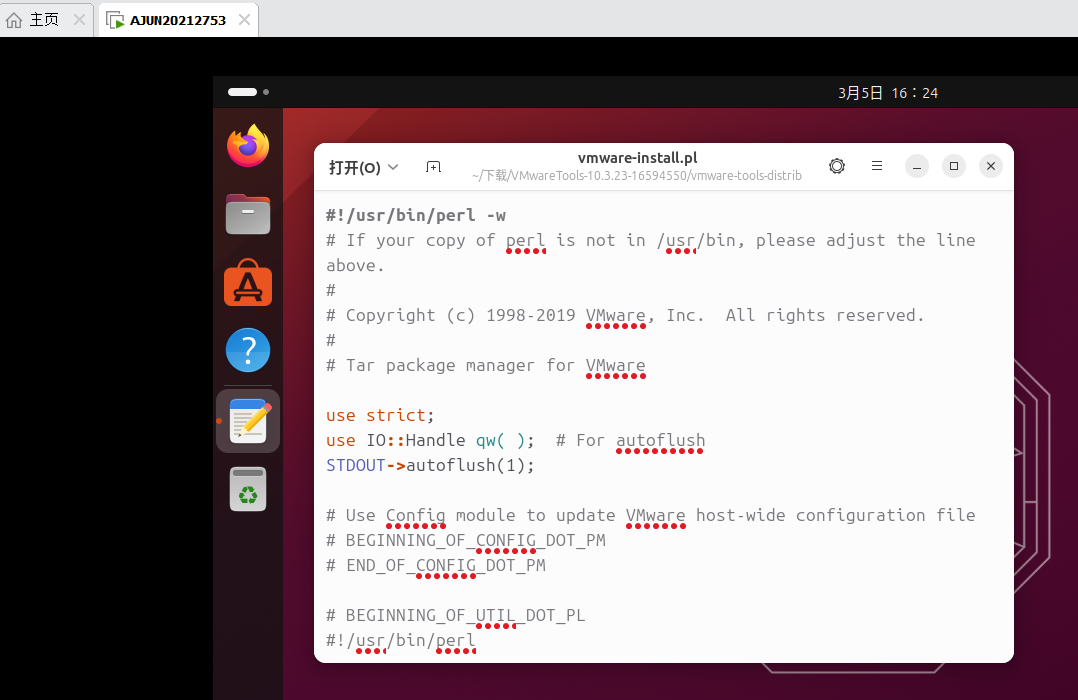
（2）启动应用程序、熟悉窗口操作。



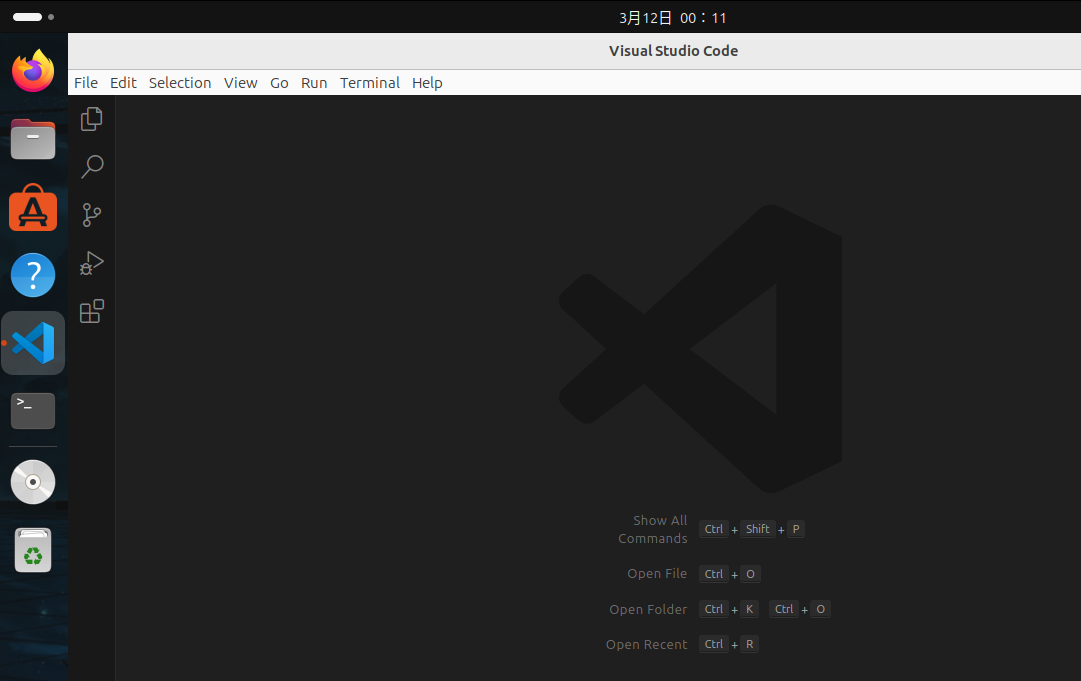
（3）使用文件管理器。



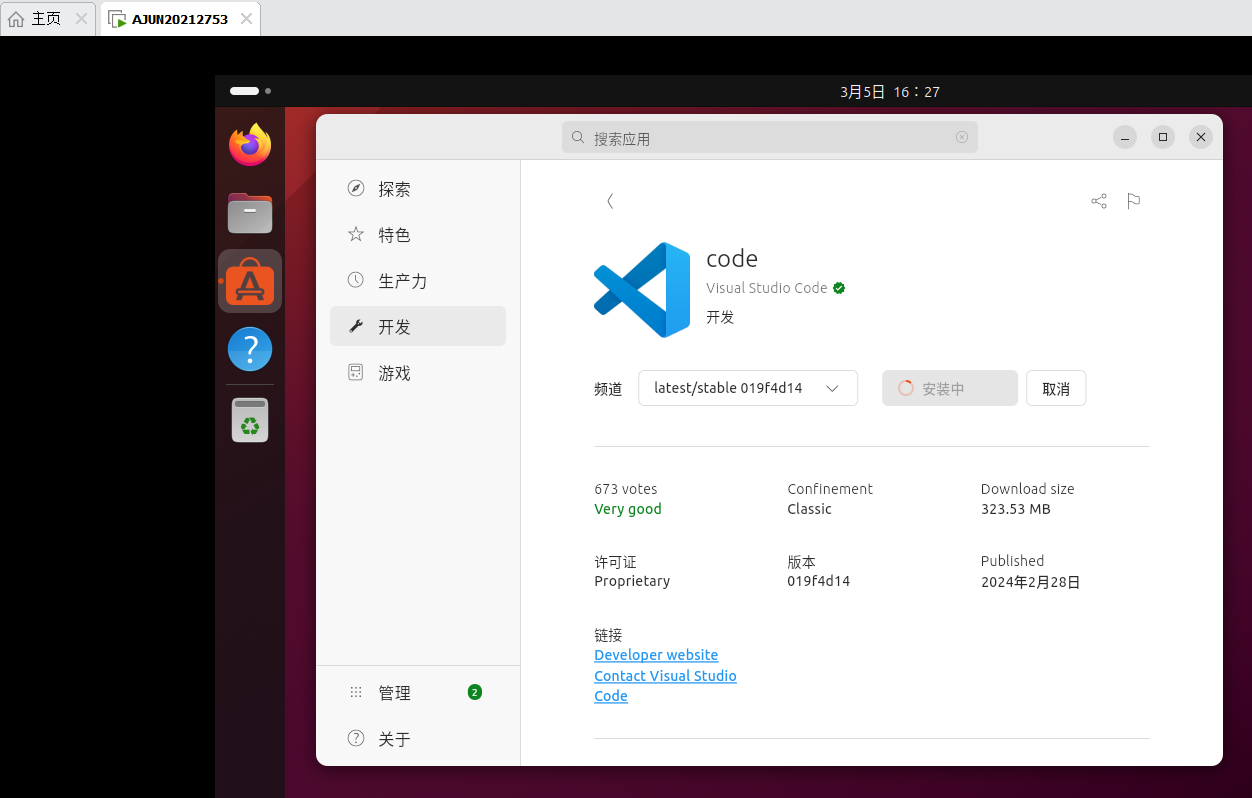
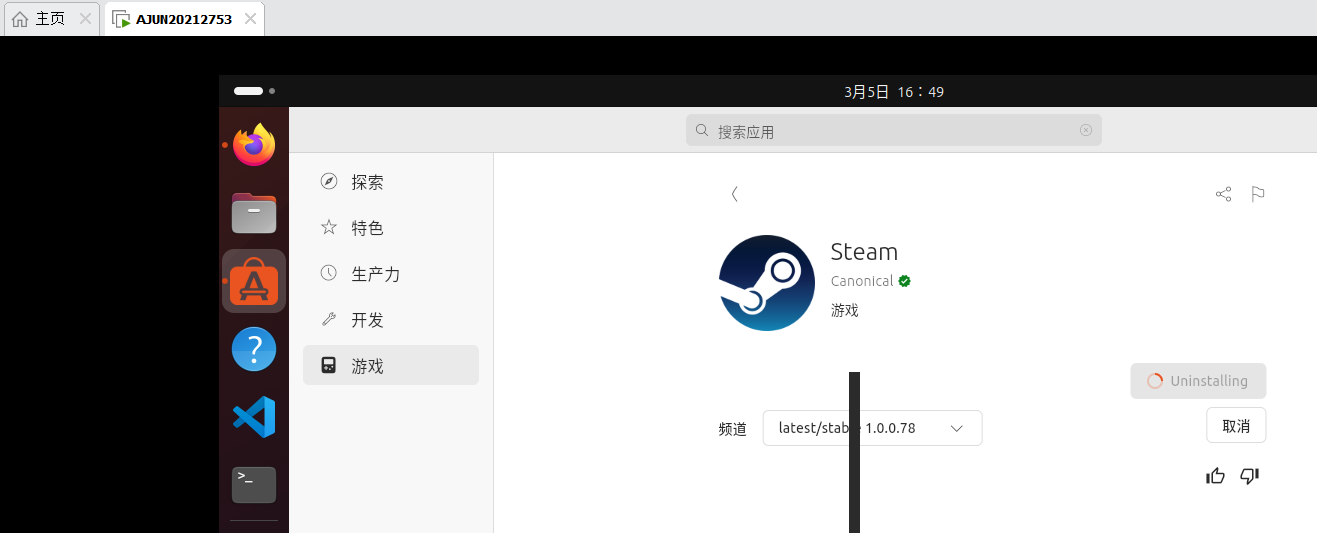
（4）使用gedit编辑器。



（5）使用office系列应用程序



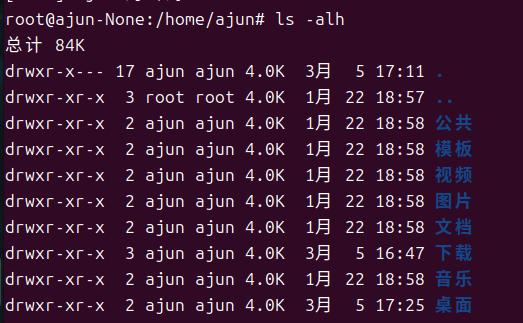
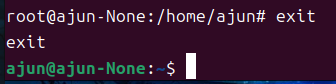
（6）使用软件中心安装和卸载软件。

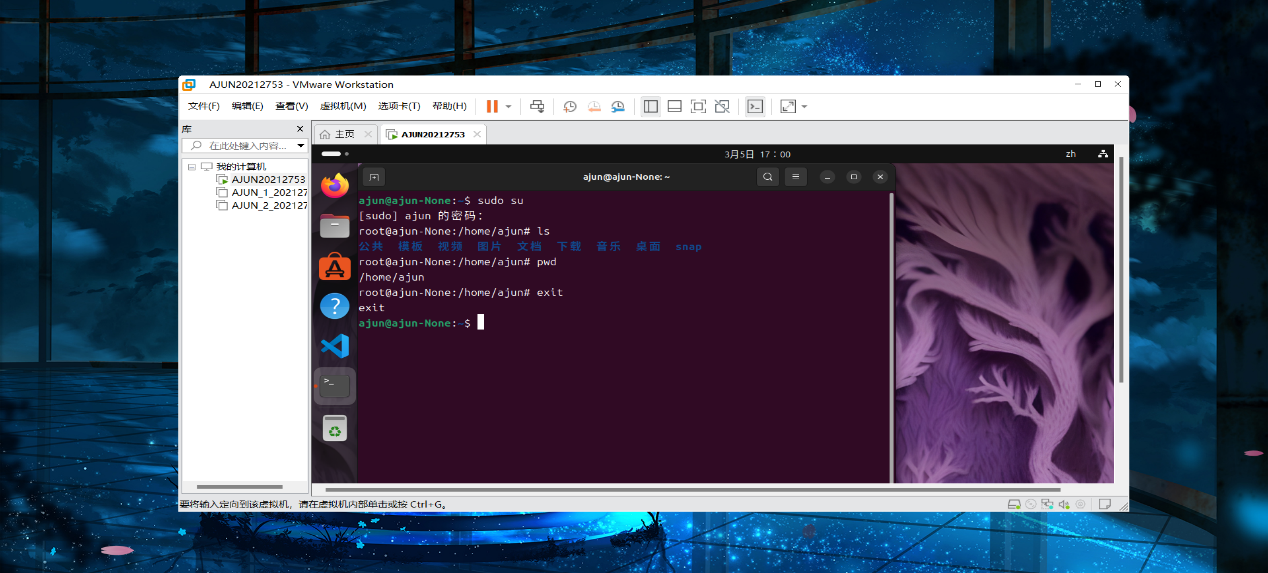
（7）进行桌面个性化设置，如：调整背景图片，分辨率等。



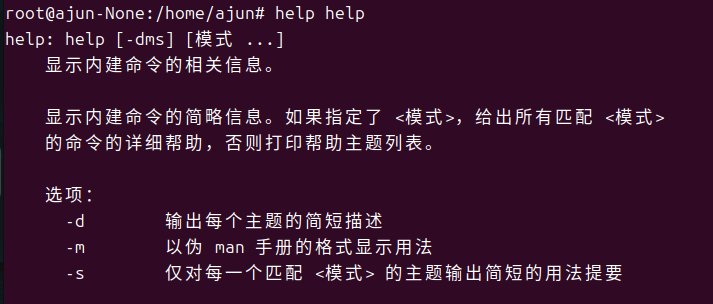
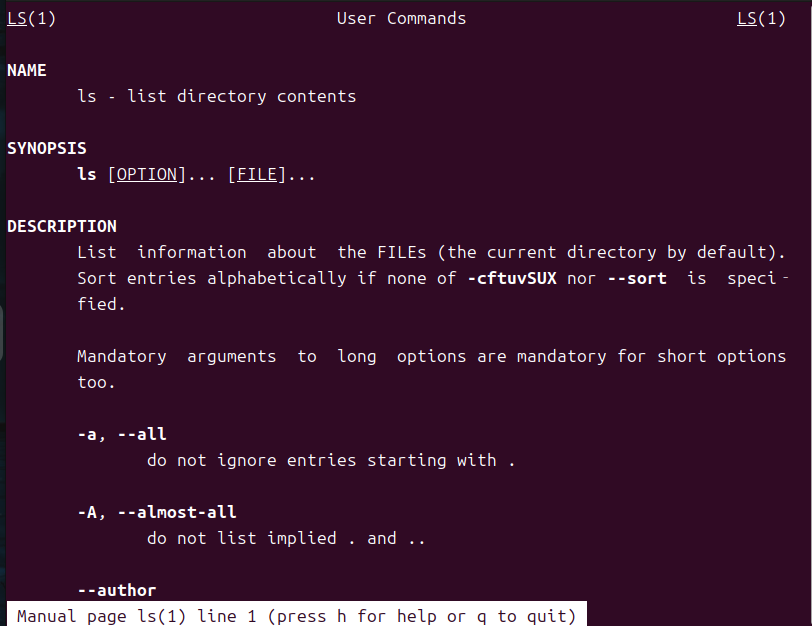
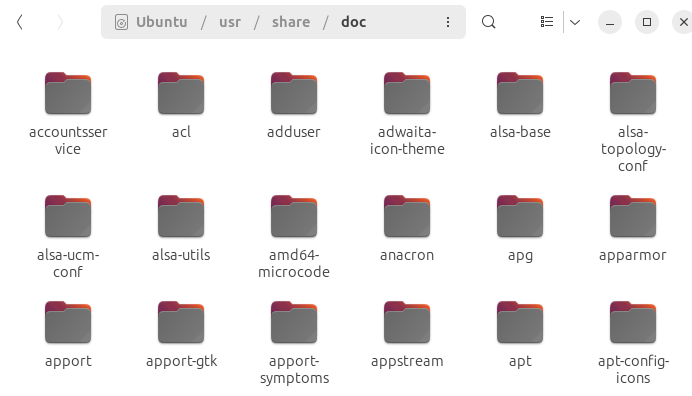
（8）使用ctrl+alt+t打开终端窗口，在界面中分别输入ls、pwd、exit命令并回车执行。

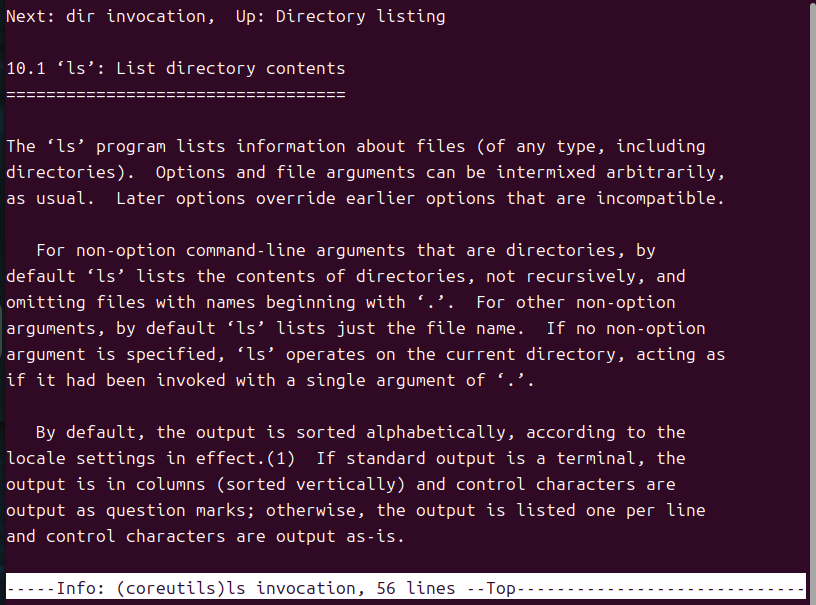
  

1. 两种系统界面的切换



1. 练习使用help、man、info，查看usr/share/doc内容



## 实验总结

ls [-alh] path 显示当前目录下的文件及文件夹

pwd 显示当前路径

exit 退出当前页面，如果处于root模式，则退出到普通模式，处于普通模式，则推出控制台

help 只针对系统内部命令

man内容清晰、详细,在线文档,支持搜索( /name ) man [ 章节 ] [name]

info 详细的帮助信息

/usr/share/doc 放帮助文档,在与软件同名的目录下有所有软件的使用文档

通过这次实验，我更深入地了解了Linux操作系统的桌面环境和基本操作，同时熟悉了一些常用的命令行工具。这为我在将来更深入地学习和使用Linux系统奠定了基础。在学习过程中，我也注意到不同Linux发行版可能存在一些差异，这进一步激发了我对Linux系统的学习和探索兴趣。

# 实验二 Linux操作系统基本操作与基本命令

## 实验目的

1. 掌握Shell基本功能与基本概念

2. 掌握Linux的基本命令

## 实验内容

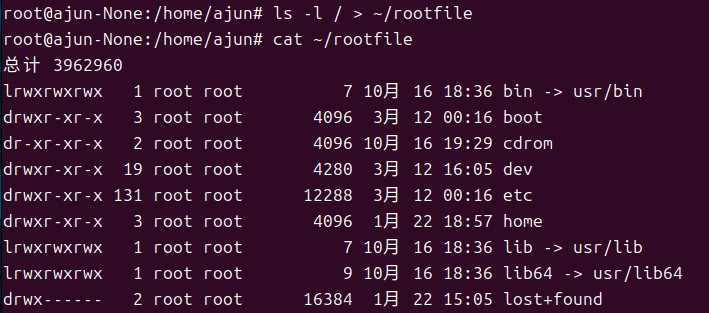
**一. Shell基本功能**

1. 练习shell命令格式与命令行编辑特性，教材3.1.5节。



1. 练习标准流与I/O重定向：
   1. 观察你的系统根目录下各目录的文件名、权限与属性，并保存结果到~/rootfile中。

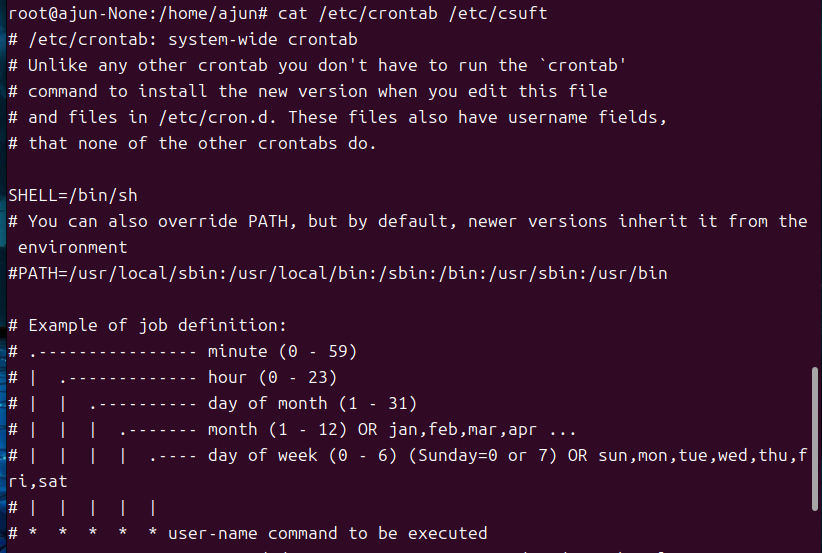
$ ll / ; ll / > ~/rootfile; cat ~/rootfile; ll ~/rootfile





* 1. 执行命令：cat /etc/crontab /etc/csuft，观察输出结果；

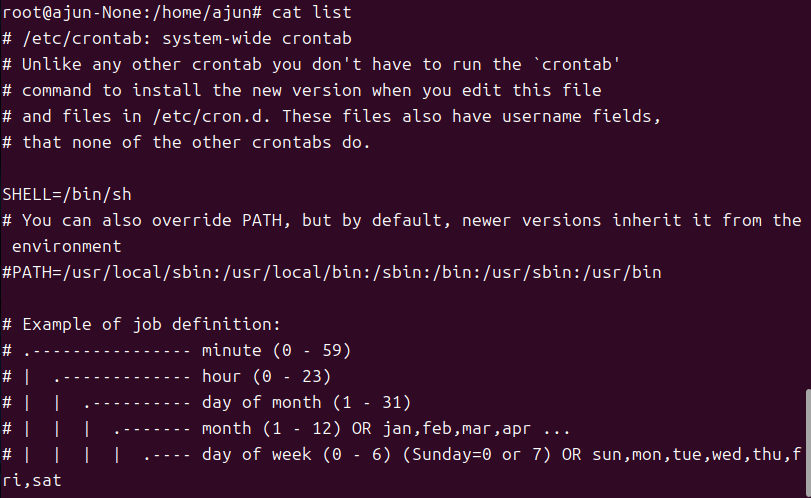
# 屏幕上会同时输出标准输出和错误错误输出。



* 1. 执行命令：cat /etc/crontab /etc/csuft > list，对比上一条命令，输出结果有何变化？为什么有此变化？

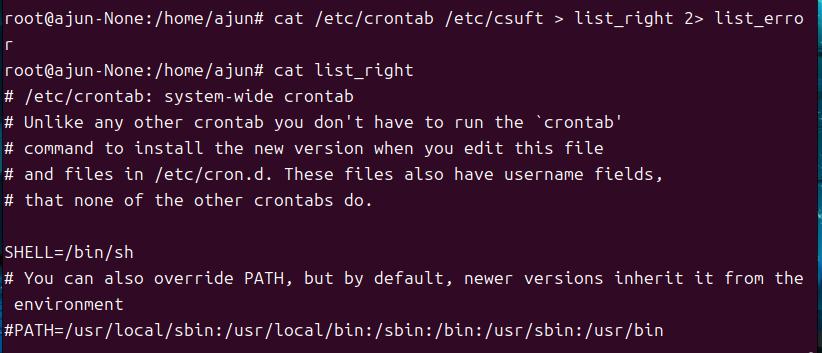
# 屏幕上只会有标准错误输出，可通过cat命令查看list文件内容是标准输出。





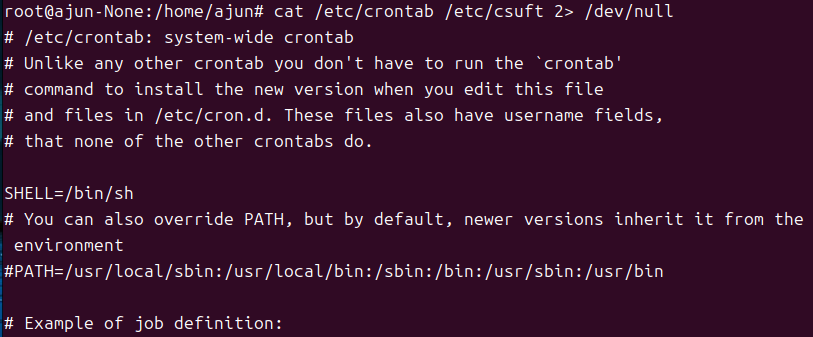
* 1. 执行命令，将stdout与stderr分别存到 list\_right 和list\_error文件中；

$ cat /etc/crontab /etc/csuft > list\_right 2> list\_error

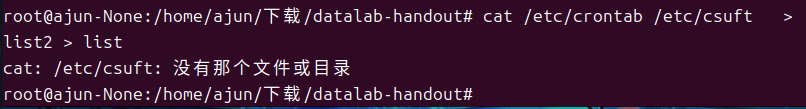
* 1. 执行命令，将错误的数据丢入/dev/null，屏幕上显示正确的数据；

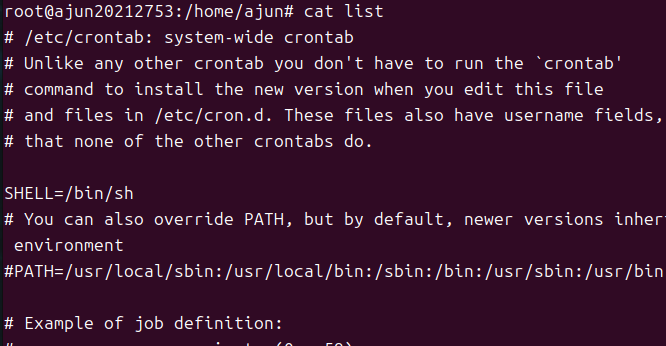
$ cat /etc/crontab /etc/csuft 2> /dev/null



* 1. 执行命令：$ cat /etc/crontab /etc/csuft > list 2> list（将数据全部写入list文件），查看list文件，与步骤（b）有何不同？为什么？

# 这条命令式错误的，两股数据同时写入一个文件，没有使用特殊的语法，两股数据可能会交叉写入该文件，造成次序的错乱。虽然最终list文件还是会产生，但是里面的数据排列与原本屏幕上的输出排序不同。



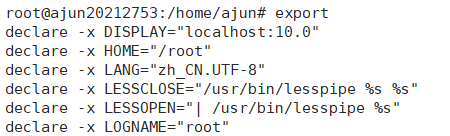


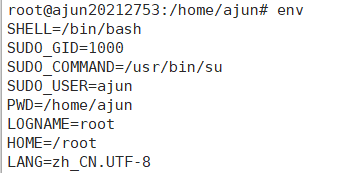
* 1. 执行指令，实现上一步的正确结果。

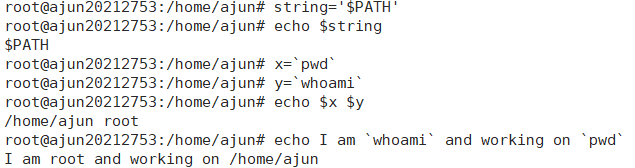
$ cat /etc/crontab /etc/csuft > list 2>&1; #或者cat /etc/crontab /etc/csuft &> list。



1. 练习教材3.1.8、3.1.9的相关内容。



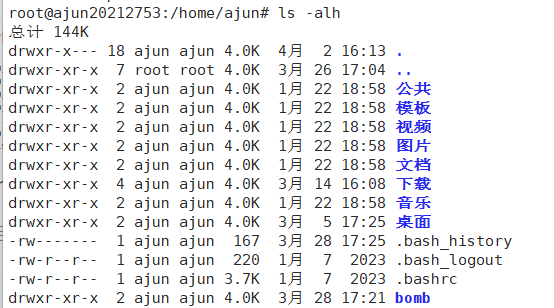


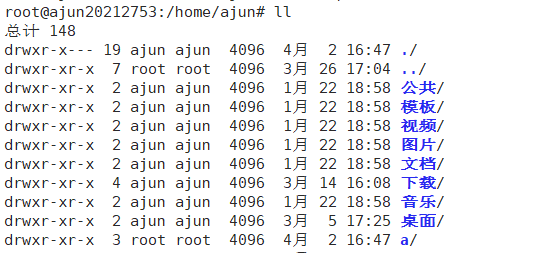




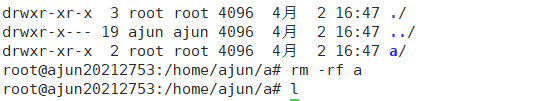
**二. Linux系统基本命令**

1. 练习教材3.2.1节目录操作基本命令ls、mkdir、rmdir、cd、pwd。

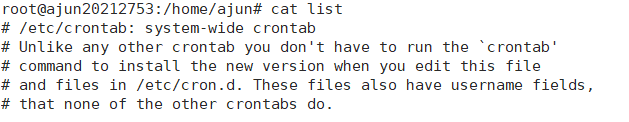


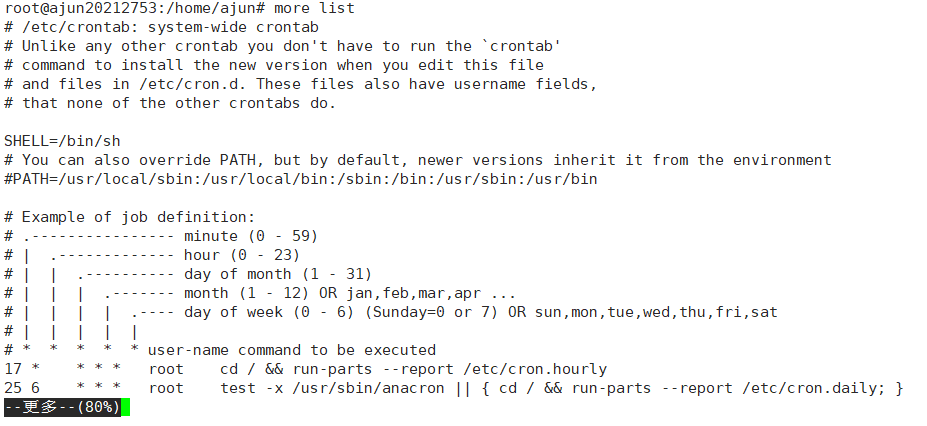






2. 练习教材3.2.2 文件操作基本命令cat、more、less、wc、head/tail、mv、rm。

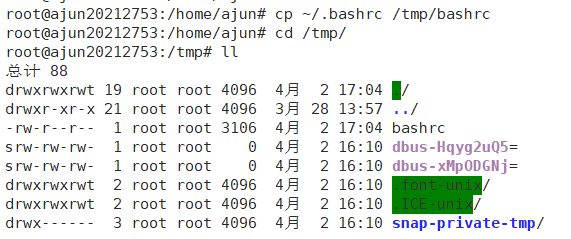




3. 练习文件和目录复制命令cp：

1. 用root身份，将主文件夹下的 .bashrc 复制到 /tmp 下，并更名为 bashrc；

[root@study ~]# cp ~/.bashrc /tmp/bashrc

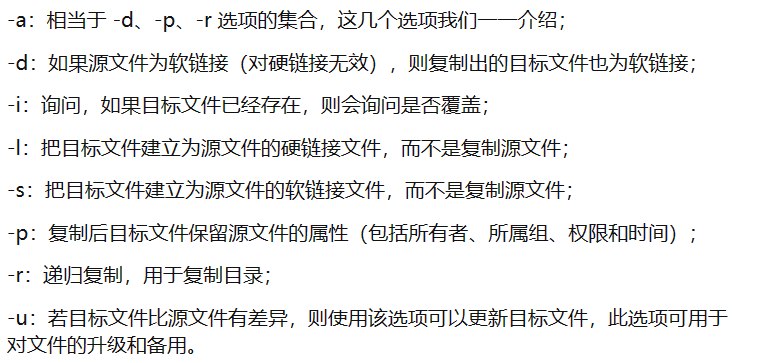


1. 变换目录到/tmp，并将/var/log/wtmp复制到/tmp并观察文件属性差异；如何将文件所有特性都复制？



-p 保持源文件属性

1. 练习cp命令其他选项。



## 实验总结

通过本次实验有效联系了linux操作系统的基本操作和基本命令，为以后的linux操作系统学习奠定了基础。

# 实验三 文本编辑器vi/vim

## 实验目的

1. 掌握vi/vim编辑器的进入和退出方式
2. 了解vi/vim的三种模式
3. 熟练vi/vim的操作命令

## 实验内容

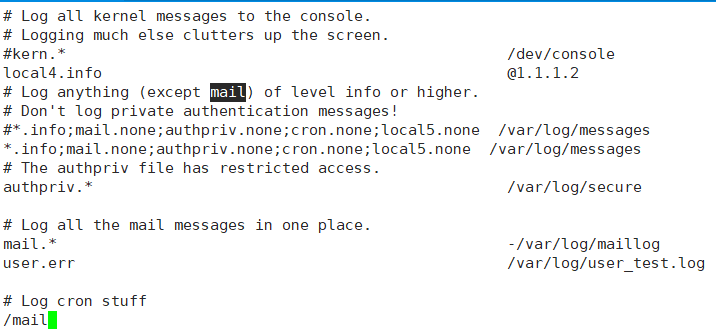
**（注意：如果是Ubuntu系统，就自行复制一个文件练习vi/vim相关命令）**

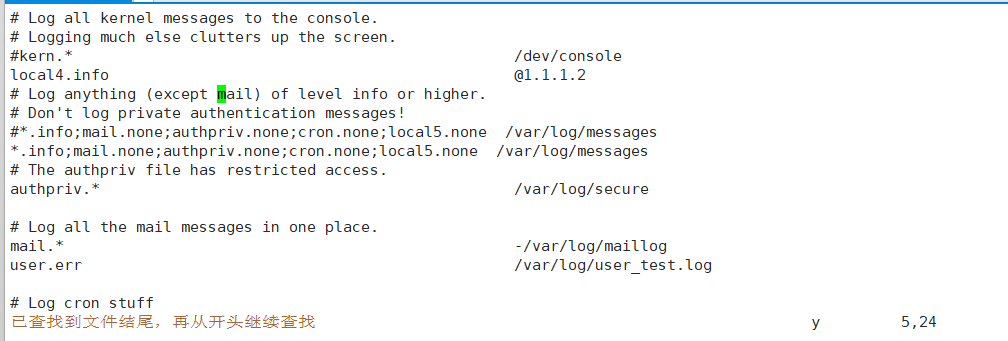
1. 在家目录下新建一个名为“vitest\_name”（“name”为学生姓名拼音）的目录。

2. 执行复制命令，将文件/etc/syslog.conf复制到vitest\_name目录下。

3. 使用vi/vim打开vitest\_name 目录下的syslog.conf文件，并完成以下操作：

1. 指出该文件中设置mail的所在行





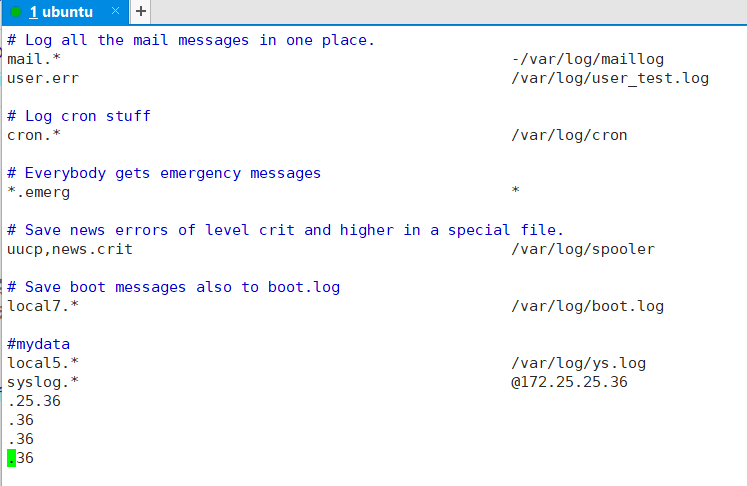
1. 将光标移到该行，并复制该行内容

`yy`



1. 将复制内容粘贴到最后一行

`G`跳转到最后一行



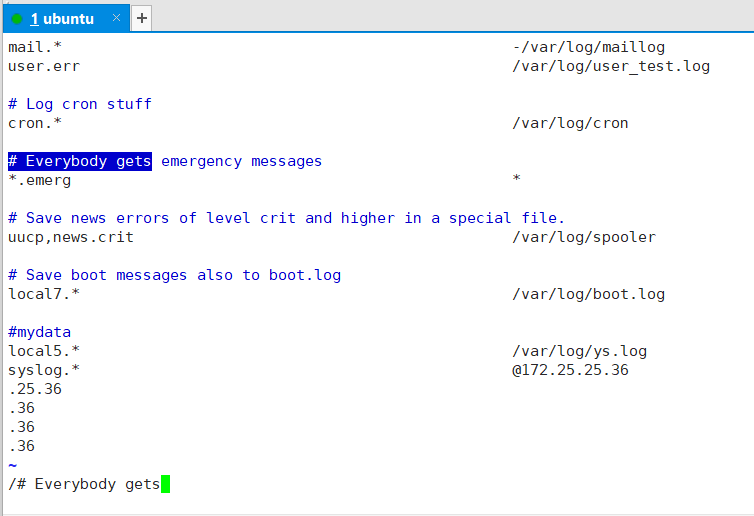


1. 撤销上一步的粘贴动作

`u`



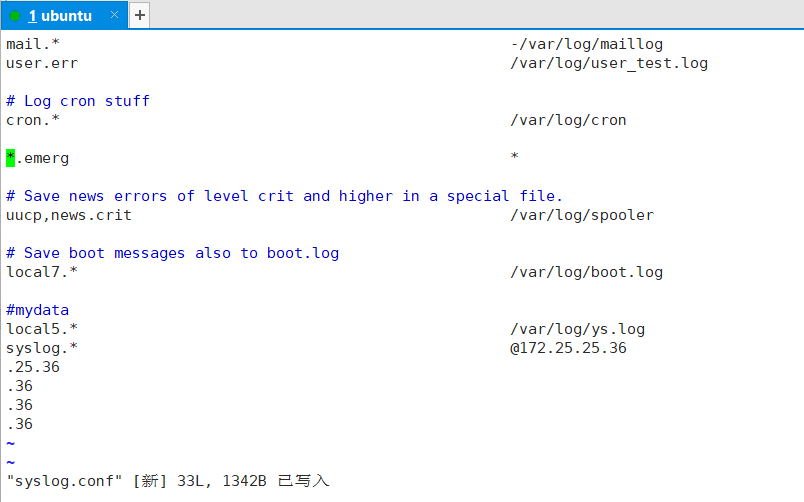
1. 将光标移到“#Everybody gets……”，删除该行



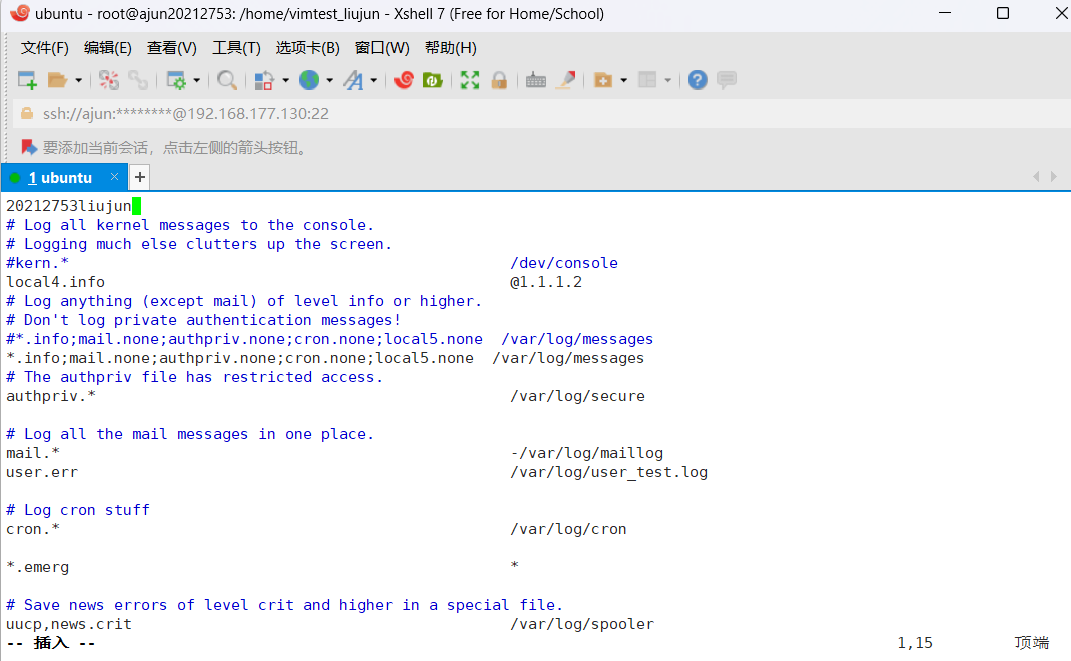
`dd`



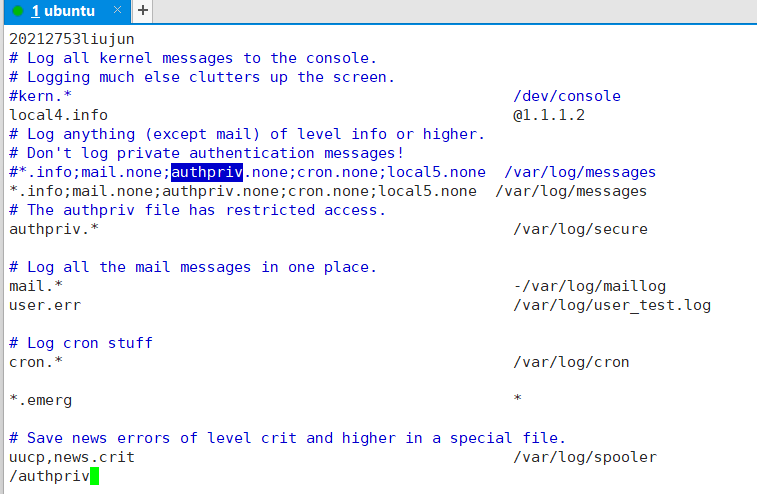
1. 存盘但不退出



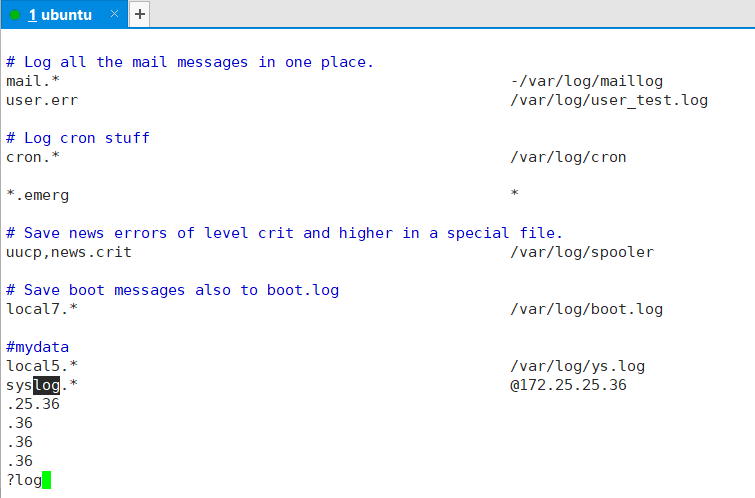
1. 将光标移至首行，并输入你的学号姓名等信息



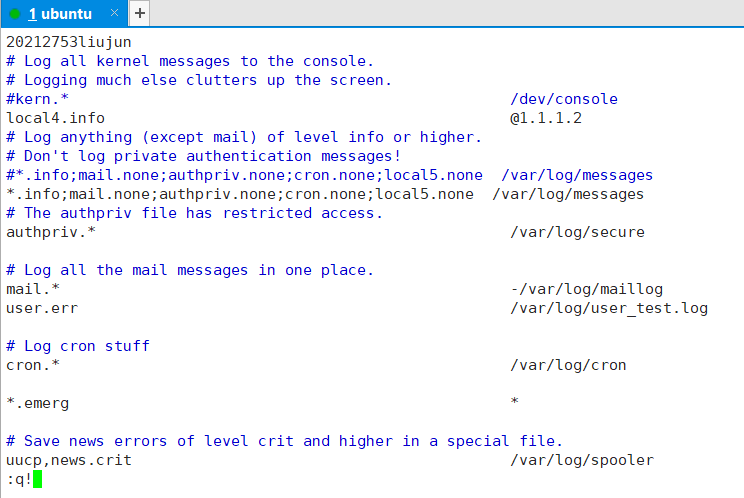
1. 向下查找字符串“authpriv”



1. 向上查找字符串“log”



1. 强制退出vi/vim，不存盘



## 实验总结

通过本次实验学习掌握了vim编辑器的使用，使我在后续的实验中能更加便利地对文件进行读写操作。

# 实验四 用户、组、密码及权限管理

## 实验目的

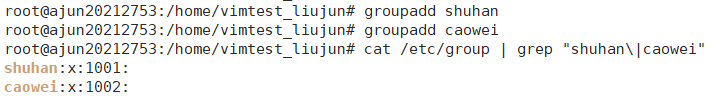
（1）掌握用户、组合密码管理基本命令

（2）掌握权限管理基本命令

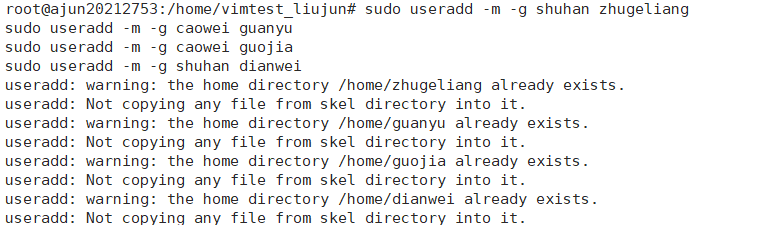
## 实验内容

**（以下步骤都需要通过命令验证操作结果）**

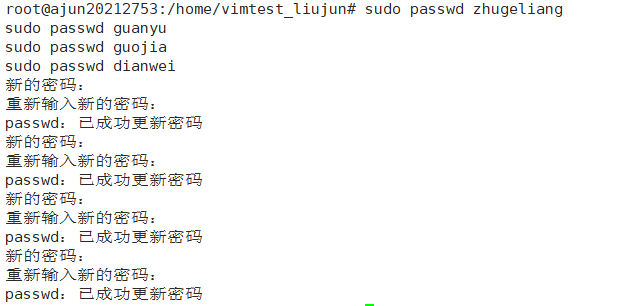
1. 创建组shuhan、caowei；验证举例：打开组相关文件，只显示shuhan和caowei两个组信息；



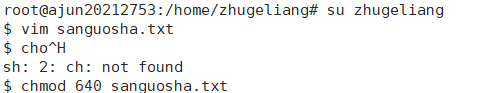
1. 创建用户 zhugeliang、guanyu，设置所属组为shuhan；创建用户 guojia、dianwei，设置所属组为caowei；



1. 设置密码



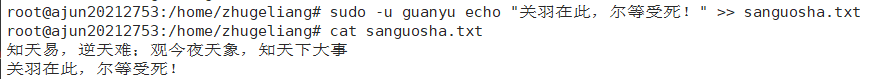
1. zhugeliang使用vim创建一个文件“sanguosha.txt”，写入“知天易，逆天难；观今夜天象，知天下大事”，并设置权限：自己可以读写，本组人可读，其他组没有任何权限；



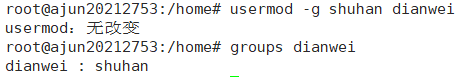
1. zhugeliang修改“sanguosha.txt”的文件属性，让本组人可以读写；



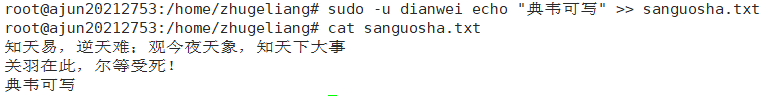
1. guanyu修改“sanguosha.txt”，加入一句话（“关羽在此，尔等受死！”）



1. dianwei投靠shuhan（修改dianwei的组）；



1. 测试dianwei是否可以读写“sanguosha.txt”，若可以，在文件中追加“典韦可写”；guojia是否可以读？若不能，为什么？应怎么解决？至少写出两种解决办法。



guojia无法读取文件，因为他不属于文件所属组。解决方法：

方法一：将guojia加入shuhan组。

sudo usermod -aG shuhan guojia

方法二：更改文件权限，使所有用户都可以读取。

chmod +r sanguosha.txt

1. dianwei修改“sanguosha.txt”，加入一句话（“吃我一戟！看我三步之内取你小命！”）



## 实验心得

在进行这个实验的过程中，我学到了如何在 Linux 环境下进行用户和组的管理，以及如何设置文件的权限。首先，我创建了两个组：shuhan 和 caowei，并且通过验证确保了它们已成功创建。接着，我创建了四个用户：zhugeliang、guanyu、guojia 和 dianwei，并分别将它们分配到相应的组中。

随后，我设置了这些用户的密码，以确保他们的安全性。然后，我让 zhugeliang 使用 vim 创建了一个文件 sanguosha.txt，并设置了特定的权限，使得只有他自己可以读写，而组内其他成员可以读取，其他组没有任何权限。

在后续的步骤中，我让 zhugeliang 修改了 sanguosha.txt 的文件属性，让组内成员也可以读写该文件，并且让 guanyu 添加了一句话到这个文件中。接着，我让 dianwei 加入了 shuhan 组，这样他就可以访问 sanguosha.txt 文件了。

最后，我测试了 dianwei 是否可以读写 sanguosha.txt 文件，结果发现他可以，并在文件中追加了一句话。同时，我尝试让 guojia 读取该文件，但由于他不属于文件所属组，所以失败了。我探索了两种解决方法：将 guojia 加入 shuhan 组，或者更改文件权限让所有用户都可以读取。

总的来说，这个实验让我更加熟悉了 Linux 系统下的用户、组和文件权限管理，也增强了我在命令行环境下进行系统管理的能力。我觉得这些知识对我以后的工作和学习都会有很大帮助。

## 实验五 文件系统管理及使用

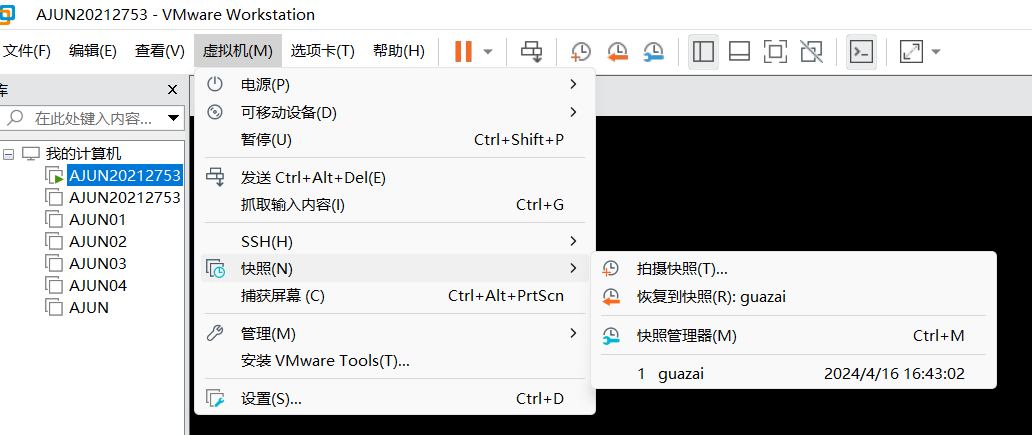
## 实验目的

1. 理解磁盘分区、挂载、卸载的概念

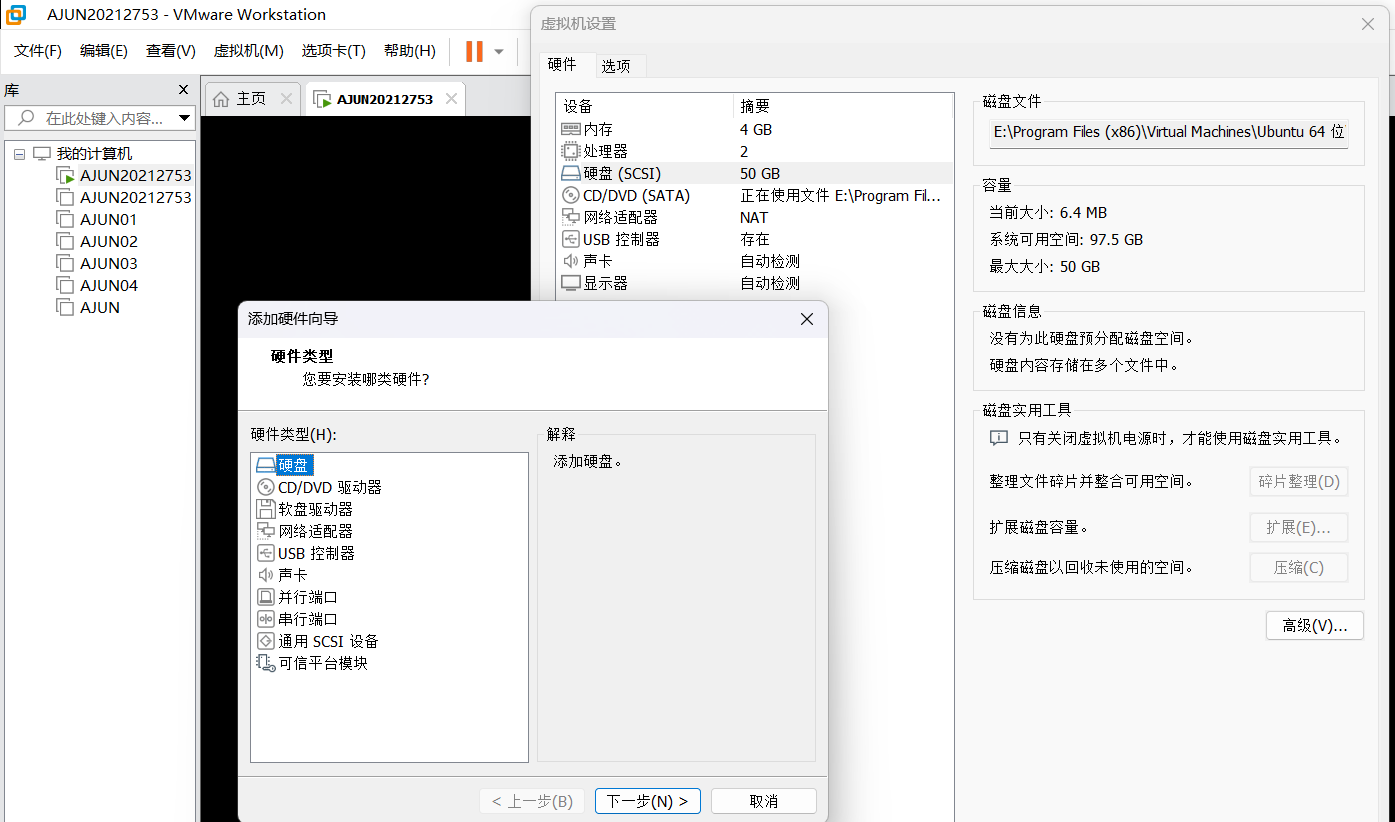
2. 熟悉与文件系统管理相关的命令

## 实验内容

1. 以增加一块硬盘为例来熟悉磁盘管理相关指令，并理解磁盘分区、挂载、卸载的概念。
2. 虚拟机添加硬盘
   1. 先在vmware中“拍摄此虚拟机的快照”，备份当前linux系统，实验中若有错误操作，可通过快照进行恢复。

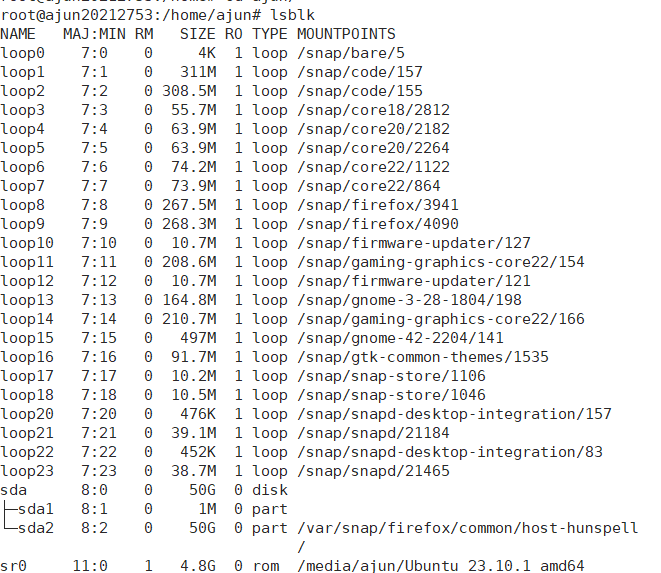


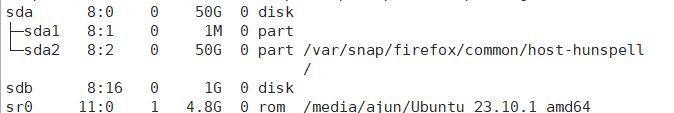
* 1. 在【虚拟机】菜单中，选择【设置】，然后设备列表里添加硬盘，然后一路【下一步】，中间只有选择磁盘大小的地方需要修改（设置1GB就可以了），直到完成。





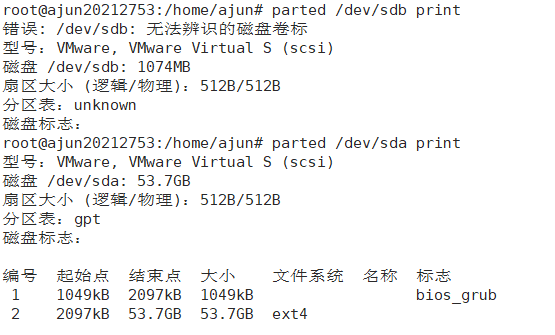
* 1. 完成后使用$ lsblk 命令查看磁盘情况，可看到只有sda，没有sdb。重启 ($ reboot) 后，系统才能识别新增的硬盘，再执行命令$ lsblk，出现sdb，但没有分区，也没有mountpoint。





1. 观察磁盘分区状态

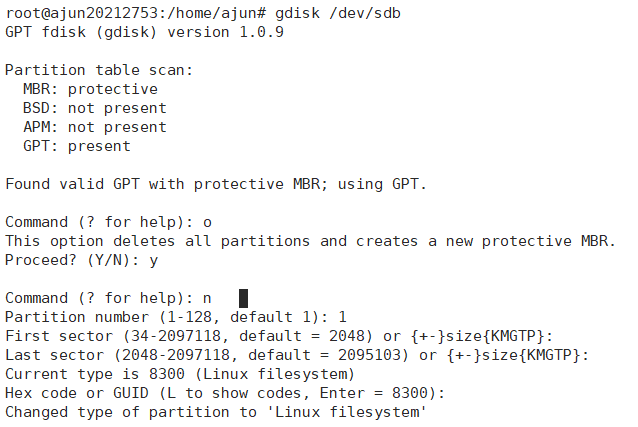
执行parted命令，列出磁盘的分区表类型与分区信息。根据分区表类型，决定下一步分区使用fdisk还是gdisk。



1. 分区

执行命令：gdisk /dev/sdb

（说明：开始分区后输入n，新增分区——>输入p，选择分区类型为主分区——> selected partition (1-4, 默认为1)，数字是几就会分几个区，此处可以设置为1——>两次回车默认剩余全部空间——>输入w，写入分区并退出，若不保存退出则输入q。）

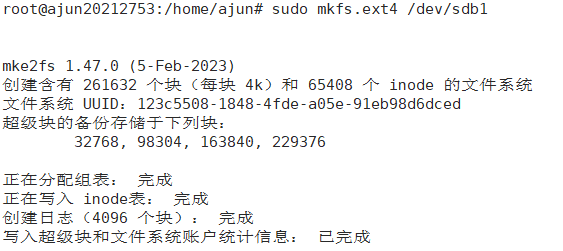




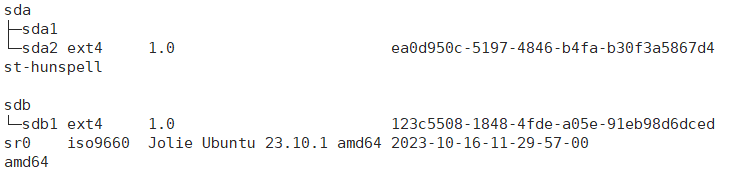
1. 格式化（创建文件系统）

可以先执行$ lsblk -f 查看sdb1是否已经格式化，有UUID就表明已经格式化；没有，则利用mkfs命令进行格式化。

执行命令：mkfs -t ext4 /dev/sdb1



再次执行命令：lsblk -f #观察sdb1是否有UUID，是否有mountpoint。



1. 挂载

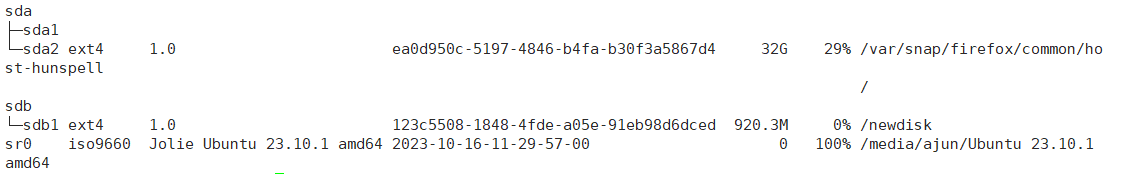
挂载：将一个分区与一个目录联系起来

命令：$ mount 设备名称 挂载目录

例如：$ mount /dev/sdb1 /newdisk

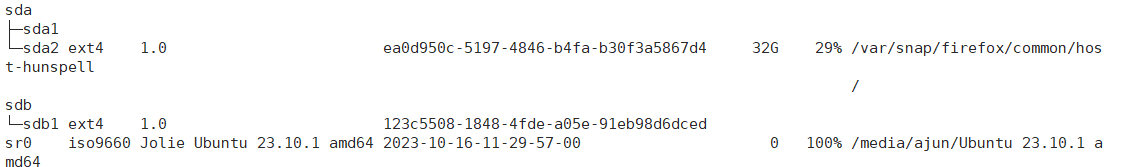


执行命令$ lsblk -f ，观察sdb1是否有挂载点



执行命令：$ reboot

执行命令 $ lsblk -f ，观察sdb1的挂载点。



1. 设置可自动挂载

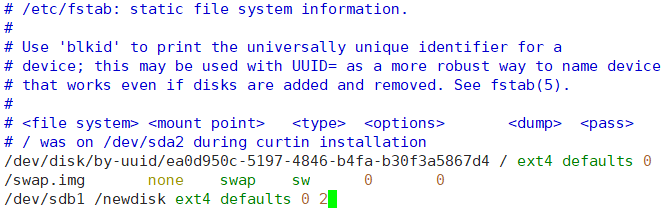
永久挂载：通过修改/etc/fstab实现

使用root权限：# vim /etc/fstab

用yy复制一行有效行，用p粘贴，修改粘贴的那行：

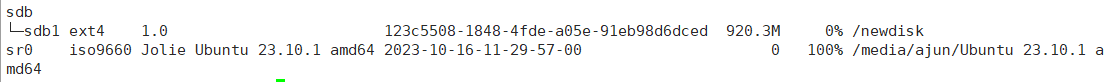
* 修改filesystem，将UUID改为sdb1的UUID（不记得UUID也可以用设备名，即/dev/sdb1）；
* 修改挂载点mount\_point为/newdisk；
* 文件类型fstype还是ext4；
* 文件系统安装选项options还是default；
* 修改备份标志dump，就填0（0表示不备份，1表示备份）；
* 修改开机时系统自动检查的顺序pass，填0（0为不检查，否则进行检查。根文件系统中1表示检查，其他文件系统中2表示检查）

保存退出vim编辑。



执行命令$ reboot，

执行命令 lsblk -f ，观察sdb1的挂载点是否还在。



1. 卸载

卸载，并通过命令查看该设备是否已经卸载

命令：$ umount 设备名称 或者 挂载目录

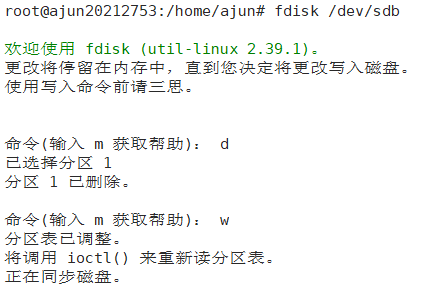
例如：$ umount /dev/sdb1 或者 $ umount /newdisk



查看是否已经卸载：$ lsblk #发现sdb1这个设备的挂载点没有了。

1. 删除分区

执行命令# fdisk /dev/sdb



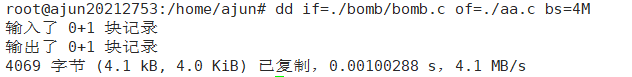
进入fdisk交互界面后，输入d删除分区，然后输入w保存并退出。

再$ lsblk，可以看见sdb1没有了。

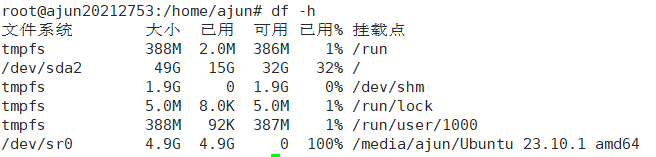
1. 练习与文件系统管理相关的命令：find、locate、dd、ln、df、du









## 实验总结

本次实验通过在虚拟机中添加硬盘，学习了磁盘管理相关指令，并深入理解了磁盘分区、挂载、卸载的概念。通过执行命令观察磁盘状态、分区、格式化、挂载以及设置自动挂载等步骤，加深了对文件系统管理相关命令的理解和掌握。同时，通过实践操作，提高了对Linux系统的熟练度，为今后的系统管理工作奠定了基础。

## 实验六 进程与任务管理

## 实验目的

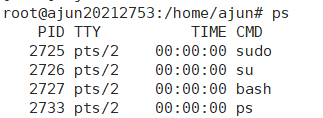
（1）熟悉进程管理常用命令

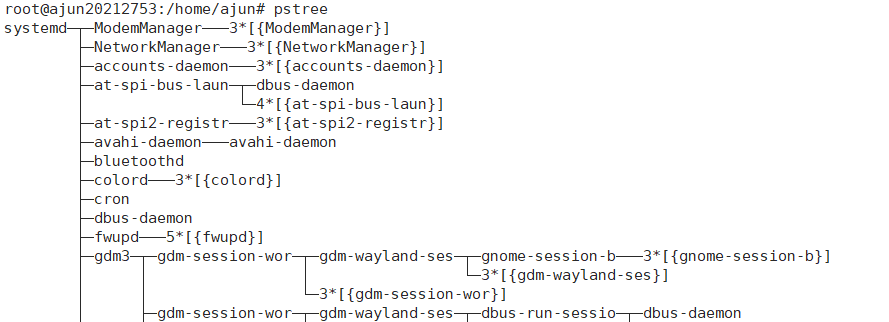
（2）熟悉定时任务与任务调度的设置

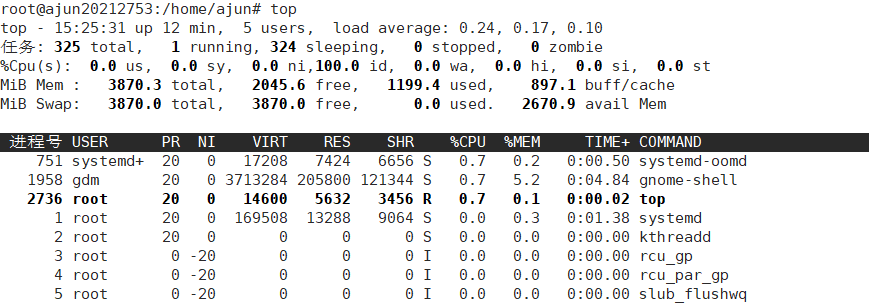
## 实验内容

**1. 进程管理**

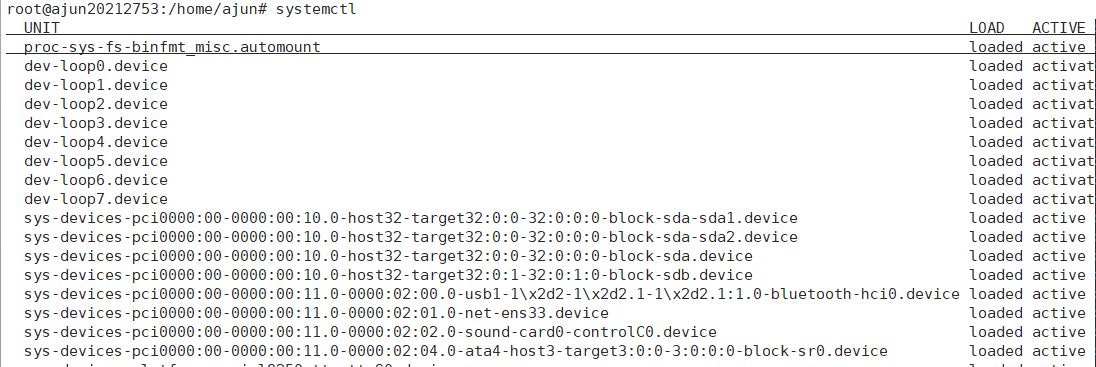
（1）练习进程状态查询命令ps、pstree、top

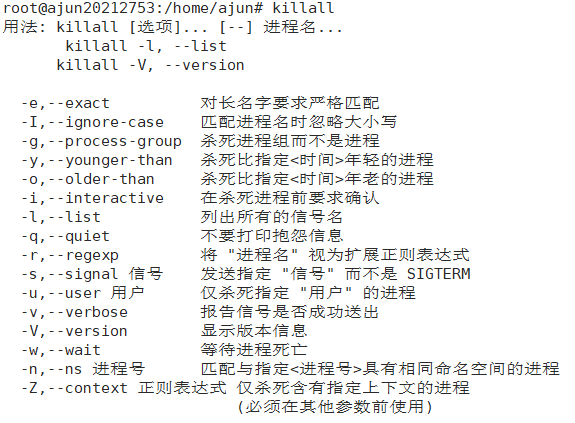


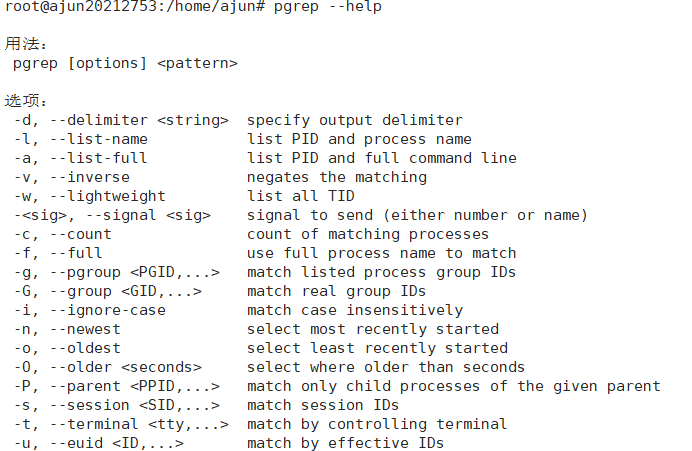


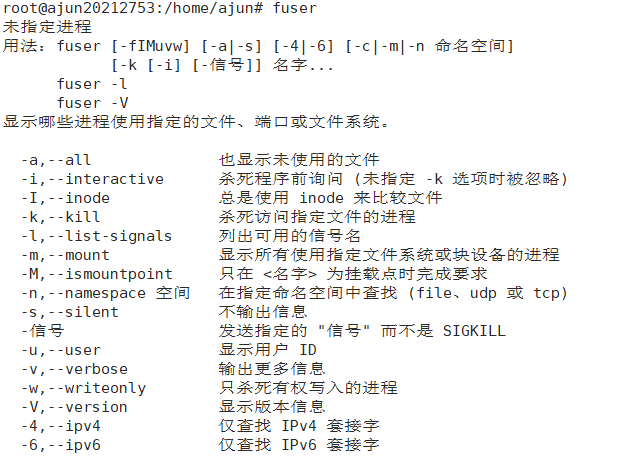


（2）练习进程管理命令systemctl、killall、pgrep/pkill、fuser、nice











（3）练习可执行文件的特殊属性操作

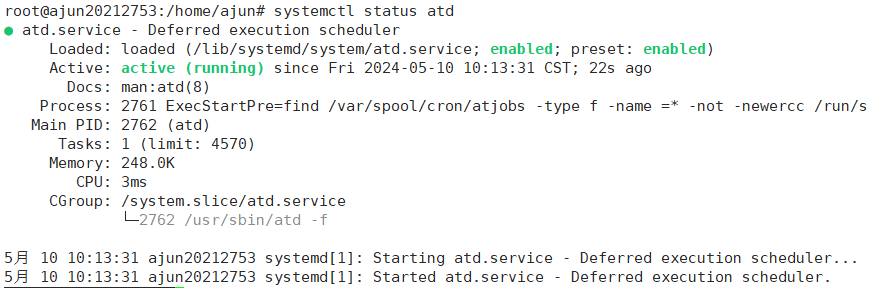
1）新建文件

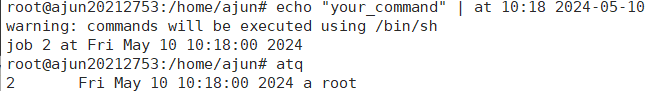
2）查看文件权限

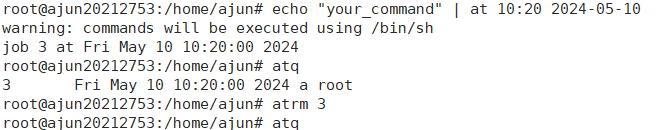
3）设置、修改、查看文件特殊权限suid、sgid及sticky

**2. 任务管理**

（1）利用at命令完成定时任务设置、查看并删除任务。注意：执行at时，需要先保证atd已启动，可执行systemctl命令重启atd，并观察atd的状态。



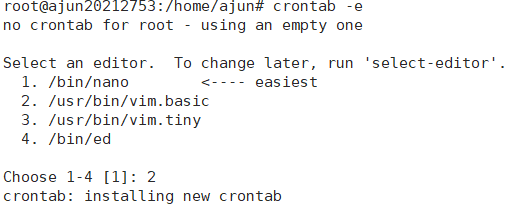




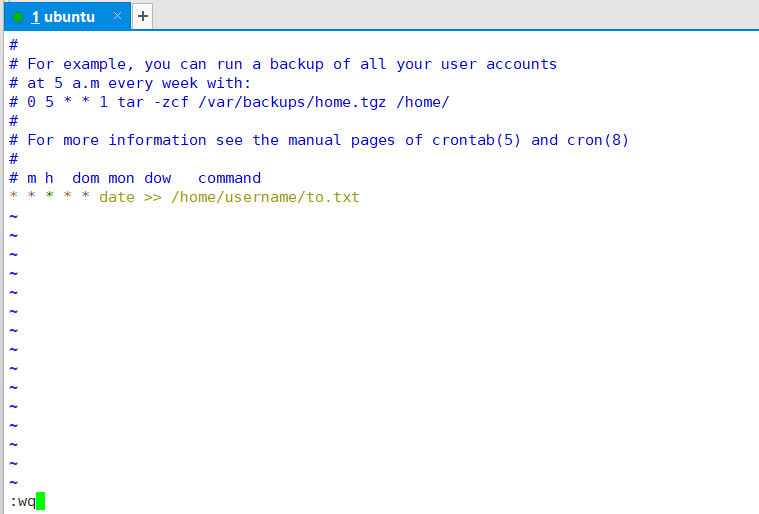
（2）利用 crontab完成任务调度设置：

* 设置任务调度文件/etc/crontab，具步骤如下：

1. 执行命令$ crontab -e，若是首次使用crontab，需选择系统已经安装的editors，建议选vi/vim；



1. 输入任务到调度文件：每隔一分钟，将当前日期追加到/home/username/to.txt；

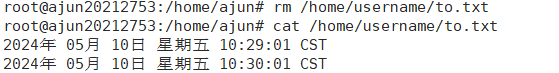


1. 保存退出，提示任务创建成功“crontab: installing new crontab”；



1. cat /home/username/to.txt，观察输出内容；



1. 删除/home/username/to.txt；
2. 一分钟后执行ls命令，查看是否有/home/username/to.txt重新生成。

* 有时候需要调度多个任务，可用crontab -e 增加多条任务，也可以将这多个任务写在一个脚本文件中，**以crond调用脚本的方式实现多个任务的提交**，但这种方式需要注意脚本文件有**执行权限**。请完成以下步骤：

1. 执行命令$ vim /home/username/my.sh，新建脚本文件并写入内容：

date >> /home/username/mycal

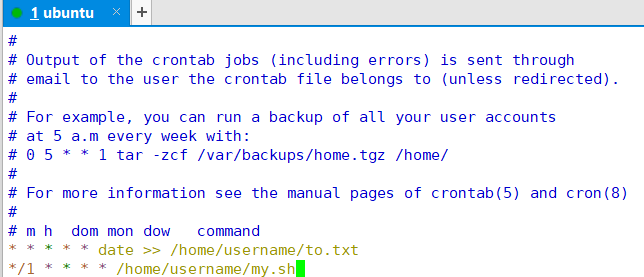
cal >> /home/username/mycal



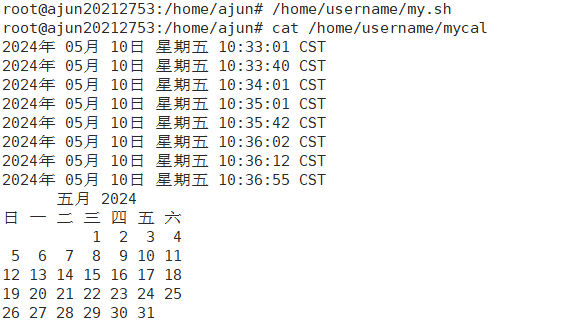
1. 给my.sh增加执行权限



1. 执行命令$ crontab -e ，增加任务：\*/1 \* \* \* \* /home/username/my.sh；保存退出。



1. 执行命令$ ./my.sh，执行脚本文件，观察输出；执行命令$ cat /home/username/mycal 查看mycal文件的内容，是否有日期和日历，且每隔一分钟，是否有追加内容。



1. 执行命令$ crontab -l ，列出当前定时任务



1. 执行命令$ crontab-r，删除定时任务



1. 重启crond。



## 实验总结

本次实验涉及了 at 和 crontab 两种常用的定时任务调度工具。通过 at 命令，能够设置一次性任务，对未来某一时刻执行操作十分便捷。而通过 crontab 命令，则可以设置周期性任务，例如每天、每周或每月执行特定任务。通过学习实验，我掌握了使用这两种工具的方法，能够更有效地管理系统任务，提高工作效率。

## 实验七 shell 编程

## 实验目的

（1）掌握linux “三剑客”grep、sed、awk的使用；

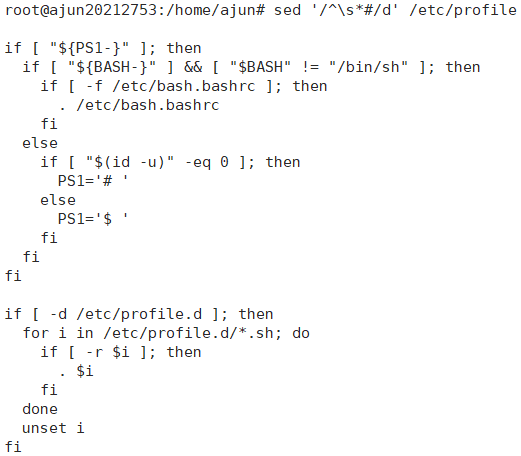
（2）掌握bshell脚本编程基本规则、函数及控制结构。

## 实验内容

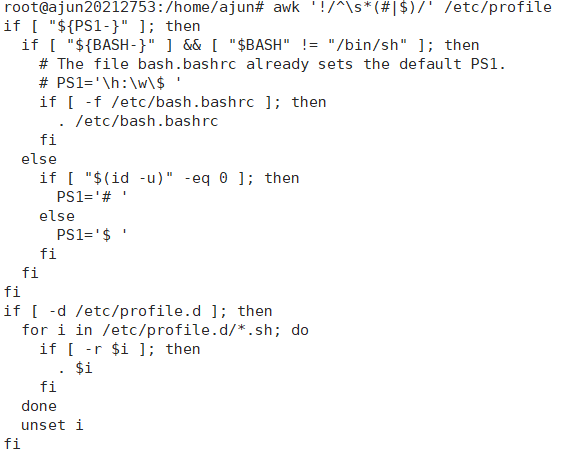
1. 试分别用grep、sed和awk实现：对某个脚本文件ifile（比如/etc/profile）进行如下操作：
2. 显示其中的所注释行（含#开始的行，或#号前全是白空格开始的行）；



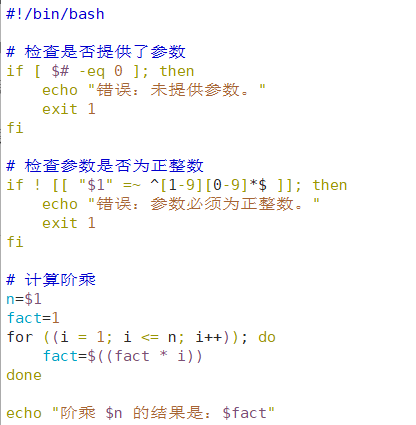
1. 显示去除了所有注释行的内容；

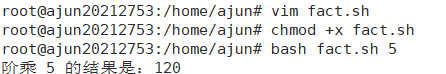


1. 显示所有的（去掉注释和空行）有效行。



1. 设计一个shell程序计算n的阶乘。要求：
2. 从命令行接收参数n；
3. 在程序开始后立即判断n的合法性，即是否有参数，若有，是否为正整数，若非法请给错误提示；
4. 最后输出计算的结果。





## 实验总结

通过本次实验学习了shell编程，能够通过shell命令实现一些简单的程序，掌握了bshell脚本编程基本规则、函数及控制结构。

## 实验八 Linux开发工具的使用

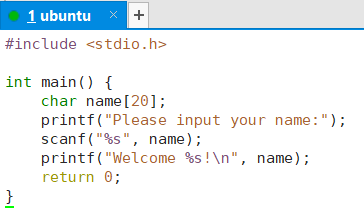
## 实验目的

（1）掌握gcc的编译方法与使用

（2）掌握Linux调试器GDB的使用

## 实验内容

1. **编译器gcc的使用**
   1. 编辑一个C语言程序文件 hello.c ，代码如下：



* 1. 编译文件： gcc -o hello hello.c。

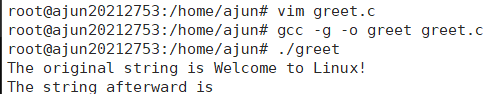


* 1. 若有错误，修改hello.c的内容，然后再次编译，直至没有错误为止。

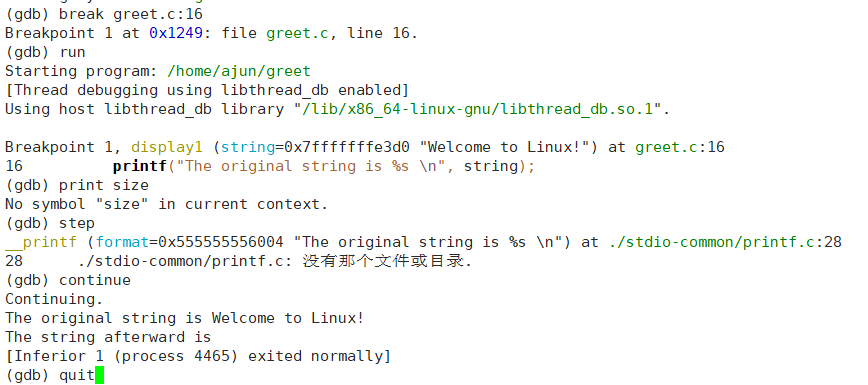
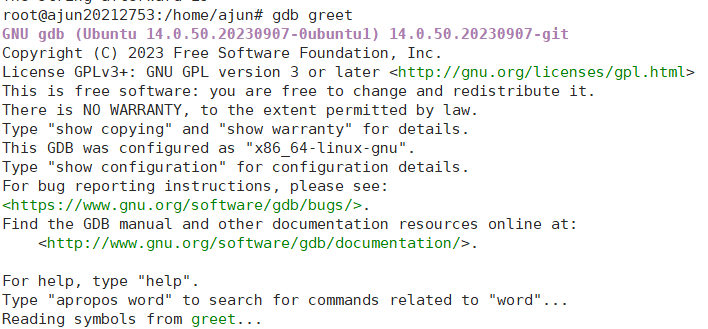
1. **使用GDB 调试程序BUG（教材12.7节）**
   1. 使用文本编辑器输入以下代码greet.c。程序试图倒序输出main 函数中定义的字符串，但结果没有显示。



* 1. 使用gcc –g 的选项编译这段代码，运行生成的可执行文件，观察运行结果。

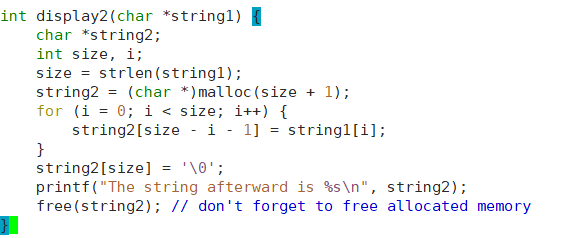


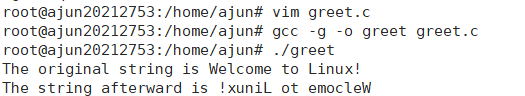
* 1. 使用gdb 调试程序，通过设置断点、单步跟踪，一步步找出错误所在。（调试过程需截图）



* 1. 纠正错误，更改源程序并得到正确的结果。

在 display2 函数中，修复内存越界错误，更改 for 循环，并将 string2[size + 1] = '\0'; 修改为 string2[size] = '\0';，以确保正确的字符串结束符位置。





## 实验总结

在本次实验中，我学习了使用GCC编译器编译C语言程序以及通过GDB调试器来定位和解决程序中的错误。通过设置断点、逐步执行程序并查看变量值，我们能够更有效地理解程序的执行流程并发现潜在的问题。这些技能对于开发和调试软件都是至关重要的，能够帮助我们提高代码质量和效率。