**第一次操作系统实验报告**

学号：58119304

姓名：朱启鹏

报告日期：2021/04/03

1. **实验目的**

简单了解Linux开发系统的编译环境，熟悉编译系统常见的命令，并学会在Linux开发系统编译简单的c++程序

1. **实验内容**

1. 虚拟机下载和安装 VMware or VirtualBox

2. Linux 安装，系统的启动、登录和停止

3.简单的系统命令ls、cd、pwd、cp、rm、mkdir、rmdirmv、cat、more、find

df、fsck

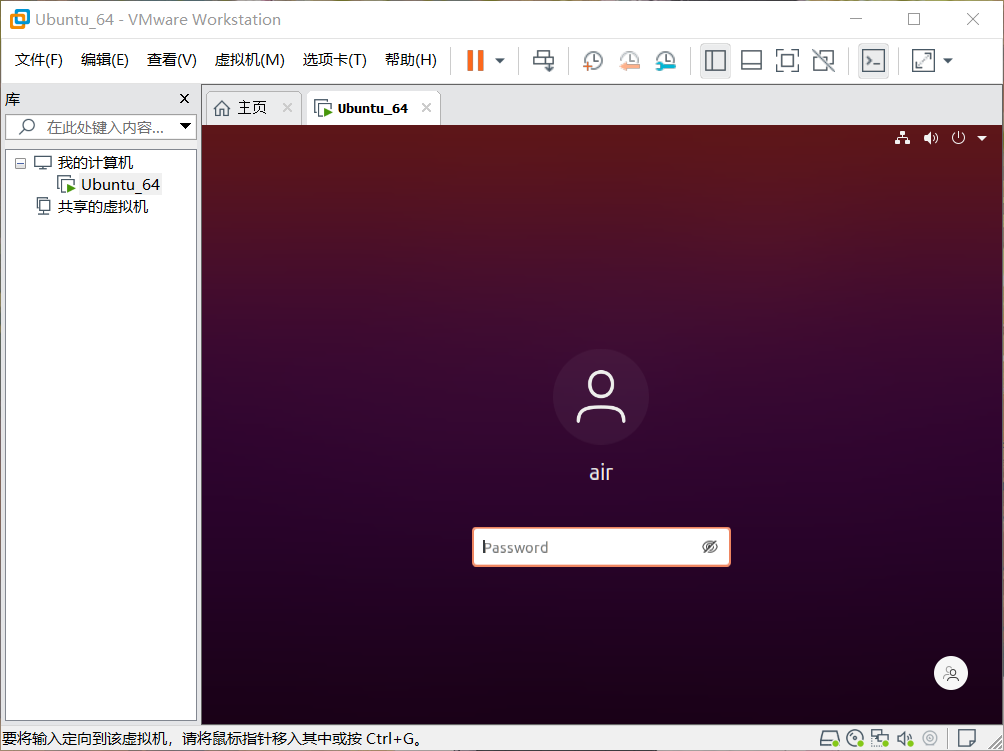
4.文件系统的shell命令

5.源程序的编辑、编译和链接Windows / Linux环境下的进程启动、察看、中止(正常、异常)

**三、实验过程**

**1.安装软件**

此次采用的安装源是VMware Workstation 8.0.4正式版(汉化包+序列号)下载地址： http://www.linuxidc.com/Linux/2012-11/74433.htm，使用的镜像为\ubuntu-20.04.2.0-desktop-amd64.iso标准安装版。



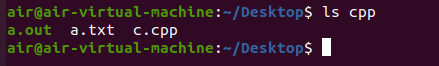
**2.常用命令**

**2.1 ls**（英文全拼：list files）命令用于显示指定工作目录下之内容（列出目前工作目录所含之文件及子目录)。

语法

ls [-alrtAFR] [name...]

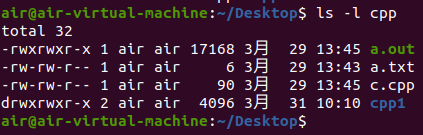
参数 :



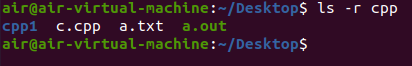
-a 显示所有文件及目录 (. 开头的隐藏文件也会列出)



-l 除文件名称外，亦将文件型态、权限、拥有者、文件大小等资讯详细列出



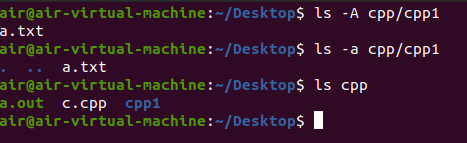
-r 将文件以相反次序显示(原定依英文字母次序)



-t 将文件依建立时间之先后次序列出

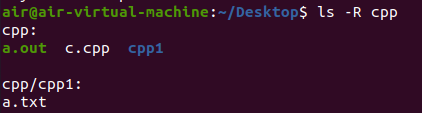


-A 同 -a ，但不列出 "." (目前目录) 及 ".." (父目录)



-F 在列出的文件名称后加一符号；例如可执行档则加 "\*", 目录则加 "/"

-R 若目录下有文件，则以下之文件亦皆依序列出



**2.2 cd**（英文全拼：change directory）命令用于切换当前工作目录。

其中 dirName 表示法可为绝对路径或相对路径。若目录名称省略，则变换至使用者的 home 目录 (也就是刚 login 时所在的目录)。

另外，~ 也表示为 home 目录 的意思， . 则是表示目前所在的目录， .. 则表示目前目录位置的上一层目录。

语法

cd [dirName]

dirName：要切换的目标目录。

实例

（1）cd 进入用户主目录

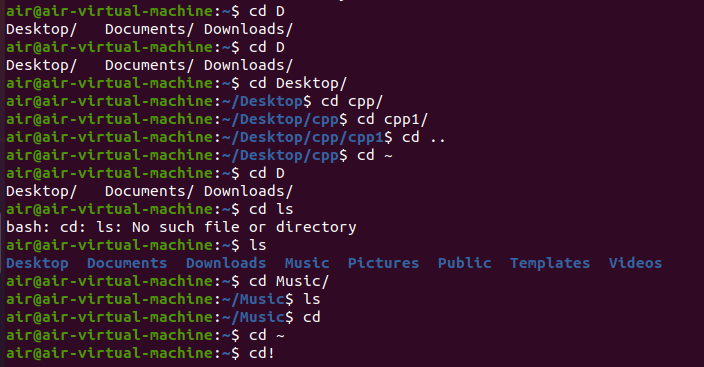
（2）cd ~ 进入用户主目录

（3）cd - 返回进入此目录之前所在目录

（4）cd .. 返回上一级目录

（5）cd ../..返回上两级目录

（6）cd !$ 把上个命令的参数作为cd 参数使用



**2.3 pwd**（英文全拼：print work directory） 命令用于显示工作目录。

执行 pwd 指令可立刻得知您目前所在的工作目录的绝对路径名称。

语法

pwd [选项]

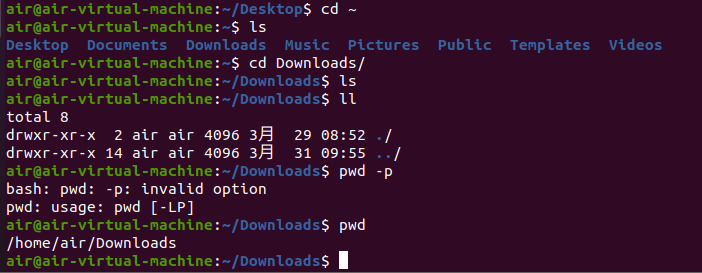
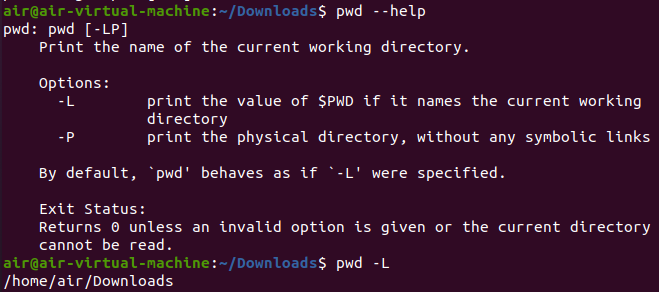
参数：

-L：--logical，显示当前的路径，有连接文件时，直接显示连接文件的路径， (不加参数时默认此方式)，参考示例1。

-p：--physical，显示当前的路径，有连接文件时，不使用连接路径，直接显 示连接文件所指向的文件，参考示例2。 当包含多层连接文件时，显示连接 文件最终指向的文件，参考示例3。

--help：显示帮助信息。

--version：显示版本信息。

**2.4 cp**用于复制文件或目录

cp命令语法

cp [options] sourcedir destdir

参数说明：

-a：此选项通常在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容。其作用等于dpR参数组合。

-d：复制时保留链接。这里所说的链接相当于Windows系统中的快捷方式。

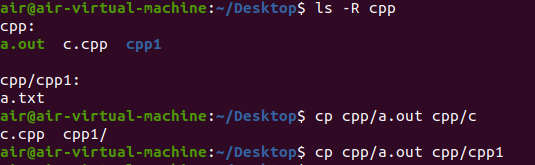
-f：覆盖已经存在的目标文件而不给出提示。

-i：与-f选项相反，在覆盖目标文件之前给出提示，要求用户确认是否覆盖，回答"y"时目标文件将被覆盖。

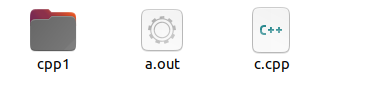
-p：除复制文件的内容外，还把修改时间和访问权限也复制到新文件中。

-r：若给出的源文件是一个目录文件，此时将复制该目录下所有的子目录和文件。

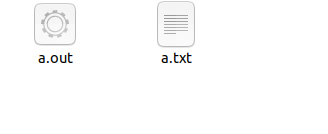
-l：不复制文件，只是生成链接文件。

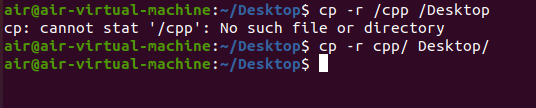


in cpp



in cpp1





**2.5 rm**（英文全拼：remove）命令用于删除一个文件或者目录。



语法

rm [options] name...

参数：

-i 删除前逐一询问确认。

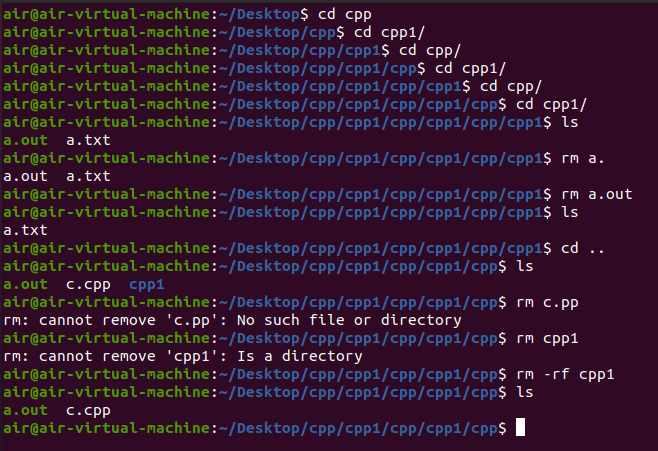
-f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。

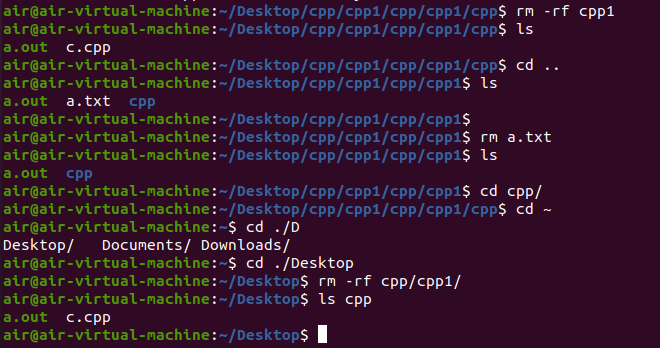
-r 将目录及以下之档案亦逐一删除。

-rf将会删除/opt/svn/目录以及其下所有文件夹，包括文件

实例

删除文件可以直接使用rm命令，若删除目录则必须配合选项"-r"，例如：





**2.6 mkdir**

mkdir 命令的功能是创建一个或多个新的目录

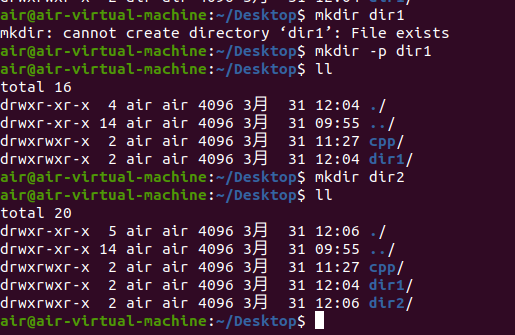
语法：

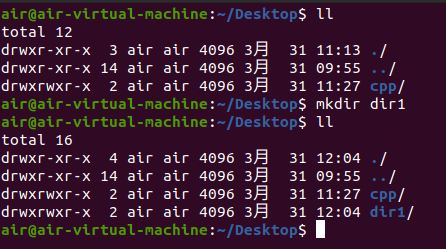
mkdir [option] path

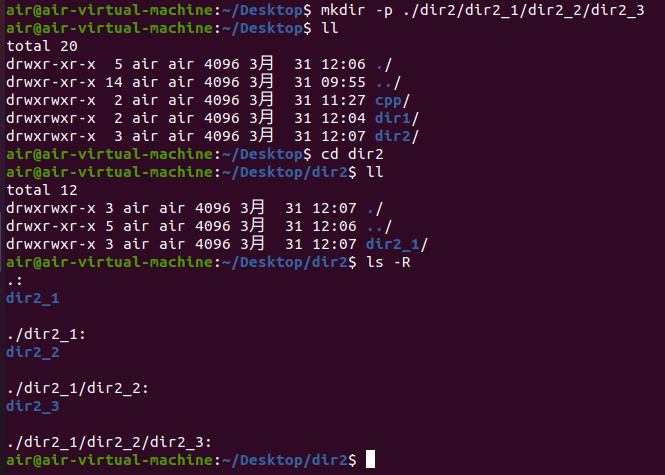
参数：

-m: 对新新建目录设置权限。

-p: 可以是一个路径名称。此时若路径的某一级目录尚不存在，使有该选项后系统会自动建立这些目录，即一次性建立多级目录







**2.7 rmdir命令**

该命令从一个目录中删除一个或多个子目录项，删除某目录时也必须具有对父目录的写权限。

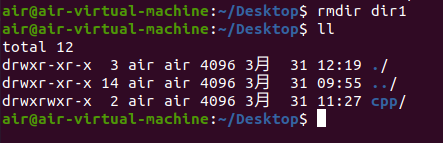
命令格式：

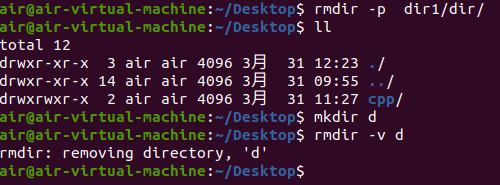
rmdir [选项]... 目录...

命令参数：

- p 递归删除目录dirname，当子目录删除后其父目录为空时，也一同被删除。如果整个路径被删除或者由于某种原因保留部分路径，则系统在标准输出上显示相应的信息。

-v, --verbose 显示指令执行过程





**2.8 mv移动命令**

语法 : mv [选项]… [-T] 源文件 目标文件

　或：mv [选项]… 源文件… 目录

　或：mv [选项]… -t 目录 源文件…

--backup[=CONTROL] 为每个已存在的目标文件创建备份

-b 类似--backup 但不接受参数

-f, --force 覆盖前不询问

-i, --interactive 覆盖前询问

-n, –nechoo-clobber 不覆盖已存在文件 如果您指定了-i、-f、-n 中的多个，仅最后一个生效。

--strip-trailing-slashes 去掉每个源文件参数尾部的斜线

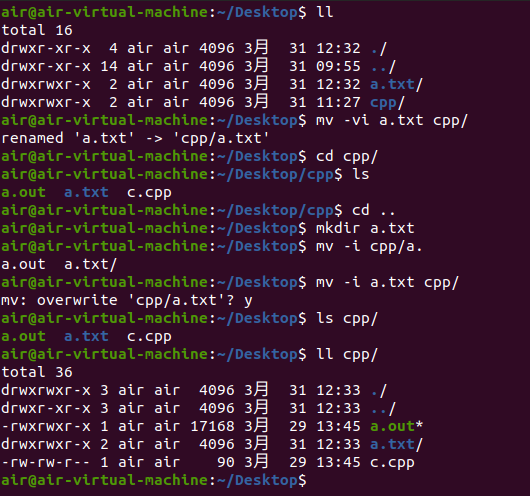
-S, --suffix=SUFFIX 替换常用的备份文件后缀

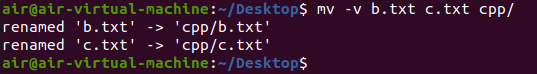
-t, --target-directory=DIRECTORY 将所有参数指定的源文件或目录 移动至 指定目录

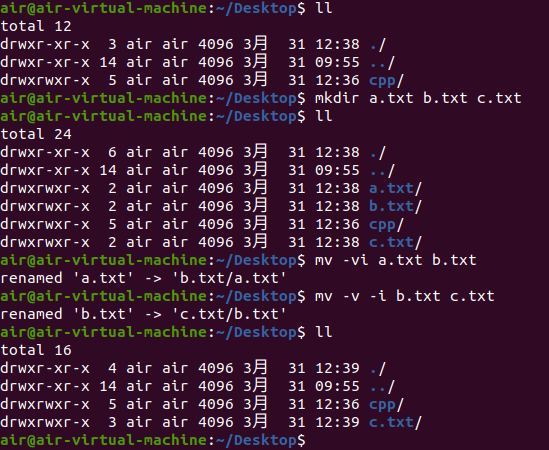
-T, --no-target-directory 将目标文件视作普通文件处理

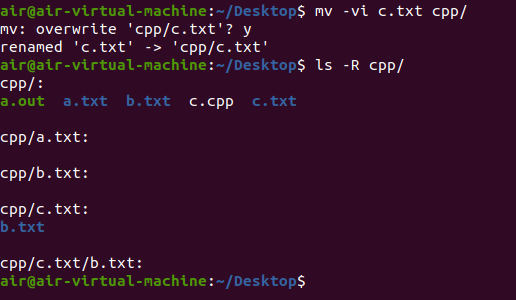
-u, --update 只在源文件文件比目标文件新，或目标文件不存在时才进行移动

-v, --verbose 详细显示进行的步骤









**2.9 cat**

linux下的一个文本输出命令，通常是用于观看某个文件的内容的；

cat主要有三大功能：

1.一次显示整个文件。

$ cat   filename

2.从键盘创建一个文件。

$ cat  >  filename

只能创建新文件,不能编辑已有文件.

3.将几个文件合并为一个文件。

$cat   file1   file2  > file

cat具体命令格式为 : cat [-AbeEnstTuv] [--help] [--version] fileName

说明：把档案串连接后传到基本输出(屏幕或加 > fileName 到另一个档案)

参数：

-n 或 –number 由 1 开始对所有输出的行数编号

-b 或 –number-nonblank 和 -n 相似，只不过对于空白行不编号

-s 或 –squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行

-v 或 –show-nonprinting

范例：

cat -n linuxfile1 > linuxfile2 把 linuxfile1 的档案内容加上行号后输入 linuxfile2 这个档案里

cat -b linuxfile1 linuxfile2 >> linuxfile3 把 linuxfile1 和 linuxfile2 的档案内容加上行号(空白行不加)之后将内容附加到linuxfile3 里。

范例：

把 linuxfile1 的档案内容加上行号后输入 linuxfile2 这个档案里

cat -n linuxfile1 > linuxfile2

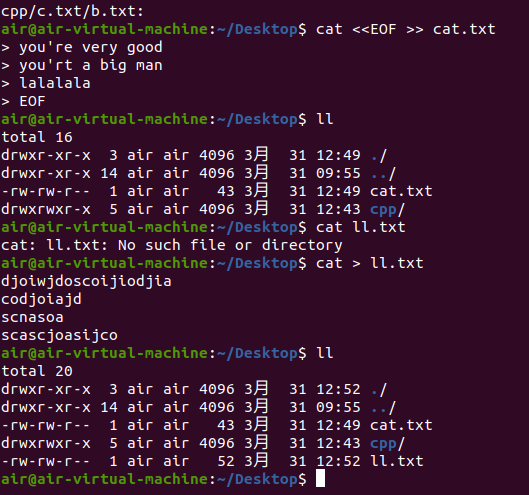
把 linuxfile1 和 linuxfile2 的档案内容加上行号(空白行不加)之后将内容附加到 linuxfile3 里。

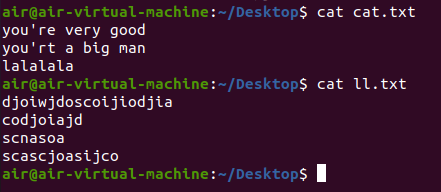
cat -b linuxfile1 linuxfile2 >> linuxfile3

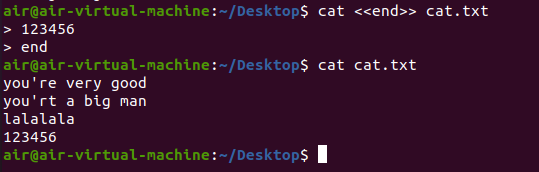
cat /dev/null > /etc/test.txt 此为清空/etc/test.txt档案内容

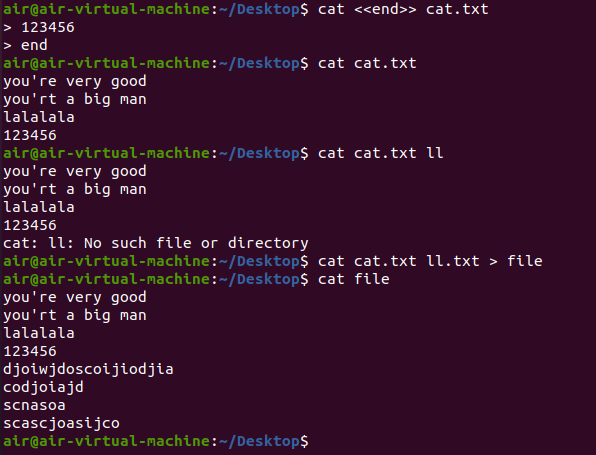
在linux shell脚本中我们经常见到类似于cat << EOF的语句，不熟悉的童鞋可能觉得很奇怪：EOF好像是文件的结束符，用在这里起到什么作用？

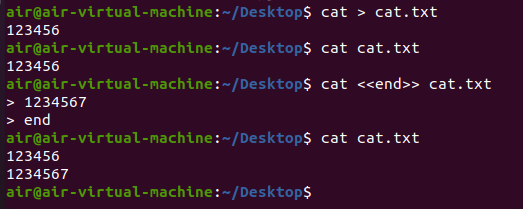
EOF是“end of file”，表示文本结束符。

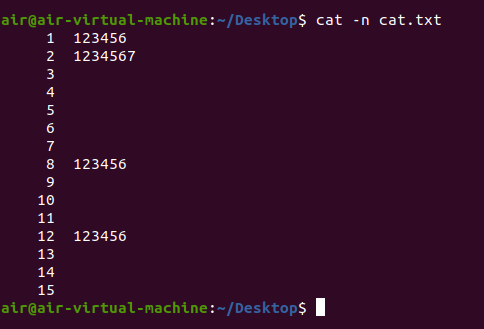
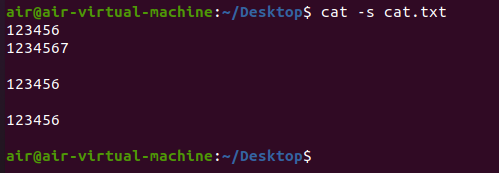












**2.10 more 命令**

功能类似 cat ，cat 命令是将整个文件的内容从上到下显示在屏幕上。 more 命令会一页一页的显示，方便使用者逐页阅读，而最基本的指令就是按空白键（space）往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能 。more 命令从前向后读取文件，因此在启动时就加载整个文件。

（1）命令格式

more [选项] 文件

（2）常用参数

参数 描述

+n 从笫 n 行开始显示

-n 定义屏幕大小为 n 行

+/pattern 在每个档案显示前搜寻该字串（pattern），然后从该字串前两行之后开始显示

-c 从顶部清屏，然后显示

-d 提示“Press space to continue，’q’ to quiet”，禁用响铃功能

-p 通过清除窗口而不是滚屏来对文件进行换页，与-c 选项相似

-s 把连续的多个空行显示为一行

-u 把文件内容中的下画线去掉

（3）常用操作

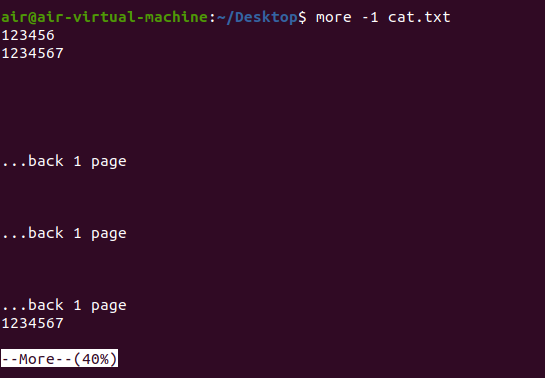
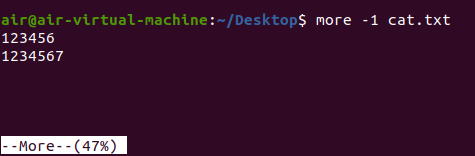
符号 描述

= 输出当前行的行号

q 退出 more

空格键 向下滚动一屏

b 返回上一屏



**2.11 find命令详解**

find命令格式：

find   path  -option  【 -print 】  【 -exec   -ok   |xargs  |grep  】 【  command  {} \;  】

find命令的参数：

1）path：要查找的目录路径。

      ~ 表示$HOME目录

       . 表示当前目录

       / 表示根目录

2）print：表示将结果输出到标准输出。

3）exec：对匹配的文件执行该参数所给出的shell命令。

      形式为command {} \;，注意{}与\;之间有空格

4）ok：与exec作用相同，

      区别在于，在执行命令之前，都会给出提示，让用户确认是否执行

5）|xargs  与exec作用相同 ，起承接作用

区别在于 |xargs 主要用于承接删除操作 ，而 -exec 都可用 如复制、移动、重命名等

6）options ：表示查找方式

options常用的有下选项：

-name   filename               #查找名为filename的文件

-perm                                #按执行权限来查找

-user    username             #按文件属主来查找

-group groupname            #按组来查找

-mtime   -n +n                   #按文件更改时间来查找文件，-n指n天以内，+n指n天以前

-atime    -n +n                   #按文件访问时间来查找文件，-n指n天以内，+n指n天以前

-ctime    -n +n                  #按文件创建时间来查找文件，-n指n天以内，+n指n天以前

-nogroup                          #查无有效属组的文件，即文件的属组在/etc/groups中不存在

-nouser                            #查无有效属主的文件，即文件的属主在/etc/passwd中不存

-type    b/d/c/p/l/f             #查是块设备、目录、字符设备、管道、符号链接、普通文件

-size      n[c]                    #查长度为n块[或n字节]的文件

-mount                            #查文件时不跨越文件系统mount点

-follow                            #如果遇到符号链接文件，就跟踪链接所指的文件

-prune                            #忽略某个目录

1、按名字查找

在当前目录及子目录中，查找大写字母开头的txt文件

$ find . -name '[A-Z]\*.txt' -print

在/etc及其子目录中，查找host开头的文件

$ find /etc -name 'host\*' -print

在$HOME目录及其子目录中，查找所有文件

$ find ~ -name '\*' -print

在当前目录及子目录中，查找不是out开头的txt文件

$ find . -name "out\*" -prune -o -name "\*.txt" -print

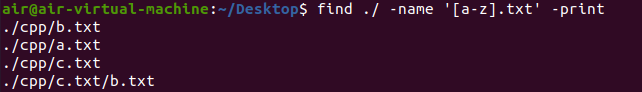
2、按目录查找

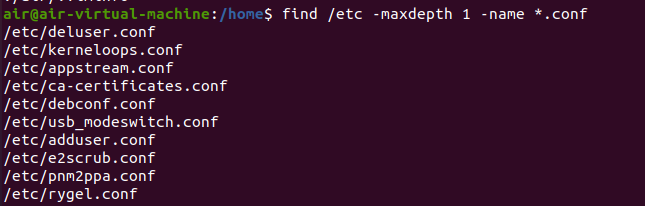
在当前目录除aa之外的子目录内搜索 txt文件

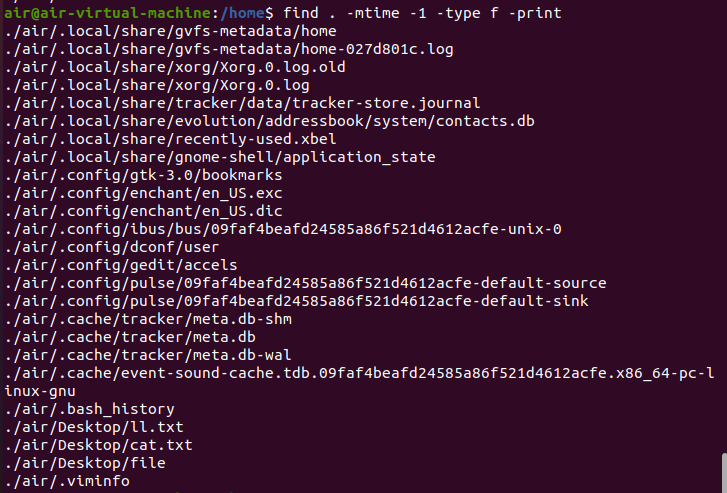
$ find . -path "./aa" -prune -o -name "\*.txt" -print

在当前目录及除aa和bb之外的子目录中查找txt文件

$ find . −path′./dir0′−o−path′./dir1′−path′./dir0′−o−path′./dir1′ -a -prune -o -name '\*.txt' -print







**2.12 df命令概述**

df命令作用是列出文件系统的整体磁盘空间使用情况。可以用来查看磁盘已被使用多少空间和还剩余多少空间。

df命令显示系统中包含每个文件名参数的磁盘使用情况，如果没有文件名参数，则显示所有当前已挂载文件系统的磁盘空间使用情况，参考示例1。

在默认情况下，磁盘空间是以1KB为单位进行显示的，但是，如果POSIXLY\_CORRECT环境变量被设置为true，这种情况下默认使用512字节为单位显示。

df命令语法

df [选项] [文件名]

参数：

-a：--all，显示所有的文件系统，包括虚拟文件系统，参考示例2。

-B：--block-size，指定单位大小。比如1k，1m等，参考示例3。

-h：--human-readable，以人们易读的GB、MB、KB等格式显示，参考示例4。

-H：--si，和-h参数一样，但是不是以1024，而是1000，即1k=1000，而不是1k=1024。

-i：--inodes，不用硬盘容量，而是以inode的数量来显示，参考示例5。

-k：以KB的容量显示各文件系统，相当于--block-size=1k。

-m：以KB的容量显示各文件系统，相当于--block-size=1m。

-l：--local，只显示本地文件系统。

--no-sync：在统计使用信息之前不调用sync命令(默认)。

-sync：在统计使用信息之前调用sync命令。

-P：--portability，使用POSIX格式显示，参考示例6。

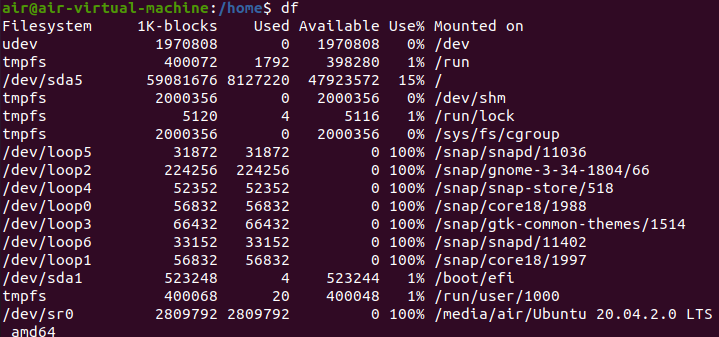
-t：--type=TYPE，只显示指定类型的文件系统，参考示例7。

-T：--print-type，显示文件系统类型，参考示例8。

-x：--exclude-type=TYPE，不显示指定类型的文件系统。

--help：显示帮助信息。

--version：显示版本信息。



输出结果列说明：

Filesystem：代表该文件系统时哪个分区，所以列出的是设备名称。

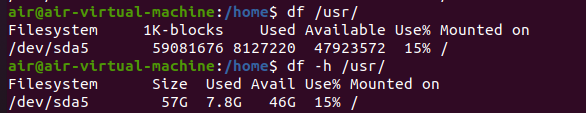
1K-blocks：说明下面的数字单位是1KB，可利用-h或-m来改变单位大小，也可以用-B来设置。

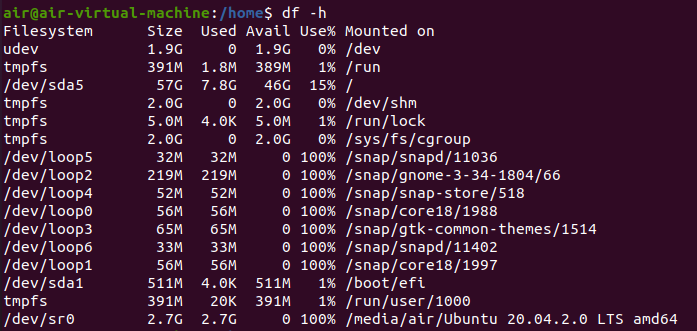
Used：已经使用的空间大小。

Available：剩余的空间大小。

Use%：磁盘使用率。如果使用率在90%以上时，就需要注意了，避免磁盘容量不足出现系统问题，尤其是对于文件内容增加较快的情况(如/home、/var/spool/mail等)。

Mounted on：磁盘挂载的目录，即该磁盘挂载到了哪个目录下面。





**2.13 fsck修复文件系统**

fsck（file system check）用来检查和维护不一致的文件系统。若系统掉电或磁盘发生问题，可利用fsck命令对文件系统进行检查。

fsck常见命令参数

-a：自动修复文件系统，不询问任何问题；

-A：依照/etc/fstab配置文件的内容，检查文件内所列的全部文件系统；

-N：不执行指令，仅列出实际执行会进行的动作；

-P：当搭配"-A"参数使用时，则会同时检查所有的文件系统；

-r：采用互动模式，在执行修复时询问问题，让用户得以确认并决定处理方式

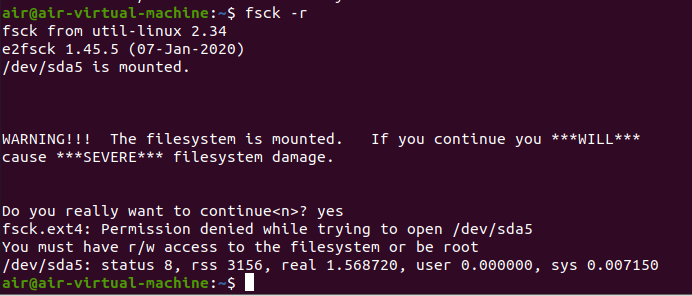
-R：当搭配"-A"参数使用时，则会略过/目录的文件系统不予检查；

-s：依序执行检查作业，而非同时执行；

-t<文件系统类型>：指定要检查的文件系统类型；

-T：执行fsck指令时，不显示标题信息；

-V：显示指令执行过程。



**2.14创建文件**

格式：touch文件名

范例：

1.touch newfile

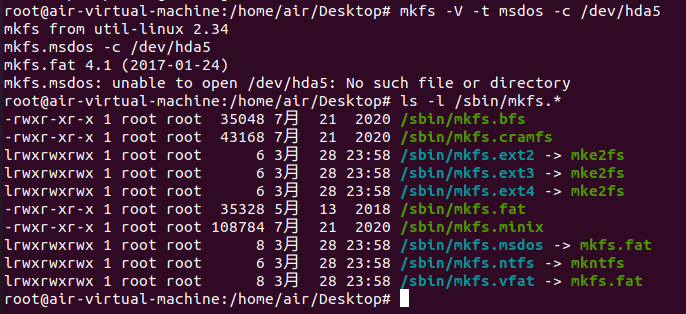
创建一个名字为“newfile”的空白文件

1. **Linux 文件管理系统**

**3.1 Linux mkfs 命令**

Linux mkfs（英文全拼：make file system）命令用于在特定的分区上建立 linux 文件系统。

使用方式 :



mkfs [-V] [-t fstype] [fs-options] filesys [blocks]

参数 ：

device ： 预备检查的硬盘分区，例如：/dev/sda1

-V : 详细显示模式

-t : 给定档案系统的型式，Linux 的预设值为 ext2

-c : 在制做档案系统前，检查该partition 是否有坏轨

-l bad\_blocks\_file : 将有坏轨的block资料加到 bad\_blocks\_file 里面

block : 给定 block 的大小

实例

在 /dev/hda5 上建一个 msdos 的档案系统，同时检查是否有坏轨存在，并且将过程详细列出来 :

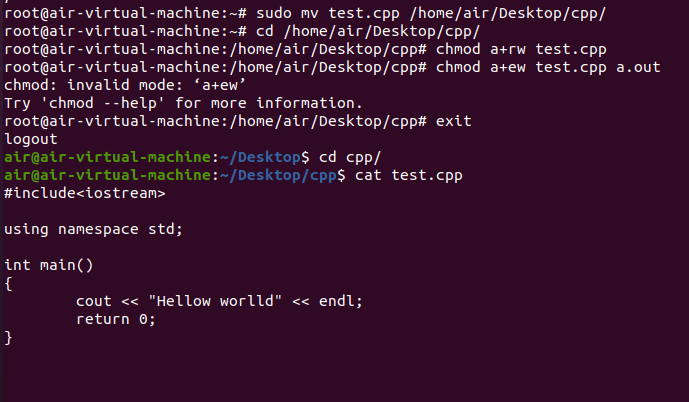
mkfs -V -t msdos -c /dev/hda5

将sda6分区格式化为ext3格式

mfks -t ext3 /dev/sda6

3.2 chmod（英文全拼：change mode）命令是控制用户对文件的权限的命令

Linux/Unix 的文件调用权限分为三级 : 文件所有者（Owner）、用户组（Group）、其它用户（Other Users）。



**3.2 Linux文件的分类**

在Linux系统中每一种文件的类型都用一个字符进行标识，主要有7种类型，可以简单的记忆为b、c、d、-、l、p、s每个字符所代表的的含义如下

**3.3 Linux目录结构**

在Linux学习（一）当中我们说过Linux有一个特点是，一切皆文件，所以目录也是一类特殊的文件。利用目录可以构成文件系统的分层属性结构。

Linux文件系统采用带链接的属性目录结构，即只有一个根目录，通常用“/”来表示，其中含有下级子目录或文件的信息；子目录中又可含有更下级的子目录或者文件的信息。这样一层一层延伸下去，构成一棵倒置的树，如下图

Linux系统的每个目录都有不同的功能

1. **vim编写C++程序过程(以hello world为例)：**

1、hello.cpp。

hello.cpp文件，并进入vim界面;

2、键入i，进入输入模式。

3、编写hello.cpp程序。

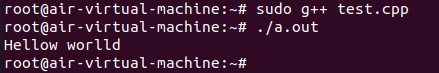
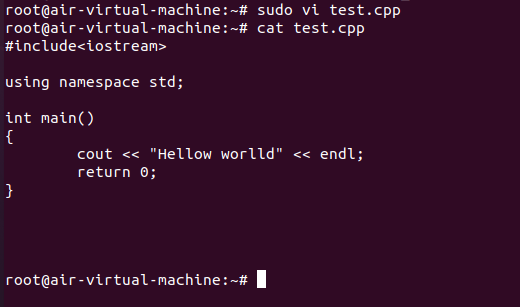
4、先按Esc键，退出输入模式；然后，按Shift+:键，进入命令模式。输入wq，保存并退出。

5、编译hello.cpp。

  5.2、对于C++程序：用g++ hello.cpp可编译。编译后，均生成a.out的可执行文件;

6、键入指令"./a.out"，运行程序。

7、在Terminal上显示"hello world"即成功



**四、实验体会**

由于对VMware不够熟练，在安装过程中出现许多问题，在卸载与安装多次后才成功。后续又遇到root的问题，后来仔细观察终端才发现，Ubunto把root的密码与普通用户密码统一了起来。总之实验过程十分坎坷，但是最终还是完成了任务。

**实验二**

1. **实验目的**

能够简单了解系统调用

1. **实验内容**

使用系统调用,用C或C++写一个程序,实现如下功能:从一个这件中读出数据，写入一个文件中。

要求:

具有良好的交互性

使用者可输入源文件和目的文件的路径和文件名。具有完善的语误处理机制

针对可能出现的各种错误,要有相应的错误提示输出,并作相应处理。

在Linux操作系统上调试并运行实验目的:

通过实验，加深对系统调用概念的理解,了解其实现机制以及使用方式。

通过在Linux操作系统上编写和词试简单程序,进一步熟悉Linux真,为近一步理解和学Linux,作系统的内核结构和核心机制作操作系维的使用:初步掌握linux环境下的C或C++编译和调试做准备。

