##### date 指令

//这个指令能够获取当前的各种时间

1. 直接使用date会获得一些常见的时间



1. date 后面跟上+，然后在用%就可以获取具体的时间 //只用来获取当前的小时，但不会有多少分。 ps一定要带上 + 这个符号

//下面的这些是大写就是大写，是小写就是小写

* 1. date +%H ：获取小时 //hour：小时



* 1. date +%M ：获取分钟 //minutes： 分钟

//此时测试的时间是19:58分

* 1. +%S :获取秒钟 //seconds ：秒钟
  2. +%X ：同时获小时，分钟，秒钟
  3. +%d :获取日 //day：日
  4. +%m: 获取月
  5. +%Y ：获取当前的年份
  6. +%F ：相当于获取当前的年月日

1. 上面的这些指令还是可以任意的组合搭配，但是格式是这样的 date +%M:%S:%d 等等组合，其中这个：符号可以换成任意的符号，这个：主要的作用就是用于间隔这些指令输出的内容
2. +%s ：时间戳 ：也就是1970一月一日到打出指令的那一刻经过了多少秒



//ps:使用date -d @时间戳 可以将时间戳的那一串数字转换成相应的日期 这个是查看时间戳的方法



//ps: 使用 date +%M:%S等 -d @时间戳 这样的格式就能以常见的形式来查看这个时间戳所代表的的具体的年份

1. date -s //设置当前时间，只有root权限才能设置，其他只能查看。

//date -s 20080523 //设置成20080523，这样会把具体时间设置成空00:00:00

//date -s 01:01:01 //设置具体时间，不会对日期做更改

//date -s “01:01:01 2008-05-23″ //这样可以设置全部时间

//date -s “01:01:01 20080523″ //这样可以设置全部时间

//date -s “2008-05-23 01:01:01″ //这样可以设置全部时间

//date -s “20080523 01:01:01″ //这样可以设置全部时间

##### cal 指令

//calendar ：日历 这个指令是和日历相关的指令

1. cal 2023 :cal后面跟上一个年份，会把这个年份上面的所有日期都显示出来



1. 只使用 cal ，只会显示出当前的月份

3.如果只想获取指定的月份，可以这样使用 cal 月份 年份

##### find 指令

//find：找到；查找

//find 可以在路径下进行文件的搜索

1. find 目录名字 -name 文件名字 ：这个指令的作用就是在特定的目录下面进行某个文件的查找。 注意这个name一定要加上



//ps：目录的名字也可以是系统目录/usr/bin里面的 这个目录是和home同级别的

//当然有可能级别不够查询不到，或者无法查询

//find是在磁盘文件中进行查找的， 另外find的查询命令分支选项有很多，用的时候继续搜索

//上面的第一种这样的使用方式就是在特定的目录下查找特定的文件

1. 这个搜索还是可以搭配前面的知识点 \* 来进行模糊搜索，例如 ：find ~ -name \*.txt 这条指令的目的是搜索所有.txt结尾的文件  
   
2. which 可以看作是在特定目录（指令文件存放的地方 /user/bin）进行指令的搜索 find在这个方面两个是差不多的。

##### whereis 指令

//这个指令类似于模糊搜索，只要带上搜索的关键字，无论是文件还是压缩包或者是其他的都能查询出来



这里只搜索到了两个内容 ps：当然也可以在特定的路径下面进行查找，先去搜索指定路径的名字，比如文件夹名字，再从这个文件夹中进行搜索。

1. 这个指令是综合了which 和 find这两个指令。 只想查找指令用which，只想查找文件用find

whereis不能用于自己指定的路径下面的查找

区别：which只能在指令路径下面查找指令文件 whereis只能在系统特定的路径下查找，比如指令、可执行程序、手册、安装包等

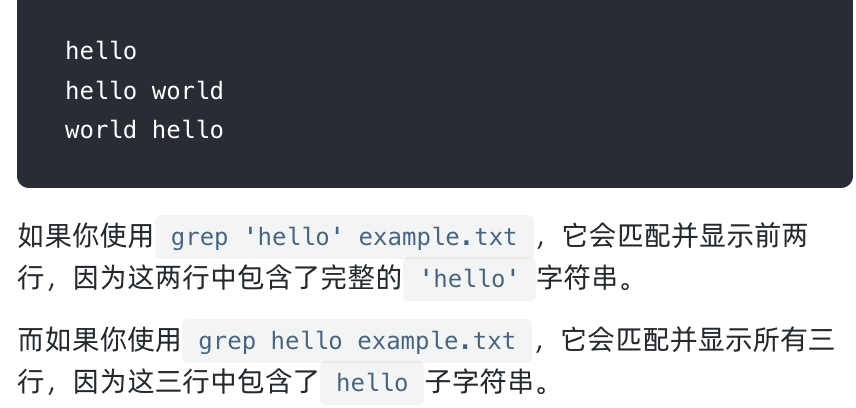
find可以在任意的自己定义的路径下面查找任意文件、指令等。 前提是需要绝对路径，另外普通用户在查找的时候还可能面临用户权限不足的情况

##### 五、grep 指令

//这个指令是一个行文本过滤的指令，通过关键字进行筛选你想要的某些行的内容

//Global Regular Expression Print"，意为全局正则表达式打印

补充： 在搜索一串字符串的时候，’字符串’ 字符串 ，添加单引号和不添加单引号是两种不同的搜索模式，也就是会将前面的看作一个整体。xxxxhelloxxxx,而不添加则是将子字符串也会搜索到。另外，空格隔开的会看做两个单独的字符串



1.使用 grep 指令对关键字进行搜索

//grep也是一个行文本过滤工具

2. grep -v

//这个选项能够输出不带关键字的行 使用的格式 grep -v ‘关键字’ 被搜索文件的名字

//另外 ’空格’ 这种字符串也是什么什么都能匹配上

3. grep -n

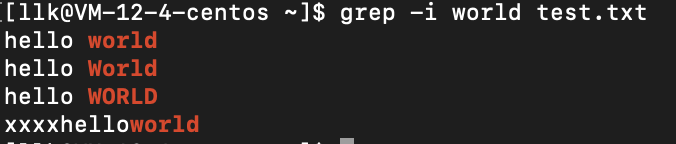
//这个选项是将匹配到的字符串所在的行显示出来



ps： 这两个选项还是能够组合起来去使用 grep -nv

4. grep -i

//这个选项对字符串的大小写忽略



5. 和管道文件进行配合

//如果不确定一个路径中都有哪些文件包含关键字，可以先将这些文件过滤出来，再和grep搭配使用

//tar命令进行打包和压缩

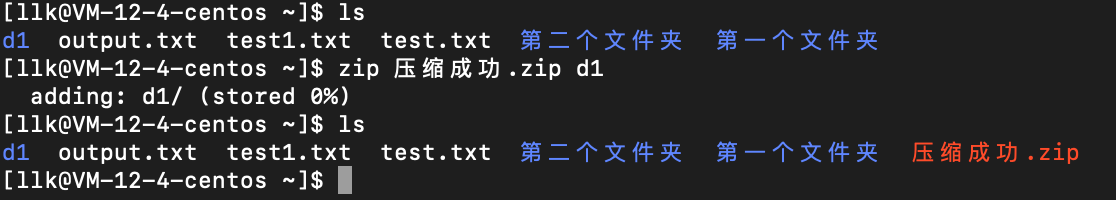
##### 六、zip和unzip指令

//这两个指令用于打包压缩和解压缩

1. zip -r指令

//这个指令能够将文件夹进行压缩，使用的格式为 zip xxx.zip 想要压缩的文件夹路径 其中需要注意的是，这个指令必须配合上-r，否则只会讲d1这个文件夹进行压缩，压缩后的文件解压后里面并没有内容

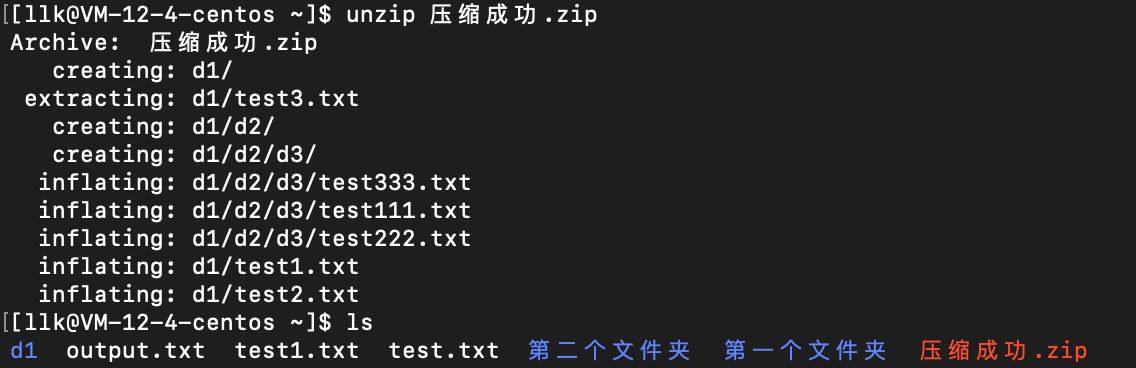
//也就是说只有带上-r才能将原来文件夹里面的内容进行压缩



//当然，压缩的时候也能够选择多个文件或者多个文件夹，或者文件夹+文件等 同时进行压缩，格式和上面的类似 zip xxx.zip 文件名 文件夹路径。。。。。。

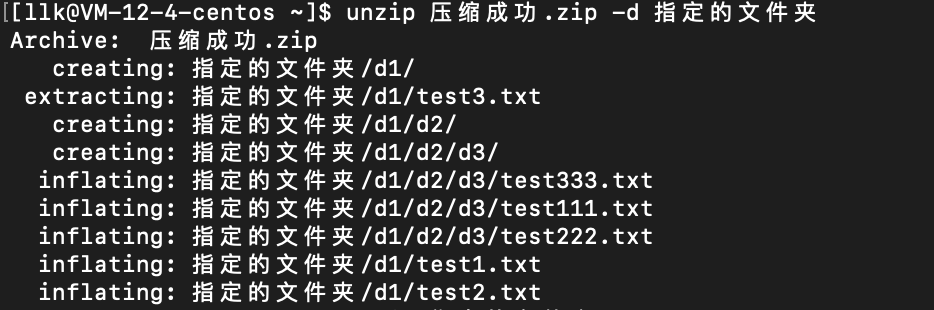
2. unzip指令

//这个指令能够将压缩文件进行解压缩 使用格式就是 unzip xxx.zip

//在解压之前已经将d1进行删除了，图片中的d1是被解压出来的

3. unzip -d

//使用的格式必须是 unzip xxx.zip -d 指定的路径

//这个指令能将解压后的内容放入指定的文件夹

//ps: 上面两个命令没有安装可以在root用户下使用 yum install -y zip unzip 可以是用 su+root用户密码，就能切换到root用户

//安装过程中 -y是用来默认yes的。安装过程中不需要再去手动确认了。

##### 七、tar 命令

//打包和压缩的选项是非常多的，也就是说压缩包的后缀和解压缩的方式是有很多种类的 这里使用的是常见的zip 和tar的方案

1. tar

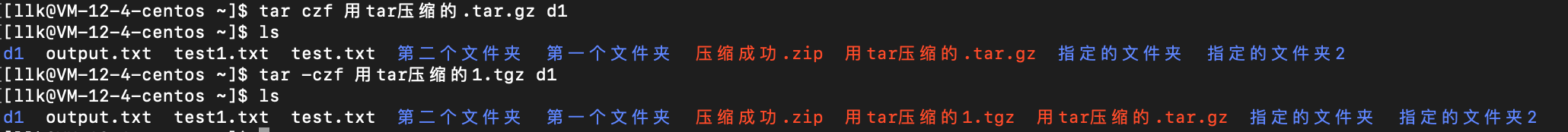
//这个命令的使用格式 tar -czf 请注意czf有的系统是使用-czf的，有的系统是直接使用不用带上-

//crf 中的c是create（创造的意思） z是zip压缩格式 f是file创建一个文件夹

//其中压缩文件也可以写成 code.tar.gz

（1）

//这两个创还能出来的效果是一样的



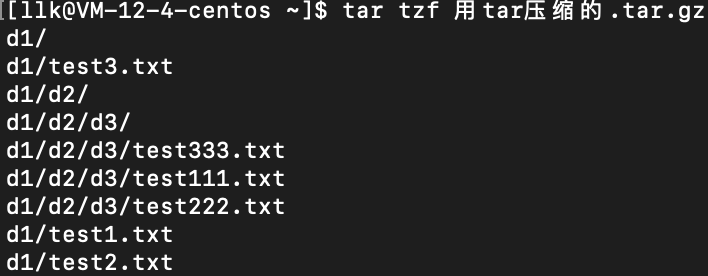
//tar是打包后缀 gz是压缩后缀

2.各种选项

//tar打包和压缩的重点在于czf中的c位置字母的替换 c代表create创造的意思

（1）tzf

//这个选项使用来查看压缩文件中的文件的，只是用来查看压缩文件里面有什么 t代表list，列表



（2）xzf

//这个选项是用来解压tgz后缀的压缩包的，使用方式和上面的一样

（3）-C

//能够解压的文件放在指定的路径下面，具体使用格式如图



##### bc命令

//这个其实就是Linux下面的计算器。 输入quit退出

1.可以和管道配合，也就是说不用去调用也能将计算的记过进行输出



##### uname命令

//这个命令能查看当前系统的信息

//关于系统方面的指令补充一点，可以使用free来查询当前系统空闲的内存，使用-g选项是以g为单位，使用-k选项，就是以k为单位等

1. uname -a
2. uname -r //这个命令是用来查体系结构的



##### 常用的几个热键

1. tab //指令补全快捷

//这个按键是用来快速补齐将要输入的指令，比如想要输入which，可以输入whic，此时再按一下tab，就能够自动补全which

//如果只输入一个首字母，再次按下这个tab（有可能是连续按动两次），就能将所有的w开头的指令进行补全

1. ctrl+r //检索输入过的指令

//按下这个按键会出现单引号，输入相应的指令，比如find，就会出现最后一次使用find的指令



1. 如果不确定某些指令有没有安装，可以使用 xxxx --version 来进行查看，如果查看会有相应的提示
2. ctrl+d 也就是退出键，退出文件，用户，甚至xshell

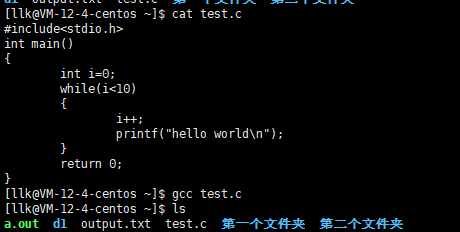
##### 用nano进行c语言代码的运行

1. 在linux中，向上的尖括号代表着ctrl



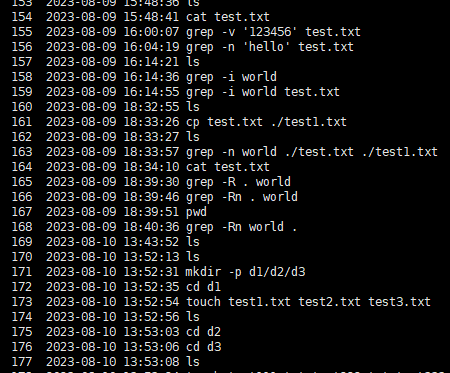
1. 在里面书写完c语言的代码以后，可以使用ctrl+x 提示后按y进行保存。

//使用cat进行输出，然后使用gcc进行编译生成可执行程序xxx.out文件如下图所示



//使用 ./xxx.out 这种格式就可以运行.out这种文件了  
如果不能运行使用chmod命令为其添加执行权限。例如：chmod +x myprogram.out。

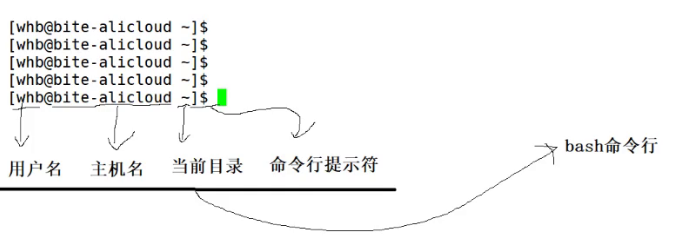
##### history指令

//输入history以后能够查看近期输入的指令  
//这个是截取了部分。默认搜索1000条，这个默认值可以修改

##### shutdown指令、

1. shutdown -h 这个指令能够将计算机休眠
2. shutdown -s 这个指令能够关闭计算机
3. shutdown -t 这个指令能够设定关机的时间，比如100秒后，这个单位是秒，-t 后面跟随的是数字

##### 指令的运行原理

1. 首先看bash命令行的组成  
   

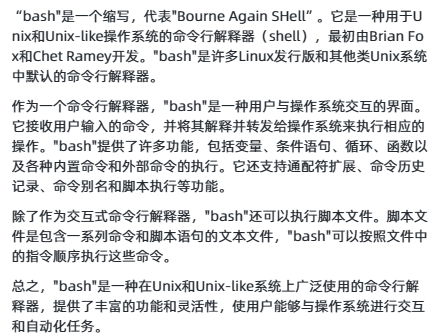
//输入指令的过程，就是输入字符串

//指令的本质就是编译好的程序和脚本，并且一定会在系统的特殊路径下面存放

//我们所有的指令，最终都要在os（操作系统）内部运行，但是os使用难度比较高，我们用户不能直接和os打交道。所以我们需要一个图形化界面或者命令行解释器

//命令行解释器：将使用者的命令翻译给核心（kernel）处理，同时，将核心的处理结果翻译给使用者

//也就是说用户在命令行指示器上面输入指令，命令行指示器再转换成相应的指令给操作系统，操作系统将相应的操作

1. 
2. 命令行解释器存在的意义
3. 进行命令行解释
4. 变相的保护了操作系统，对于用户的非法请求直接拦截
5. bash/sh shell 命令行 命令行解释器 等的区别是什么
6. 命令行解释器就是一个统称
7. 在linux下面就叫做命令行，也叫做bash，sh，shell bash sh这两个是具体的解释器的种类 shell是命令行解释器的统称
8. 在win系统下就是图形化界面，

##### 十三、

##### 十三、

##### 十三、

##### 十三、

##### 十三、