**Linux 文件系统**

Linux虚拟目录结构只包含一个称为根（root）目录的基础目录。路径本身并没有提供任何关于文件存放在哪个物理磁盘上的信息。例如：

/home/Rich/Documents/test.doc

**常见Linux目录名称**

/ 虚拟目录的根目录。通常不会在这里存储文件

/etc 系统配置文件目录

/home 主目录，Linux在这里创建用户目录

/root root用户的主目录，登录系统后的初始目录

/tmp 临时目录可以在该目录中创建和删除临时工作文件

/usr 用户二进制目录，大量用户级的GNU工具和数据文件都存储在这里

注意：在登录系统并获得一个shell CLI提示符后，会话将从主目录开始。

**切换目录命令 cd**

后面可以跟绝对路径和相对路径，绝对文件路径总是以正斜线（/）作为起始，指明虚拟文件系统的根目录。相对路径是以文件名或者一个特殊字符开始的。

**过滤输出列表 ls –l XXX?\***

当用户指定特定文件的名称作为过滤器时， ls 命令只会显示该文件的信息。如不知文件的确切名称时，可以使用通配符，并在过滤器中用它们进行模式匹配：

？代表一个字符；例如： ll mysql\_loc?l.sh

\*代表多个字符；例如：ll mysql\_\*

[abc]代表可能出现的字符；例如ls -l my\_scr[ai]pt ls -l f[a-i]ll

注意 ls -l文件夹/ 可以直接显示某个文件夹内容:

ls -l Scripts/

**cp 命令**需要两个参数——源对象和目标对象：

cp source destination

注意：如果目标文件已经存在， cp 命令可能并不会提醒这一点。最好是加上 -i 选项，强制shell询问是否需要覆盖已有文件。

cp -i test\_one test\_two

注意：将一个文件复制到一个目录下时，可以使用相对路径或者绝对路径，但是记得在目标目录后加正斜线/，表明是一个目录，否则会创建一个文件，而且没有提示：

cp -i test\_one /home/christine/Documents/

**cp 命令的 -R**

可以用它在一条命令中递归地复制整个目录的内容。

$ cp -R Scripts/ Mod\_Scripts

整个Scripts目录中的内容都被复制到Mod\_Scripts目录中(没有则新建)，注意Mod\_Scripts后没有正斜线，加上就成了复制到这个目录下并创建同名文件夹了。

**重命名文件称为移动moving**

mv 命令可以将文件和目录移动到另一个位置或重新命名

注意：mv 命令中使用 -i参数，覆盖已有文件时会提示。

**移除removing**

$ rm -i fall

rm: remove regular empty file 'fall'? y

注意：bash shell中没有回收站或垃圾箱，文件一旦删除，就无法再找回。如果要删除很多文件且不受提示符的打扰，可以用 -f 参数强制删除。小心为妙！

**创建目录**

创建一个目录：$ mkdir New\_Dir

创建一个多级目录，需要参数-p：mkdir -p New\_Dir/Sub\_Dir/Under\_Dir

注意：一口气删除目录及其所有内容的终极大法就是使用带有 -r 参数和 -f 参数的 rm 命令。

rm -rf Small\_Dir

**查看整个文件**

cat -n test1 显示文本文件中所有的数据，给所有行加上行号

more 分页显示文本文件的数据，空格或者回车逐行向前浏览，按q退出

less

**查看部分文件**

tail 默认显示文件最后10行的内容，-n来指定显示的行数，例如 $ tail -n 2 log\_file #只显示两行或者直接 -行数：$ tail - 2 log\_file

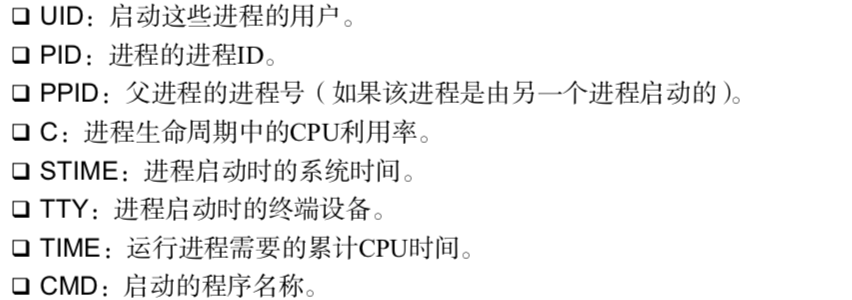
tail -f参数是tail命令的一个突出特性。它允许你在其他进程使用该文件时查看文件的内容。 tail命令会保持活动状态，并不断显示添加到文件中的内容。这是实时监测系统日志的绝妙方式。

head 默认显示文件前10行文本 ，同样支持-n 行数 或者-行数，不支持-f

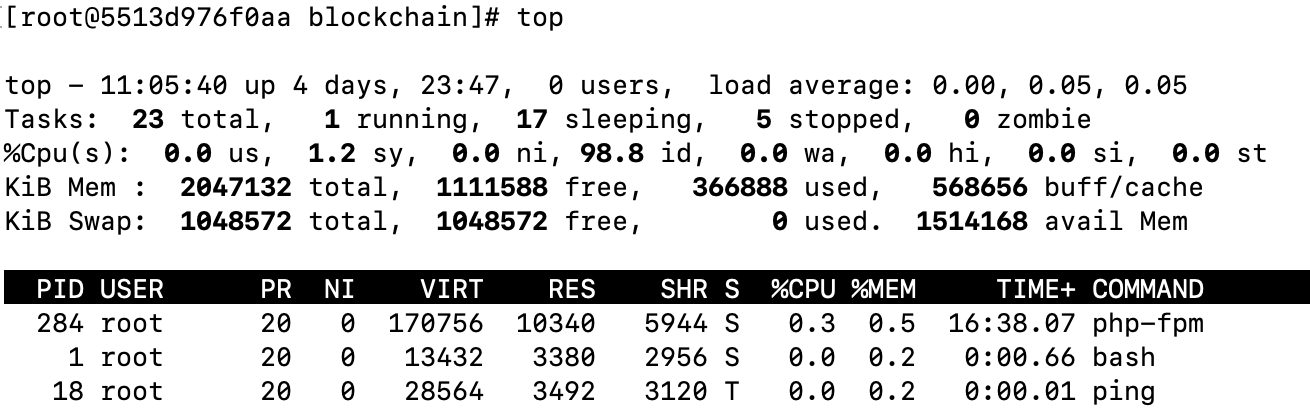
**进程**

ps命令默认情况下，只会显示运行在当前控制台下的属于当前用户的进程

ps -ef 查看系统上运行的所有进程



top命令跟ps命令相似，能够显示进程信息，但它是实时显示的，而ps只能显示某个特定时间点的信息



第一行显示了当前时间，系统运行时间，登录用户数，系统平均负载，平均负载有3个值:最近1分钟的、最近5分钟的和最近15分钟的平均负载。值越大说明系统 的负载越高。

注意：出现最近1分钟的高负载值也很常见，但如果近15分 钟内的平均负载都很高，就说明系统可能有问题。

进程列表中一些主要的列：

PR：进程优先级

S: 进程的状态(D代表可中断的休眠状态，R代表在运行状态，S代表休眠状态，T代表跟踪状态或停止状态，Z代表僵化状态)。

TIME+:自进程启动到目前为止的CPU时间总量。

COMMAND:进程所对应的命令行名称，也就是启动的程序名。

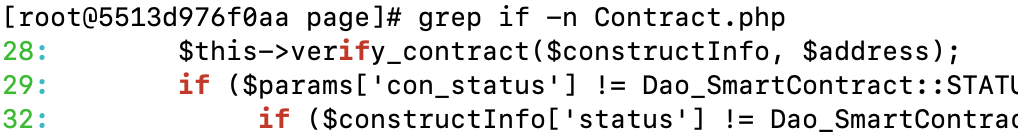
**搜索数据**

大文件中找一行数据，可以使用grep命令

grep [options] pattern [file] 输出的就是包含匹配模式的行

如果要显示匹配模式的行所在的行号，可加-n参数。

$ grep -n t file1



如果只要知道有多少行含有匹配的模式，可用-c参数。

$ grep -c t file1

tar命令的格式:

-c --create 创建一个新的tar归档文件

-x --extract 从已有tar归档文件中提取文件

-f file 输出结构到文件或设备

-v 在处理文件时显示文件

这些选项经常一起用，例如:

tar -cvf test.tar test/ test2/ #创建一个归档文件tar，含有test1和test2目录

tar -tf test.tar #列出tar包里的内容

tar -xvf test.tar #tar包提取内容

注意：.tgz结尾的文件时gzip压缩过的tar文件，可以使用命令 tar -zxvf 解压。

**shell 脚本文件**

在创建shell脚本文件时，必须在文件的第一行指定要使用的shell。其格式为: #!/bin/bash

注意：#号做注释用 #!告诉shell用哪个shell来运行脚本

**输出重定向**

最基本的重定向将命令的输出发送到一个文件中，可以使用重定向操作符（大于号>）实现：

含有输出的命令 > 要保存到的文件

例如：

$ date > test6

$ cat test6

Thu Feb 10 17:56:58 EDT 2014

如果保存的文件存在则覆盖，不存在新建

如果不想覆盖文件，可以使用双大于号>>追加数据。

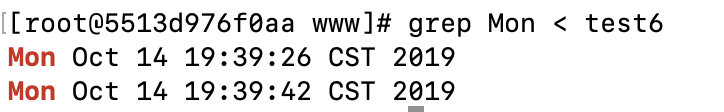


**输入重定向**

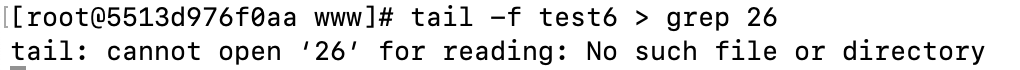
将文件的内容重定向到命令，使用小于号<

注意：剪头指向哪，就是把数据输出到哪

例如：把test6的文件内容输出给搜索命令grep



注意，输入输出重定向只能将命令执行结果输出到文件上，不能直接输出给命令，命令输出到命令可以使用管道



管道

如果将一个命令的输出作为另一个命令的输入，如果使用重定向，需要将命令1的结果重定向>文件上，再命令2输入重定向< 文件，比较繁琐，可以使用管道连接符 |：

# tail -f test6 | grep 26

Mon Oct 14 19:39:**26** CST 2019

管道放到命令中间，可以将一个命令的输出重定向到另一个命令中，第一个命令产生输出的同时，输出会立即传给第二个命令。

Ctrl+C组合键会生成SIGINT信号 ，停止shell中当前运行的进程 。

Ctrl+Z组合键会生成一个SIGTSTP信号，停止shell中运行的任何进程。

正则

锚字符

1锁定在行首，脱字符（^）定义从数据流中文本行的行首开始的模式，会匹配每个由换行符决定的新数据行

# grep ^1 test6

111222333

111222333

2锁定在行尾，美元符（$）定义在行尾查找

# grep 2019$ test6

Mon Oct 14 20:18:44 CST 2019

Mon Oct 14 20:19:01 CST 2019

3匹配一个，点号（.）用来匹配除换行符之外的任意单个字符。它必须匹配一个字符，如果在点号字符的位置没有字符，那么匹配不到。

注意：在正则表达式中，空格也是字符，所以点号可以匹配空格

# grep .O test6

Mon **O**ct 14 20:18:44 CST 2019

Mon **O**ct 14 20:19:01 CST 2019

4匹配字符组，[可能出现的字符]，方括号中包含所有要匹配的字符

5排除字符组