My Perspective on Linux

基本命令

日期与时间

```
$ date # 查看当前时间
Sat Apr 13 14:46:18 DST 2019
$ cal # 查看当月日历
     April 2019
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

$ cal 2019 # 查看2019年日历
                               2019
      January               February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                  1  2                  1  2
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   3  4  5  6  7  8  9
13 14 15 16 17 18 19  10 11 12 13 14 15 16  10 11 12 13 14 15 16
20 21 22 23 24 25 26  17 18 19 20 21 22 23  17 18 19 20 21 22 23
27 28 29 30 31        24 25 26 27 28        24 25 26 27 28 29 30
                                            31

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6            1  2  3  4                     1
 7  8  9 10 11 12 13   5  6  7  8  9 10 11   2  3  4  5  6  7  8
14 15 16 17 18 19 20  12 13 14 15 16 17 18   9 10 11 12 13 14 15
21 22 23 24 25 26 27  19 20 21 22 23 24 25  16 17 18 19 20 21 22
28 29 30              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30

        July                August               September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6               1  2  3   1  2  3  4  5  6  7
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20  11 12 13 14 15 16 17  15 16 17 18 19 20 21
21 22 23 24 25 26 27  18 19 20 21 22 23 24  22 23 24 25 26 27 28
28 29 30 31           25 26 27 28 29 30 31  29 30


      October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                  1  2   1  2  3  4  5  6  7
```

```
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
27 28 29 30 31        24 25 26 27 28 29 30  29 30 31

$ cal 1 2019 # 查看2019年1月日历
    January 2019
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

文件操作

```
$ cd ~ # 切换到家目录
$ mkdir tutorial # 创建目录
$ ls # 查看当前目录信息
tutorial
$ cd tutorial/ # 切换到tutorial目录
$ ls # 没有输出，则目录为空
$ touch file1.txt # 创建文件
$ touch file2.txt # 创建文件
$ ls
file1.txt  file2.txt # 创建的文件
$ vim file1.txt # 修改文件内容
$ vim file2.txt # 修改文件内容
$ cat file1.txt # 输出文件内容
hello1
$ cat file2.txt # 输出文件内容
hello2
$ cp file1.txt file3.txt # 复制文件
$ ls
file1.txt  file2.txt  file3.txt
$ cat file1.txt
hello1
$ cat file3.txt
hello1
$ diff file1.txt file3.txt # 列出连个文件的不同之处，没有输出说明完全相同
$ diff file1.txt file2.txt
1c1
< hello1
---
> hello2
$ mv file1.txt helloworld.txt # 对文件重命名
$ ls
file2.txt file3.txt helloworld.txt
$ mv helloworld.txt ../ # 移动文件到指定目录(这里指上一级目录)
$ rm file2.txt # 删除文件
```

```
$ rm file3.txt
$ rmdir tutorial/ # 删除目录，不能删除非空目录
```

文件查看

```
$ cat invictus # 输出文件内容
Out of the night that covers me,

Black as the pit from pole to pole,
I thank whatever gods may be,
For my unconquerable soul,
In the fell clutch of circumstance,
I have not winced nor cried aloud,
Under the bludgeoning of chance,
My head if bloody, but unbowed,

Beyond this place of wrath and tear,

Looms but the Horror of the shade,
And yet the menace of the years,
Finds, and shall find, me unafraid,
It matters not how strait the gate,
How charged with punishments the scroll,
I am the master of my fate,
I am the captain of my soul.
$ head invictus -n 5 # 只显示文件前5行
Out of the night that covers me,

Black as the pit from pole to pole,
I thank whatever gods may be,
For my unconquerable soul,
$ tail invictus -n 5 #只显示文件后5行
Finds, and shall find, me unafraid,
It matters not how strait the gate,
How charged with punishments the scroll,
I am the master of my fate,
I am the captain of my soul.
$ wc invictus # 查看文件行数与单词数以及大小
 19 103 559 invictus # 19行 103个单词 559个字节 文件名称
```

命令路径查看

```
$ which git # 查看某个命令的路径
/usr/bin/git
```

文件结构与权限

文件结构

Linux文件属性

1. 用户身份
    1. 拥有者(user)
    2. 用户组(group)
    3. 其他人(others)
2. 权限
    1. 可读(r)
        1. 对于文件：可以使用类似cat命令查看
        2. 对于目录：可以使用类似ls命令查看
    2. 可写(w)
        1. 对于文件：可以使用类似vim命令进行更改
        2. 对于目录：可以使用类似touch命令在目录中创建新文件
    3. 可执行(x)
        1. 对于文件：可以运行
        2. 对于目录：可以使用类似cd命令进入
3. 类型
    1. 目录(d)
    2. 文件(-)
    3. 链接(l)

```
$ pwd # 查看当前路径
/home/wangyuxiang0829
$ cd / # 切换到根目录
$ ls # 查看当前目录信息
bin   dev  home  lib    lib64   media  opt   root  sbin  sys  usr
boot  etc  init  lib32  libx32  mnt    proc  run   srv   tmp  var
$ ls -l # 查看当前目录详细信息
total 88
lrwxrwxrwx  1 root root     7 Feb 20 00:30 bin -> usr/bin
drwxr-xr-x  1 root root   512 Nov 29 21:49 boot
drwxr-xr-x  1 root root   512 Apr  8 11:21 dev
drwxr-xr-x  1 root root   512 Apr  6 22:18 etc
drwxr-xr-x  1 root root   512 Apr  6 22:18 home
-rwxr-xr-x  1 root root 87944 Jan  1  1970 init
lrwxrwxrwx  1 root root     7 Feb 20 00:30 lib -> usr/lib
lrwxrwxrwx  1 root root     9 Feb 20 00:30 lib32 -> usr/lib32
lrwxrwxrwx  1 root root     9 Feb 20 00:30 lib64 -> usr/lib64
lrwxrwxrwx  1 root root    10 Feb 20 00:30 libx32 -> usr/libx32
drwxr-xr-x  1 root root   512 Feb 20 00:30 media
drwxr-xr-x  1 root root   512 Apr  6 22:17 mnt
drwxr-xr-x  1 root root   512 Feb 20 00:30 opt
dr-xr-xr-x  9 root root     0 Apr  8 11:21 proc
drwx------  1 root root   512 Feb 20 00:30 root
```

```
drwxr-xr-x  1 root root   512 Apr  8 11:21 run
lrwxrwxrwx  1 root root     8 Feb 20 00:30 sbin -> usr/sbin
drwxr-xr-x  1 root root   512 Feb 20 00:30 srv
dr-xr-xr-x 12 root root     0 Apr  8 11:21 sys
drwxrwxrwt  1 root root   512 Feb 20 00:34 tmp
drwxr-xr-x  1 root root   512 Feb 20 00:30 usr
drwxr-xr-x  1 root root   512 Feb 20 00:30 var
```

权限更改

```
$ cd ~
$ ls
$ touch file1.txt # 创建文件
$ ls
file1.txt # 刚创建的文件
$ ls -l file1.txt # 列出文件属性
-rw-rw-rw- 1 wangyuxiang0829 wangyuxiang0829   0 Apr 10 22:49 file1.txt
$ chmod 777 file1.txt # 更改文件权限
$ ls -l file1.txt # 列出文件属性
-rwxrwxrwx 1 wangyuxiang0829 wangyuxiang0829   0 Apr 10 22:49 file1.txt
$ chmod 000 file1.txt # 更改文件权限
$ ls -l file1.txt # 列出文件属性
---------- 1 wangyuxiang0829 wangyuxiang0829   0 Apr 10 22:49 file1.txt
```

---

grep命令与正则表达式

```
$ ls
invictus
$ cat invictus
Out of the night that covers me,

Black as the pit from pole to pole,
I thank whatever gods may be,
For my unconquerable soul,
In the fell clutch of circumstance,
I have not winced nor cried aloud,
Under the bludgeoning of chance,
My head if bloody, but unbowed,

Beyond this place of wrath and tear,

Looms but the Horror of the shade,
And yet the menace of the years,
Finds, and shall find, me unafraid,
```

```
It matters not how strait the gate,
How charged with punishments the scroll,
I am the master of my fate,
I am the captain of my soul.

$ grep Out invictus # 在指定文件中查找所有与字符串"Out"匹配的行

`Out` of the night that covers me,

$ grep of invictus # 在指定文件中查找所有与字符串"of"匹配的行

Out `of` the night that covers me,
In the fell clutch `of` circumstance,
Under the bludgeoning `of` chance,
Beyond this place `of` wrath and tear,
Looms but the Horror `of` the shade,
And yet the menace `of` the years,
I am the master `of` my fate,
I am the captain `of` my soul.

$ grep c.*n invictus # 正则表达式，表示字符'c'与'n'之间含有零个或多个任意字符

For my un`con`querable soul,
In the fell `clutch of circumstan`ce,
I have not win`ced n`or cried aloud,
Under the bludgeoning of `chan`ce,
Beyond this pla`ce of wrath an`d tear,
How `charged with punishmen`ts the scroll,
I am the `captain` of my soul.

$ grep [Aa]nd invictus # 正则表达式，表示字符'A'或者'a'后跟字符串"nd"

Beyond this place of wrath `and` tear,
`And` yet the menace of the years,
Finds, `and` shall find, me unafraid,

$ grep [A-Za-z]ou invictus # 正则表达式，表示一个任意的大写字母或者一个任意的小写字母后跟字符串"ou"

For my unconquerable `sou`l,
I have not winced nor cried a`lou`d,
I am the captain of my `sou`l.

$ grep [A-Za-z]ou invictus | wc # 将正则表达式的输出结果使用wc命令进行统计
3       18       94 # 3行，18个单词，94个字节
```

变量

普通变量

```
$ a=10
$ echo $a
10
```

全局变量

```
$ echo $USER ##当前登陆系统的用户的用户名
wangyuxiang0829
$ echo $HOME ##当前用户的主目录
/home/wangyuxiang0829
$ cd $HOME ##切换到用户主目录
$ cd ~ ##另一种方式切换到用户主目录
$ echo ~ ##等价于$HOME
/home/wangyuxiang0829
```

```
$ echo $PATH ##等价于windows中环境变量->系统变量->Path
...
$ PATH=$PATH:/home/wangyuxiang0829/workspace ##添加某个路径进环境变量
$ echo $SHELL ##当前用户的默认shell
/bin/bash
```

打包与解包

zip与unzip

```
$ ls # 空目录
$ touch file1.txt # 创建文件
$ touch file2.txt # 创建文件
$ ls
file1.txt  file2.txt
$ zip file.zip file1.txt file2.txt # 将file1.txt文件和file2.txt文件打包为file.zip文件
  adding: file1.txt (stored 0%)
  adding: file2.txt (stored 0%)
$ ls
file.zip  file1.txt  file2.txt
```

- 如果需要打包的文件很多，则可以使用`-r`选项来递归的打包

```
### 该shell脚本将在该目录下创建10个文本文件
a=1
while [ $a -le 10 ]
do
        touch file$a.txt
        a=`expr $a + 1`
        echo $a
done
```

- 我们运行以上脚本来产生足够多的文件，并使用`zip -r`命令全部打包

```
$ ls
test.sh
$ sh test.sh # 执行test.sh脚本，创建多个文件
2
3
4
5
6
7
8
9
10
11
$ ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt
file8.txt  file9.txt  test.sh
$ zip -r file.zip ./ # 将当前目录下的所有文件打包为file.zip文件
  adding: file1.txt (stored 0%)
  adding: file10.txt (stored 0%)
  adding: file2.txt (stored 0%)
  adding: file3.txt (stored 0%)
  adding: file4.txt (stored 0%)
  adding: file5.txt (stored 0%)
  adding: file6.txt (stored 0%)
  adding: file7.txt (stored 0%)
  adding: file8.txt (stored 0%)
  adding: file9.txt (stored 0%)
  adding: test.sh (deflated 7%)
```

- 既然打包完成了，我们可以使用`unzip`命令对`.zip`文件解包，可以使用`-d`选项指定输出目录

```
$ mkdir unzip # 创建unzip/目录，之后会解包到该目录
$ ls
file.zip   file10.txt  file3.txt  file5.txt  file7.txt  file9.txt  unzip
file1.txt  file2.txt   file4.txt  file6.txt  file8.txt  test.sh
$ unzip file.zip -d unzip/ # 将file.zip文件解包到unzip/目录
Archive:  file.zip
 extracting: unzip/file1.txt
 extracting: unzip/file10.txt
 extracting: unzip/file2.txt
 extracting: unzip/file3.txt
```

```
 extracting: unzip/file4.txt
 extracting: unzip/file5.txt
 extracting: unzip/file6.txt
 extracting: unzip/file7.txt
 extracting: unzip/file8.txt
 extracting: unzip/file9.txt
  inflating: unzip/test.sh
$ ls
file.zip   file10.txt  file3.txt  file5.txt  file7.txt  file9.txt  unzip
file1.txt  file2.txt   file4.txt  file6.txt  file8.txt  test.sh
$ cd unzip/
$ ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt
file8.txt  file9.txt  test.sh # 解包出来的文件
```

- 有时候我们并不像解压，只是简单的查看`.zip`文件中所包含的内容，则可以使用`unzip -l`

```
$ unzip -l file.zip # 使用-l选项列出file.zip中所包含的所有文件
Archive:  file.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2019-04-12 19:37   file1.txt
        0  2019-04-12 19:37   file10.txt
        0  2019-04-12 19:37   file2.txt
        0  2019-04-12 19:37   file3.txt
        0  2019-04-12 19:37   file4.txt
        0  2019-04-12 19:37   file5.txt
        0  2019-04-12 19:37   file6.txt
        0  2019-04-12 19:37   file7.txt
        0  2019-04-12 19:37   file8.txt
        0  2019-04-12 19:37   file9.txt
       76  2019-04-12 19:37   test.sh
---------                     -------
       76                     11 files
```

tar

```
$ tar --help
Usage: tar [OPTION...] [FILE]...
GNU 'tar' saves many files together into a single tape or disk archive, and can restore
individual files from the archive.
# -c 压缩(compress)
# -x 提取(extract)
# -v 显示详细过程(verbose)
# -t 仅显示一个归档文件中的所有文件(list)
# -f 指定文件名称(需要是最后一个参数)(file)
##Examples:
  tar -cf archive.tar foo bar  # Create archive.tar from files foo and bar.
  tar -tvf archive.tar         # List all files in archive.tar verbosely.
  tar -xf archive.tar          # Extract all files from archive.tar.
```

压缩与提取

.xz

```
$ xz --help
Usage: xz [OPTION]... [FILE]...
Compress or decompress FILEs in the .xz format.
# -z 压缩(compress)
# -d 解压缩(decompress)
##Examples:
        xz -d filename.tar.xz   # 将会生成filename.tar文件
```

.bz2

```
$ bzip2 --help
bzip2, a block-sorting file compressor.  Version 1.0.6, 6-Sept-2010.
   usage: bzip2 [flags and input files in any order]
# -z 压缩(compress)
# -d 解压缩(decompress)
## Examples:
        bzip2 -d filename.tar.bz2       # 将会生成filename.tar文件
```

.gz

```
$ gzip --help
Usage: gzip [OPTION]... [FILE]...
Compress or uncompress FILEs (by default, compress FILES in-place).
# -d 解压缩(decompress)
## Examples:
        gzip -d filename.tar.gz         # 将会生成filename.tar文件
```

Shell 脚本

控制语句与变量

shell脚本的关系操作符

- `>` --> `-gt`
- `<` --> `-lt`
- `>=` --> `-ge`
- `<=` --> `-le`
- `==` --> `-eq`
- `!=` --> `-ne`

shell脚本的if语句

```
if [ ... ]
then
        ...
        ...
else
        ...
        ...
fi
```

shell脚本的for语句

```
for x in ...
do
        ...
        ...
done
```

shell脚本的while语句

```
while [ ... ]
do
        ...
        ...
done
```

重定向

```
$ ls
invictus
$ ls > ls.txt # 重定向操作符,重定向输出,`<`将重定向输入
$ ls
invictus  ls.txt
$ cat ls.txt
invictus
ls.txt
```

算术操作符

```
+  # 加法
-  # 减法
\* # 乘法
/  # 除法
```

举例

简单的shell脚本

- 脚本

```
ls
cal
date
```

- 输出(运行`sh file.sh`命令)

```
invictus
     April 2019
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

Thu Apr 11 22:52:55 DST 2019
```

变量的赋值与运行

- 脚本

```
a=10 # 对变量的赋值操作符两边不能有空格
echo $a
```

- 输出

```
10
```

字符串输出

- 脚本

```
a=10
echo $a
echo "Hello World" # 输出字符串
```

- 输出

```
10
Hello World
```

字符串的连续输出

- 脚本

```
a=10
echo $a
echo "Hello World" $a # 连续输出字符串
```

- 输出

```
10
Hello World 10
```

echo输出

- 脚本

```
a=10
echo $a
echo Hello World $a # 字符串可以不带双引号，echo会原封不动的将之后的内容输出，包括空格
```

- 输出

```
Hello World 10
```

变量求和并输出

- 脚本

```
a=10
b=3
c=`expr $a + $b` # 加法操作符
echo $c
```

- 输出

```
13
```

变量求和并输出

- 脚本

```
a=10
b=3
# c=`expr $a * $b` # 乘法操作符不能这样写,必须进行转义
c=`expr $a \* $b` # 乘法操作符
echo $c
```

- 输出

```
30
```

条件语句

- 脚本

```
a=1
b=3
# if [ $a > $b ] # if语句格式,`>`操作符不能这样写
if [ $a -gt $b ] # `-gt`表示大于
then
        echo $a
else
        echo $b
fi # 不能少,表示if语句的结束
```

- 输出

```
3
```

for循环

- 脚本

```
for x in 1 2 3 4 5 6 7 8 9 # for语句格式
do
        echo $x
done
```

- 输出

```
1
2
3
4
5
6
7
8
9
```

while循环

- 脚本

```
x=1
while [ $x -le 10 ] # while语句格式,`-le`表示小于或等于
do
        echo $x
        x=`expr $x + 1`
done
```

- 输出

```
1
2
3
4
5
6
7
8
9
10
```

字符串与数组

字符串变量

- 脚本

```
a="hello"
b="world"

echo $a
echo $b
```

- 输出

```
hello
world
```

输入与输出

- 脚本

```
echo "Please enter a = "
read a
echo "Please enter b = "
read b

c=`expr $a + $b`
echo $a + $b = $c
```

- 输出

```
Please enter a =
1
Please enter b =
1
1 + 1 = 2
```

字符串的比较

- 脚本

```
password="hello123"

echo "Please enter a password: "
read userPassword

if [ $password = $userPassword ] # 字符串的比较必须使用单个`=`符号表示相等，'!='表示不相等
then
        echo "密码正确"
else
        echo "密码错误"
fi
```

- 输出

```
Please enter a password:
hello123
密码正确
```

字符串拼接

- 脚本

```
str1="hello"
str2="world"

str3="$str1 $str2"
echo $str3
```

- 输出

```
hello world
```

判断字符串是否为空

- 脚本

```
string=""

if [ -z $string ] # `-z`表示字符串为空 `-n`表示字符串非空
then
        echo "string is zero"
else
        echo "string is not zero"
fi
```

- 输出

```
string is zero
```

数组定义(需要bash运行而不是sh)

- 脚本

```
arr=(7 8 9 10) # 数组定义
echo ${arr[1]} # 输出下标为1的元素
```

- 输出(应该使用bash filename.sh命令)

```
# test.sh: 1: test.sh: Syntax error: "(" unexpected # 使用sh命令会报错
8 # 使用bash命令运行结果
```

数组遍历

- 脚本

```
arr=(7 8 9 10)

for i in ${arr[@]} # 类似于java中的for-each语法
do
        echo $i
done
```

- 输出

```
7
8
9
10
```

下载

```
$ wget --help
GNU Wget 1.19.4, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...
```