Module01 - Linux 基础

Linux 基础 - 课程目标

通过本次课程的学习，我们将熟悉以下 4 部分内容：

- 常用 Linux 命令
- 深入了解 Linux Shell – bash
- Shell 命令相关的正则表达式
- 四个强大的 UNIX 工具： find 、 grep 、 sed 、 awk

Module01-01
Linux 基础 : Linux 常用命令

Linux 基础 - 常用 Linux 命令

- 常用 Linux 命令
- 深入了解 bash
- 正则表达式基础
- find 、 grep 、 sed 、 awk

常用 Linux 命令 - 命令的格式

- 一般 Linux 命令的格式
  - command [-options] [arguments]
    - 通常一个命令会有（也可以没有）选项和参数，如：

```
kwarph@xuanyuan-soft:~$ ls -l ./backups
```

上面 ls 是命令， l 是选项（前面加 - ），而 ./backups 是 ls 命令的参数。

- 多个选项可以连写，也可以分开，如：

```
kwarph@xuanyuan-soft:~$ ls -lha ./backups
kwarph@xuanyuan-soft:~$ ls -l -h -a ./backups
```

  - 一些通用的选项
    - -f：强制执行
    - -h： human readable（如将文件尺寸单位表示为 K 或 M，便于阅读）
    - -i：开启交互模式
    - -R（或 -r ）：递归的执行（操作目录相关的命令）

常用 Linux 命令 - 命令的格式

- 一般 Linux 命令的格式（续）
  - 选项的位置
    - 命令的选项放置的位置比较灵活，如：

```
kwarph@xuanyuan-soft:~$ ls -l ./backups
kwarph@xuanyuan-soft:~$ ls ./backups -l
```

上面 ls 命令两种执行方式没有差异。

- 注：上述方式仅适用于 bash，其他 shell 不一定支持。
  - BSD 风格的选项
    - 部分 Linux 命令也支持 BSD 风格的选项格式，选项前不需带 -，如，对于 tar 命令而言，下面 2 种方式都可以：

```
kwarph@xuanyuan-soft:~$ tar -zcf bg.png.tar.gz bg.png
kwarph@xuanyuan-soft:~$ tar zcf bg.png.tar.gz bg.png
```

常用 Linux 命令 - 命令历史

- 调用历史命令
  - 通过 history 或 fc 命令查看之前 n 个执行过的命令（ n=16 ， 或 500 ）
  - !n：
  - !cmd：调用历史命令 cmd ，且 cmd 可以是命令的前 1 个或几个字母（也即不一定是完整的命令）
  - 向上箭头键：提取上次命令（可以一直往前遍历，配合向下箭头键往后遍历）；

```
$ history
.......
  493  cat /etc/fstab      #493 是命令历史号
.......
$ !493                     # 重新执行 cat /etc/fstab
$ !ca              # 如果之前没有执行过其它以 ca 开头的命令，则同上
```

常用 Linux 命令 - 文件名补全

- 文件名补全
  - 文件名补全可以补全：命令名、文件名、目录名
  - 代码补全使用 tab 键

常用 Linux 命令 - 文件管理

- 常用 Linux 命令
  - 文件管理
  - 打包备份
  - 网络通讯
  - 系统管理
  - 其它
- 深入了解 bash
- 正则表达式基础
- find、 grep、 sed、 awk

常用 Linux 命令 - 文件管理

- 文件管理（命令列表）

命令	简介
man	我们的第一个命令 -UNIX 命令参考手册
cd	切换路径
pwd	查看当前目录路径
ls	目录和文件列表
chmod	改变文件或目录的权限
chown	改变文件或目录的属主
chgrp	改变目录或文件的所属的组
cp	复制文件或目录
mv	移动目录、文件或目录、文件改名
rm	删除目录或文件
rmdir	删除空目录
file	查看文件类型
touch	更改文件的时间戳
mkdir	创建目录

常用 Linux 命令 - 文件管理

命令	简介
ln	为文件或目录创建连接（相当于 Windows 下的快捷方式）
more、 less	分页查看文件内容
head、 tail	查看文件的头部或尾部的内容
cat	合并文件或查看它们的内容
wc	统计文件的行、单词、字符的数量
tr	字符转换
paste	以行对行的方式合并 2 个或多个文件
split	平均分割文件
cut	从文件的每一行中提取片断
colrm	删除文件中指定的列
sort	将文件按行排序
uniq	检查及删除文本文件中重复出现的行
whereis	查找文件所在的位置（通常用于查找可执行文件或配置）
which	确认某个命令当前执行的是哪个目录下的可执行文件
locate	定位文件（查找指定文件的位置）

常用 Linux 命令 - man

- man – UNIX 命令参考手册（ Refernce Manuals ）
  - 用途：查看 Linux/UNIX 命令的手册页（在 linux 系统中，我们还可以使用 info 命令查看更详细的帮助文档）
  - Linux 手册页组织成 9 个部分
    - 1，可执行程序和 shell 命令
    - 2，系统调用（内核相关的函数）
    - 3，库调用（程序库相关的函数）
    - 4，一些特别的文件（通常是位于 /dev/ ）
    - 5，文件格式和约定，如 /etc/passwd
    - 6，游戏
    - 7，杂项
    - 8，系统管理命令
    - 9，内核函数等（非标准的）

常用 Linux 命令 - man

- man 使用示例

```
kwarph@xuanyuan-soft:~$ man man
kwarph@xuanyuan-soft:~$ man ls
kwarph@xuanyuan-soft:~$ man creat
kwarph@xuanyuan-soft:~$ man 3 sprintf
```

以下是 sprintf 的手册页片段（标准 C 库函数）

```
PRINTF(3)     Linux Programmer's Manual     PRINTF(3)

NAME
       printf,  fprintf,  sprintf, snprintf, vprintf,
       vfprintf, vsprintf, vsnprintf - formatted out-
       put conversion

SYNOPSIS
       #include <stdio.h>

       int printf(const char *format, ...);
```

常用 Linux 命令 - cd

- cd 命令 预备知识 1- Linux 目录结构

/	整个文件系统的根
|--- /bin/	存放着最经常使用的命令
|--- /boot/	存放启动 Linux 时使用的一些核心文件
|--- /dev/	设备文件目录
|--- /etc/	系统管理所需要的配置文件和子目录
| |--- /etc/init.d/	系统服务启动配置脚本目录
| +--- /etc/...	
|--- /home/	用户的主目录
|--- /lib/	存放系统所需的共享库和静态库
|--- /media/	光驱、软驱、 USB 存储设备加载所用的目录
|--- /mnt/	加载的文件系统目录
|--- /opt/	某些可选软件安装后放入此目录
|--- /proc/	不是真正的文件系统，操作系统运行时，进程信息及内核信息（比如 cpu 、硬盘分区、内存信息等）存放在这里
|--- /root/	根用户（ root 用户）的主目录
|--- /sbin/	存放系统管理命令，一般只供 root 用户使用
|--- /tmp/	存放系统运行过程中的临时文件，一般在系统重启后将被清空

常用 Linux 命令 - cd

- cd 命令 预备知识 1- Linux 目录结构（续）

```
|--- /usr/                          存放用户级的命令、应用程序、库以及它们的
|     |                             配置、帮助文档
|     |--- /usr/bin/                存放用户级的命令、应用程序
|     |--- /usr/include/            存放开发所需的 C/C++ 头文件
|     |--- /usr/lib/                存放开发所需的 C/C++ 共享库和静态库
|     |--- /usr/sbin/               类似于 /sbin
|     |--- /usr/share/              存放应用程序、命令的 manpage 等文档
|     |--- /usr/local/              一般用户应用程序、库以及它们的配置、文档等
|     |     |--- /usr/local/bin/          同 /usr/bin/
|     |     |--- /usr/local/include/      同 /usr/include/
|     |     |--- /usr/local/lib/          同 /usr/lib/
|     |     |--- /usr/local/sbin/         同 /usr/sbin/
|     |     +--- /usr/local/...
|     +--- /usr/...
|--- /var/                          存放不断变化的文件如日志、安装包缓存目录、
|                                    web 目录、 ftp 、 mailserver 相关目录等
+--- /...
```

常用 Linux 命令 - cd

- cd 命令 预备知识 2- 绝对路径与相对路径
  - UNIX 路径分隔符为： /
  - 绝对路径
    - 简单而言：在 linux 系统中，凡是以 / 打头的路径表示，均被 shell 视为绝对路径。
    - 示例： /home/kwarph， /usr/include， ...
  - 相对路径
    - 与绝对路径对应，凡是没有以 / 打头的路径表示，均被 shell 视为相对路径。
    - 示例：如当前的目录是 /usr，则 include 代表一个相对路径，表示为 /usr/ 目录下的 include（include 是相对 usr 目录而言，用绝对路径表示是： /usr/include ）。
    - 两个特殊的相对路径： .（当前目录）和 ..（当前目录的父目录）
    - Home 目录 ~： ~ 表示当前用户的主目录， ~kwarph 表示 kwarph 用户的主目录

常用 Linux 命令 - cd 、 pwd

- cd - 切换路径
- pwd – 查看当前目录路径
- 示例

```
$ cd /home/kwarph/linux_cmd
$ cd files      # 切换到 /home/kwarph/linux_cmd/files
$ ls
f1   f2   f3
$ cd f1         # 切换到 /home/kwarph/linux_cmd/files/f1
$ cd ..         # 切换到 /home/kwarph/linux_cmd/files
$ cd ./f2       # 切换到 /home/kwarph/linux_cmd/files/f2
$ cd ../f3      # 切换到 /home/kwarph/linux_cmd/files/f3
$ cd            # 切换到当前用户的主目录，等同执行： cd ~
$ pwd           # 查看当前目录
/home/kwarph
$ cd ~/linux_cmd    # 切换到当前用户的主目录下的 linux_cmd 目录
$ pwd           # 查看当前目录
/home/kwarph/linux_cmd
```

常用 Linux 命令 - ls

- ls – 目录和文件列表
  - 执行方式： ls [options] [files]
  - 常用选项：
    - `-l`：详细格式列表，每项内容占一行
    - `-a`：显示指定目录下所有文件和目录
    - `-A`：显示指定目录下所有文件和目录，但不列出 . 和 ..
    - `-i`：显示文件的 inode
    - `-R`：递归式的列举指定目录下的文件或子目录
    - `-h`： human readable（须与 `-l` 选项一同使用）
    - `-d`：只列举目录名称，而不列举其内的文件或目录
    - `-F`：标识文件类型，如目录后面加 /，符号连接后加 @，可执行文件加 *， socket 文件加 =，管道加 |， ...

常用 Linux 命令 - ls

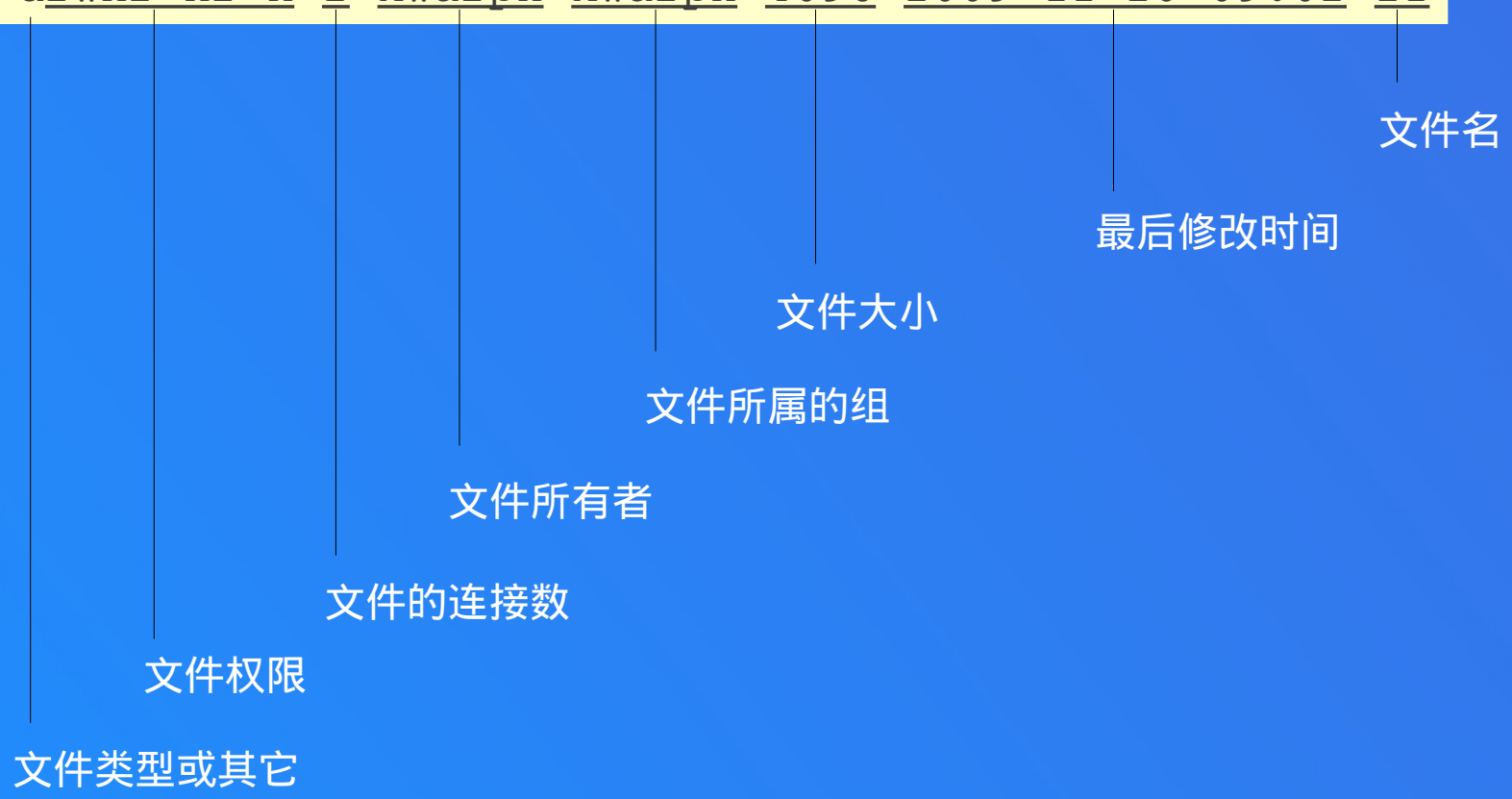
- ls – 目录和文件列表（续）
  - 示例：

```
$ pwd
/home/kwarph/linux_cmd/files
$ ls
alink f1  f2  f3  some_txt
$ ls -a    # 列出所有内容
.  ..  .ahidden  alink  f1  f2  f3  some_txt
$ ls -F    # 标识文件类型（注意：alink 是符号连接）
alink@  f1/  f2/  f3/  some_txt
$ ls -la   # 详细方式列出当前目录下所有内容（注意 alink 是目录 f1 的符号连接）
总用量 28
drwxr-xr-x 5 kwarph kwarph 4096 2009-11-20 09:27 .
drwxr-xr-x 3 kwarph kwarph 4096 2009-11-20 09:02 ..
-rw-r--r-- 1 kwarph kwarph   21 2009-11-20 09:23 .ahidden
lrwxrwxrwx 1 kwarph kwarph    2 2009-11-20 09:27 alink -> f1
drwxr-xr-x 2 kwarph kwarph 4096 2009-11-20 09:02 f1
drwxr-xr-x 2 kwarph kwarph 4096 2009-11-20 09:02 f2
drwxr-xr-x 2 kwarph kwarph 4096 2009-11-20 09:02 f3
-rw-r--r-- 1 kwarph kwarph   21 2009-11-20 09:22 some_txt
```

常用 Linux 命令 - ls

- ls – 目录和文件列表（续）
  - 解读 ls -l 输出内容：

```
drwxr-xr-x 2 kwarph kwarph 4096 2009-11-20 09:02 f1
```



常用 Linux 命令 - ls

- ls 命令相关知识
  - 隐藏文件：在 UNIX 系统中凡是文件名是以 . 开头的文件，都是隐藏文件，如 .bash_profile ， ls -a 可以列出隐藏文件
  - 文件类型：在 UNIX 系统中一切都是文件，包括设备、目录、可执行文件等等， ls -l 输出行中，文件类型通常有：
    - -：普通文件
    - d：目录
    - l：符号连接
    - c：字符设备（什么是字符设备？）
    - b：块设备（什么是块设备？）
    - s： Socket（套接字）
    - p：具名管道（ FIFO ）

常用 Linux 命令 - ls

- ls 命令相关知识（续）
  - 文件权限： UNIX 系统中用 3 个八进制数来表示文件的操作权限，如

	所有者	组	其它	权限描述
文字表示	rwx	rwx	rwx	文件所有者可读、可写、可执行，其同组成员可读、可写、可执行，其它用户可读、可写、可执行
二进制表示	111	111	111	
八进制表示	7	7	7	

    - r： 可读
    - w： 可写（即可修改、重命名、移动、删除文件等）
    - x： 可执行
    - -： 无权限（权限位设 0，如下所示）

	所有者	组	其它	权限描述
文字表示	rwx	r-x	r-x	文件所有者可读、可写、可执行，其同组成员可读、不可写、可执行，其它用户可读、不可写、可执行
二进制表示	111	101	101	
八进制表示	7	5	5	

常用 Linux 命令 - ls

- ls 命令相关知识（续）
  - Setuid： 有效用户 ID
  - Setgid： 有效组 ID
  - Sticky： 限制其它用户的修改权限

```
$ ls -l a.out
-rwxr-xr-x 1 kwarph kwarph 115473 2009-11-22 11:52 a.out
$ chmod u+s a.out    # 此命令执行后文件权限为: 4755
$ ls -l a.out
-rwsr-xr-x 1 kwarph kwarph 115473 2009-11-22 11:52 a.out
$ chmod g+s a.out    # 此命令执行后文件权限为: 6755
$ ls -l a.out
-rwsr-sr-x 1 kwarph kwarph 115473 2009-11-22 11:52 a.out
$ chmod +t a.out    # 此命令执行后文件权限为: 7755
$ ls -l a.out
-rwsr-sr-t 1 kwarph kwarph 115473 2009-11-22 11:52 a.out
```

常用 Linux 命令 - chmod, chown, chgrp

- chmod： 更改文件的权限
  - 执行方式： chmod permissions files
- chown： 更改文件的所有者
  - 执行方式： chown user[:group] files
- chgrp： 更改文件的所属组
  - 执行方式： chgrp group files

```
$ ls -l samp.sh
-rw-r--r-- 1 kwarph kwarph 0 2009-11-20 12:00 samp.sh
$ chmod +x samp.sh   # 为所有用户增加可执行权限
$ ls -l
-rwxr-xr-x 1 kwarph kwarph     0 2009-11-20 12:00 samp.sh
$ sudo chown kwarph:dba samp.sh # 注意: 需要 root 权限来执行命令
$ ls -l
-rwxr-xr-x 1 kwarph dba        0 2009-11-20 12:00 samp.sh
$ chmod -R 644 f1    # 递归式的更改 f1 目录及其内所有文件、子、孙目录等权限
                       ( -R 选项，可以用在所有操作目录的命令中 )
```

常用 Linux 命令 - cp

- cp： 复制文件或目录（ copy ）
  - 执行方式： cp [options] src-files target
  - 常用选项：
    - -R 或 -r：递归式的将源（目录）复制到指定的位置
  - 示例：

```
$ cp samp.sh some_txt f1   # 将 samp.sh 和 some_txt 复制到 f1 目录
$ cd f1
$ ls
gk2   samp.sh   some_txt
$ cp some_txt some_txt.new # 复制 some_txt，新文件名 some_txt.new
$ ls
gk2   samp.sh   some_txt   some_txt.new
```

常用 Linux 命令 - mv

- mv：移动文件、目录或更改文件、目录名称 (move)
  - 执行方式： mv src-files targets
  - 示例：

```
$ pwd;ls
/home/kwarph/linux_cmd/files/f1
gk2   samp.sh   some_txt   some_txt.new
$ mv some_txt.new ../f2   # 将 some_txt.new 移到 ../f2 目录
$ ls ../f2
some_txt.new
$ ls
gk2   samp.sh   some_txt
$ mv some_txt some_txt.old # 将 some_txt 改名为 some_txt.old
$ ls
gk2   samp.sh   some_txt.old
```

常用 Linux 命令 - rm

- rm： 删除文件或目录（ remove ）
  - 执行方式： rm [options] files
  - 常用选项：
    - -R 或 -r： 递归式的删除目录及其子、孙等目录、文件
    - -f： 强制性删除
  - 示例：

```
$ pwd;ls
/home/kwarph/linux_cmd/files
alink   f1   f2   f3   samp.sh   some_txt
$ cp -R f1 f1-bak
$ ls
alink   f1   f1-bak   f2   f3   samp.sh   some_txt
$ rm -rf f1   # 强制性、递归的删除目录 f1
$ ls
alink   f1-bak   f2   f3   samp.sh   some_txt
```

常用 Linux 命令 - rmdir

- rmdir： 删除空目录
  - 执行方式： rmdir dir
    - 提示：使用 rm -r 方式来删除目录更方便
  - 常用选项
    - -p： 删除目录及其父、祖等目录（所有目录必须为空），如：rmdir -p a/b/c 等同于 rmdir a/b/c a/b a
  - 示例：

```
$ pwd;ls
/home/kwarph/linux_cmd/files
a  alink  f1  f2  f3  samp.sh  some_txt
$ rmdir f3
$ ls
a  alink  f1  f2  samp.sh  some_txt
$ rmdir -p a/b/c     # 也可以用: rm -r a
$ ls
alink  f1  f2  samp.sh  some_txt
```

常用 Linux 命令 - file

- file： 查看文件的类型
  - 说明：与 windows 不同， UNIX 不以后缀名区分文件类型
  - 执行方式： file files
  - 示例：

```
$ ls
alink   f1   f2   samp.sh   some_txt
$ file some_txt f1 alink samp.sh
some_txt: ASCII text
f1:       directory
alink:    symbolic link to `f1'
samp.sh:  empty
```

常用 Linux 命令 - touch

- touch： 更改文件的时间戳
  - 执行方式： touch files （文件不存在则创建之）
  - 示例：

```
$ ls
alink  f1  f2  samp.sh  some_txt
$ ls -l some_txt
-rw-r--r-- 1 kwarph kwarph 21 2009-11-20 09:53 some_txt
$ touch some_txt ; ls -l some_txt
-rw-r--r-- 1 kwarph kwarph 21 2009-11-20 13:49 some_txt
$ touch -d '2009-11-18 12:32:00' some_txt
$ ls -l some_txt
-rw-r--r-- 1 kwarph kwarph 21 2009-11-18 12:32 some_txt
$ touch newfile   #newfile 不存在，创建之
$ ls
alink  f1  f2  newfile  samp.sh  some_txt
$ touch -c newfile2  #newfile2 不存在，但并不创建
$ ls
alink  f1  f2  newfile  samp.sh  some_txt
```

常用 Linux 命令 - mkdir

- mkdir：创建目录（ make directory ）
  - 执行方式： mkdir [options] dirs
  - 常用选项：
    - -p：如果父目录不存在，则创建之
  - 示例

```
$ pwd;ls
/home/kwarph/linux_cmd/files
alink  f1  f2  newfile  samp.sh
$ mkdir f3
$ ls
alink  f1  f2  f3  newfile  samp.sh
$ mkdir -p f4/{t1/{v1,v2},t2/{u1,u2}}
$ ls
alink  f1  f2  f3  f4  newfile  samp.sh
```

```
$ ls -R f4
f4:
t1  t2

f4/t1:
v1  v2

f4/t1/v1:

f4/t1/v2:

f4/t2:
u1  u2

f4/t2/u1:

f4/t2/u2:
```

常用 Linux 命令 - ln

- ln：在文件间创建连接
  - 执行方式：
    - ln [-f | -n] [-s] src-file target
    - ln [-f | -n] [-s] src-files target-dir

      注： -s 选项创建的连接为符号连接、软连接 (Symbolic link)
  - 示例：

```
$ ls f3/      # 目前为空目录
$ ln newfile some_txt f3
$ ls f3/
newfile  some_txt
$ ln -s samp.sh samp2.sh
$ ls -l samp2.sh
lrwxrwxrwx 1 kwarph kwarph 7 2009-11-20 15:19 samp2.sh -> samp.sh
```

软连接与硬连接？？

常用 Linux 命令 - more, less

- more：分页查看器
- less：另一个 more（增强的 more）
  - 执行方式：
    - more files 或 less files （用来查看文件内容）
    - 用于管道，实现分屏显示：`ls -l | more` 或 `ls -l | less`
  - 常用指令：

键盘指令	效果	键盘指令	效果
空格键 或 f	下翻一屏（forward）	b	上翻一屏（backward）
Ctrl+d 或 d	下翻预设行数（如 18 行）	Ctrl+u 或 u	上翻预设行数（如 18 行）
回车 或 j	下翻一行	y 或 k	上翻一行
/str	从前往后查找字符串 str	?str	从后往前查找字符串 str
n	继续上一次查找	/ 或 ?	继续上次 /str 或 ?str 查找
ESC	一系列指令（见 man less）	q	退出 more 或 less 命令

常用 Linux 命令 - head, tail

- head： 查看文件头部内容
- tail： 查看文件尾部内容
  - 执行方式： head [-number] files （ tail 同 ）
  - 示例：

```
$ head -5 some_txt     # 查看 some_txt 最前 5 行
$ tail -6 some_txt     # 查看 some_txt 最后 6 行
$
$ tail -f /opt/tomcat5/logs/catalina.out
          # 动态查看 catalina.out 日志文件的内容，当有新的内容添加
            到文件后，会实时输出到屏幕
```

常用 Linux 命令 - cat

- cat：合并文件或查看它们的内容（ catenate ）
  - 执行方式： cat [options] [files]
  - 常用选项：
    - -n：为每一行加行号
  - 示例：

```
$ cat   # 将标准输入（键盘）的内容输出到标准输出（屏幕）， ctrl+d 结束
$ cat > out.txt  # 将标准输入的内容输出到文件， ctrl+d 结束
$ cat > out2.txt << X   # 将标准输入的内容输出到文件， X 为结束标志
$ cat newfile some_txt  # 将 newfile 和 some_txt 文件的内容输出到
屏幕
$ cat newfile some_txt > out3.txt
        # 将 newfile 和 some_txt 文件的内容输出到文件 out3.txt
```

常用 Linux 命令 - wc

- wc：统计文件的行、单词、字节数等
  - 执行方式： wc [-l][-w][-c] files
  - 常用选项：
    - -l：查看行数
    - -w：查看单词数
    - -c：查看字节数
    - -m：查看字符数（注意：字节数不一定等于字符数）
  - 示例：

```
$ wc out2.txt some_txt
   2    2   22 out2.txt   #2 行、 2 个单词、 22 字节
  20  180 1220 some_txt
  22  182 1242 总用量
```

常用 Linux 命令 - tr

- tr：转换或删除字符（只处理单个字符）
  - 执行方式： tr [options] src-str dest-str < input-file
  - 常用选项：
    - -d：删除 src-str 中所有输入字符。
    - -s：删除所有重复出现字符序列，只保留第一个
  - 示例：

```
$ cat out.txt
hello
world
$ tr -s '[a-z]' < out.txt
Helo     #2 个连续的 l，只保留第一个
world
$ tr '[hw]' '[HW]' < out.txt   # 注意源和目标字符出现的次序对应
Hello    # 凡是小写的 h 转换成大写 H，小写 w 转换成大写 W
World
```

常用 Linux 命令 - paste

- paste：以行对行的方式合并文件
  - 执行方式： paste [options] file1 file2 …
  - 常用选项：
    - -d：指定列分隔字符或字符串（默认为 \t ）
    - -s：转列为行合并
  - 示例：

```
$ cat nf1
1
2
3
$ cat nf2
x
y
z
```

```
$ paste -d ' ' nf1 nf2
1 x
2 y
3 z
$ paste -d ' ' -s nf1 nf2
1 2 3
x y z
```

常用 Linux 命令 - split

- split： 分割大文件
  - 执行方式： split [options] < 文件 > [分割小文件名称前缀]
  - 常用选项：
    - -b ： 指定分割文件的大小，可以使 k 、 m 、 g 等为单位
  - 示例：

```
$ ls -lh eclipse.tar.gz
-rw-r--r-- 1 kwarph kwarph 183M 2009-11-20 17:42 eclipse.tar.gz
$ split -b 50m eclipse.tar.gz eclipse_part_
$ ls -lh eclipse*
-rw-r--r-- 1 kwarph kwarph  50M 2009-11-20 17:45 eclipse_part_aa
-rw-r--r-- 1 kwarph kwarph  50M 2009-11-20 17:45 eclipse_part_ab
-rw-r--r-- 1 kwarph kwarph  50M 2009-11-20 17:45 eclipse_part_ac
-rw-r--r-- 1 kwarph kwarph  33M 2009-11-20 17:45 eclipse_part_ad
-rw-r--r-- 1 kwarph kwarph 183M 2009-11-20 17:42 eclipse.tar.gz
$ cat eclipse_part_* > eclipse_new.tar.gz    # 合并起来也很简单！！
```

常用 Linux 命令 - cut

- cut：在文件的每一行中提取片断
  - 执行方式（通常）： cut -c 起始列 - 结束列 < 文件 >
  - 常用选项:
    - -c：指定列（格式如 -c2-9 ）
    - -f：指定字段（ field，格式如 -f1 ）
    - -d：指定字段分隔符（格式如 -d: 或 -d':' 或 -d ':' ）
  - 示例:

```
$ cat names
46012    DULANEY      EVAN          MOBILE    AL
46013    DURHAM       JEFF          MOBILE    AL
$ cut -c1-5 names    # 提取每行中第一列到第五列的内容
46012
46013
$ cut -d: -f1 /etc/passwd # 提取文件的第一个字段: 用户名
```

常用 Linux 命令 - colrm

- colrm： 删除文件中指定的列（ column remove ）
  - 执行方式（通常）： colrm 起始列 结束列 < inputfile
  - 示例:

```
$ cat mycpp
     1  #include <iostream>
     2
     3  int main() {
     4      std::cout << "hello, world!\n";
     5      return 0;
     6  }
$ colrm 1 8 < mycpp   # 删除每行前 1 到 8 列的字符
#include <iostream>

int main() {
    std::cout << "hello, world!\n";
    return 0;
}
```

常用 Linux 命令 - sort

- sort： 对文件进行行排序
  - 执行方式： sort [options] files
  - 常用选项：
    - -b： 忽略行首的空白字符
    - -d： 按字典顺序
    - -f： 忽略大小写
    - -i： 忽略非打印字符
    - -u： 去重复
    - -n： 按数值方式排序
    - -r： 反向排序
  - 示例：

```
$ cat sort_test
hello tiger
Hello joy
HELLO Tiger
apache
apache server
$ sort sort_test
apache
apache server
Hello joy
hello tiger
HELLO Tiger
$ sort -fu sort_test
apache
apache server
Hello joy
hello tiger
```

常用 Linux 命令 - uniq

- uniq： 检查及删除文本文件中重复出现的行
  - 执行方式： uniq [options] < 文件 > （一般与 sort 配合使用）
  - 常用选项：
    - -c：统计行出现的次数
    - -d：只列出重复的行
    - -i：忽略大小写
    - -u：只列出不重复的行
  - 示例：

```
$ cat sort_test
hello tiger
Hello joy
HELLO Tiger
apache
apache server
```

```
$ sort sort_test | uniq -i
apache
apache server
Hello joy
hello tiger
$ sort sort_test | uniq -iu
apache
apache server
Hello joy
$ sort sort_test | uniq -icu
      1 apache
      1 apache server
      1 Hello joy
$ sort sort_test | uniq -icd
      2 hello tiger
```

常用 Linux 命令 - whereis, which, locate

- whereis： 查找文件所在的位置
  - 执行方式： whereis files
- which： 确认某个命令当前执行的是哪个目录下的可执行文件，假设 /usr/bin 下和 /usr/local/bin 下都有命令 cmd， which 可以确认默认调用的是哪个 cmd
  - 执行方式： which < 命令名 >
- locate： 定位文件
  - 执行方式： locate < 文件名样式 >

```
$ whereis ssh
ssh: /usr/bin/ssh /etc/ssh /usr/share/man/man1/ssh.1.gz
$ which ls
/bin/ls
$ locate -b '\apache2'  # 凡是文件名、目录名为 apache2 的条目均被列出
```

文件管理命令 - 小结

- 遍历文件系统： cd 、 pwd
- 文件权限管理： ls 、 chmod 、 chown 、 chgrp
- 创建、修改、删除、一些更改：
  cp 、 rm 、 mv 、 mkdir 、 rmdir 、 touch 、 ln
- 查看文件内容和其它：
  cat 、 more 、 less 、 head 、 tail 、 wc 、 file
- 文本处理：
  tr 、 paste 、 split 、 cut 、 colrm 、 sort 、 uniq ，注意：它们只改变输出的结果，不是对原文件进行修改！
- 文件定位： whereis 、 which 、 locate
- 在线指南： man 和 info

常用 Linux 命令 - 打包备份

- 常用 Linux 命令
  - 文件管理
  - 打包备份
  - 网络通讯
  - 系统管理
  - 其它
- 深入了解 bash
- 正则表达式基础
- find、 grep、 sed、 awk

常用 Linux 命令 - 打包备份

- 打包备份（命令列表）

命令	简介
tar	UNIX 经典打包工具
gzip	创建 gzip 格式的压缩文件
gunzip	解压 gzip 格式的压缩文件
bzip2	创建 bzip2 格式的压缩文件
bunzip2	解压 bzip2 格式的压缩文件
zip	创建 zip 格式的压缩文件
unzip	解压 zip 格式的压缩文件
zcat	查看 gzip 压缩包内文件的内容
zmore	查看 gzip 压缩包内文件的内容
bzcat	查看 bzip2 压缩包内文件的内容
bzmore	查看 bzip2 压缩包内文件的内容
ar	创建、修改、解包归档文件（类似于 tar ）

常用 Linux 命令 - tar

- tar： UNIX 打包工具
  - 执行方式： tar action [options] [files]
  - 常用操作（操作前的 - 可要可不要）：
    - [-]c：创建
    - [-]t：查看 tarfile 里面的文件
    - [-]x：解包
  - 常用选项：
    - -C：指定路径（大写 C ）
    - -f：指定文件名称，注意：此选项后最好紧跟文件名，否则可能报错
    - -j：操作 bzip2 格式的文件（ GNU tar 的扩展，其它 UNIX tar 不支持）
    - -p：操作过程中保持各文件的原有的权限
    - -v：操作过程中显示文件
    - -z：操作 gzip 格式的文件（ GNU tar 的扩展，其它 UNIX tar 不支持）

常用 Linux 命令 - tar

- 示例：

```
#1，将指定的一个或多个文件、目录内容打成 tar 包，并查看打包过程、保留文件权限
$ tar -cvpf first.tar tree *.sql linux_cmd/*
tree
um091117.sql
linux_cmd/files/
linux_cmd/files/.a_hidden
......
$ ls -l first.tar
-rw-r--r-- 1 kwarph kwarph 61440 2009-11-21 11:50 first.tar
$
#2，同 #1，但将上述文件、目录打成 tar 包，并用 gzip 格式压缩
$ tar -czpf first.tar.gz tree *.sql linux_cmd/*     # 注意 z 选项
$ ls -l first.tar.gz
-rw-r--r-- 1 kwarph kwarph 13900 2009-11-21 11:57 first.tar.gz
#3，同 #2，但将上述文件、目录打成 tar 包，并用 bzip2 格式压缩
$ tar -cjpf first.tar.bz2 tree *.sql linux_cmd/*     # 注意 j 选项
$ ls -l first.tar.bz2
-rw-r--r-- 1 kwarph kwarph 12568 2009-11-21 12:02 first.tar.bz2
```

常用 Linux 命令 - tar

- 示例 (续) :

```
#4，利用 tar 来复制，将 /etc 下所有内容复制到 ~/backups 目录下
$ cd ~/backups
$ tar -cvf - /etc | tar -xvf -
$
#5，解包 tar 文件
$ tar -xvf first.tar      # 解压到当前目录
$ tar -xvf first.tar -C ~/backups   # 解压到指定的目录，注意 -C 选项和格式
$
#6，解包压缩过 tar 文件，操作同 #5，注意压缩格式，z-gzip，j-bzip2
$ tar -xzvf first.tar.gz    # 注意 z 选项，解 gzip 压缩格式的 tar 包
$ tar -xjvf first.tar.bz2   # 注意 j 选项，解 bzip2 压缩格式的 tar 包
$
#7，选择性打包，如将 books/ 下所有修改时间晚于 2009/11/10 的文件打包
$ tar -N '2009/11/10' -cf newbooks.tar books/
$ ls -lh newbooks.tar
-rw-r--r-- 1 kwarph kwarph 151M 2009-11-21 12:22 newbooks.tar
$ du books/ -h | grep G
......
4.6G    books/   #books 下共有 4.6G 的文件，而 2009/11/10 后修改的文件只有 151M
```

常用 Linux 命令 - gzip

- gzip： 创建 gzip 格式的压缩包
  - 执行格式： gzip [options] [-S suffix] files
  - 常用选项：
    - -c： 将压缩后的内容输出到标准输出（屏幕）
    - -d： 解压缩
    - -r： 递归式的压缩目录中的文件（注意是目录内每个文件一个 .gz 文件，不是整个目录压缩成一个 .gz 文件！）
    - -S： 指定压缩文件的后缀名，默认是 .gz
    - -t： 检查 gzip 文件的完整性
    - -v： 操作过程中显示文件
    - -#： 1 ~ 9，压缩级别，默认 6，数字越大压缩比越大，所需时间更长
  - 需要注意：
    - 默认情况下， gzip 会为每个单个文件创建一个 .gz 包
    - 创建压缩包后，源文件被删除，只留压缩文件，所以不能处理连接文件

常用 Linux 命令 - gzip

- 示例：

```
$ ls *.sql
um091117.sql  um091118.sql
$ gzip *.sql
$ ls *.sql
ls: 无法访问 *.sql: 没有该文件或目录                    # 源文件被删除
$ ls *.sql.gz
um091117.sql.gz  um091118.sql.gz
$ ls mp3
out.ogv  qzgy.mp3
$ gzip -r mp3/                  # 对目录打包，但是产生的不是一个包
$ ls mp3
out.ogv.gz  qzgy.mp3.gz   # 源文件被删除，产生 2 对应的个压缩包
$
$ ls
alink  f1  f2  f3  f4  newfile  some_txt
$ cat newfile some_txt | gzip > join2.gz   # 压缩多个文件
$ ls
alink  f1  f2  f3  f4  join2.gz  newfile  some_txt
     # 注意: 解压 join2 后就不能得到 2 个文件 newfile 和 some_txt 了
```

常用 Linux 命令 - bzip2

- bzip2： 创建 bzip2 格式的压缩包
  - 执行格式： bzip2 [options] [-S suffix] files
  - 常用选项：
    - `-k`： 保留源文件（默认会删除源文件）
    - `-z`： 压缩
    - 其它选项与 gzip 类似
  - 需要注意：
    - 默认情况下， bzip2 会为每个单个文件创建一个 .bz2 包
    - 创建压缩包后，源文件被删除，只留压缩文件，所以不能处理连接文件
  - 示例： bzip2 操作与 gzip 类似

常用 Linux 命令 - zip

- zip： 创建 zip 格式的压缩包
  - 执行方式： zip [options] files
  - 常用选项：
    - -b：指定压缩过程中使用的临时目录
    - -f：用新版本的项目替代包内旧版本的项目
    - -i：只包含（处理）指定样式的文件
    - -r：递归的处理目录及其子、孙等目录、文件
  - 简单示例：

```
$ zip tiger . *.sql    # 将当前目录所有 .sql 文件压缩到 tiger.zip
  adding: um091117.sql (deflated 59%)
  adding: um091118.sql (deflated 59%)
$ zip -r umsql . -i \*.sql
            # 递归的将当前目录下所有 .sql 压缩到 umsql.zip
```

常用 Linux 命令 - gunzip, bunzip2, unzip

- gunzip： 解压 gzip 格式的压缩包（相当于 gzip -d ）
  - 执行方式： gunzip gzip-file
- bunzip2： 解压 bzip2 格式的压缩包（相当于 bzip2 -d ）
  - 执行方式： bunzip2 <bzip2 压缩包 >
- unzip： 解压 zip 格式的压缩包
  - 执行方式： unzip <zip 压缩包 >

注意：

- 默认情况下，通过 gunzip 或 bunzip2 解压后，压缩文件被删除

常用 Linux 命令 - zcat, zmore, bzcat, bzmore

- zcat, zmore, bzcat, bzmore： 在不解压缩包的情况下查看压缩包内文件的内容
  - 执行方式： zcat gzip-file
  - zcat 、 zmore 区别： cat 查看不分页，而 more 为分页查看
  - 示例：

```
$ more test_file
just one line here.
$ bzip2 -k test_file        # 创建压缩包，且保留源文件
$ ls
alink  f1  f2  f3  f4  some_txt  test_file  test_file.bz2
$ bzcat test_file.bz2       # 不解压查看 test_file.bz2 的内容
just one line here.
```

常用 Linux 命令 - ar

- ar：创建、修改、解压归档文件（ archive ）
  - 说明： ar 可用来创建 c/c++ 静态库文件
  - 执行方式： ar [-X32_64] [-][options] archive-file src-files
  - 常用选项：
    - q：将源添加到归档文件末尾，不检查重复项
    - r：类似 q 选项，但检查重复项，在创建静态库时使用
    - t：查看归档文件中的内容（项目）
    - x：解包
  - 示例：

```
$ ar -r libartest.a  fun1.o fun2.o
```

打包备份命令 - 小结

- 最常用的 Linux 打包名： tar ，结合 -z 和 -j 选项，可以处理 gzip 、 bzip2 格式的压缩包
- 可以使用 gzip 、 gunzip 命令处理 gzip 格式的压缩包
- 可以使用 bzip2 、 bunzip2 命令处理 bzip2 格式的压缩包
- 对于常见的 zip 格式的压缩包， linux 系统中则提供了 zip 和 unzip

常用 Linux 命令 - 网络通讯

- 常用 Linux 命令
  - 文件管理
  - 打包备份
  - 网络通讯
  - 系统管理
  - 其它
- 深入了解 bash
- 正则表达式基础
- find、 grep、 sed、 awk

常用 Linux 命令 - 网络通讯

- 网络通讯（命令列表）

命令	简介
telnet	基于 telnet 协议，与其它计算机通信
ssh	基于 OpenSSH 协议的远程登录客户端（ Secure Shell ）
scp	基于 OpenSSH 协议的安全复制远程文件
ftp、 sftp	文件传输工具
ifconfig	配置网络接口
netstat	查看网络状态
tcpdump	网络传输数据分析工具（抓包工具）
ping	查看主机是否可到达
whois	域名查询工具
talk	与其它在线用户交流
write	向其他在线用户发送消息
wall	向所有在线用户发消息 (write all)

常用 Linux 命令 - telnet

- telnet：基于 telnet 协议，与其它计算机通信
  - 常用执行方式： telnet hostname/ip [port]
  - 说明：
    - 如果未指定端口，采用默认端口： 23
    - 由于 telnet 是明文传输，所以对于密码等敏感数据不安全
    - 连接主机后，进入命令模式： ctrl+]
    - 作为服务器端开发人员， telnet 命令更多的是用来检测服务进程是否运行，如：我们开发的服务器端程序，在主机 xuanyuan-soft.org.cn 开启的服务侦听于 8868 端口，我们可以这样检测：

```
$ telnet xuanyuan-soft.org.cn 8868
```

常用 Linux 命令 - ssh

- ssh ： 远程登录到指定的主机（ Secure Shell ）
  - 常用执行方式：
    - ssh -l login-name hostname/ip [port]
    - ssh login-name@hostname/ip [port]
  - 说明：
    - 如果未指定端口，采用默认端口： 22
    - 相对于 telnet ， ssh 数据是加密传输，所以相对安全

```
$ ssh kwarph@xuanyuan-soft.org.cn
$ ssh -l kwarph xuanyuan-soft.org.cn
```

常用 Linux 命令 - scp

- scp：基于 OpenSSH 协议安全复制远程文件
  - 执行方式（行为类似 cp 命令）：
    - 本机 -> 远程主机：scp local-files loginname@remote-host/ip:[dir/file]
    - 远程主机 -> 本机：scp login-name@remote-host/ip:files local-dir
  - 常用选项：
    - -p：保持文件权限
    - -r：递归式复制
  - 示例：

```
# 将当前目录所有 .sql 文件上传到 xuanyuan-soft.org.cn 主机的 kwarph 用户的主目录下的 normal 目录下
$ scp *.sql kwarph@xuanyuan-soft.org.cn:normal

# 将 xuanyuan-soft.org.cn 主机上 kwarph 用户主目录下的 normal 目录所有内容复制到本机当前目录下的 backups 目录内
$ scp -r kwarph@xuanyuan-soft.org.cn:normal ./backups
```

常用 Linux 命令 - ftp

- ftp：基于 ftp 协议的文件传输客户端
  - 常用执行方式： ftp [options] hostname/ip
  - 常用选项：
    - -i：关闭多文件传输过程中的交互式确认动作
  - 常用命令：

命令	描述	命令	描述
ascii	文本传输模式	binary(bin)	二进制传输模式
bye	退出 ftp 命令	cd、 lcd	切换远程、本地目录
cdup	远程切换到上一级目录	chmod	同 linux chmod
delete	删除文件	get、 mget	下载单个、多个文件
ls、 lls	列举远程、本地文件	put、 mput	上传单个、多个文件
rename	重命名远程文件	rmdir	删除远程目录

常用 Linux 命令 - ftp

- ftp：基于 ftp 协议的文件传输客户端
  - 示例：

```
$ ftp -i 192.168.0.6
......
ftp> cd normal/
250 Directory successfully changed.
ftp> lcd backups/
Local directory now /home/kwarph/backups
ftp> mget *.sql
local: cb.sql remote: cb.sql
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for cb.sql (7225
bytes).
226 File send OK.
7225 bytes received in 0.00 secs (9271.6 kB/s)
......
ftp> bye              # 退出 ftp
221 Goodbye.
```

常用 Linux 命令 - sftp

- sftp：类似于 ftp，基于 OpenSSH 协议，数据为加密传输
  - 常用执行方式： sftp login-name@hostname/ip
  - 常用命令：类似 ftp，命令执行方式更接近系统命令

常用 Linux 命令 - ifconfig

- ifconfig： 配置网络接口
  - 通常运行方式：
    - 查看网络接口信息： ifconfig [-a|-s] [interfaces]
    - 设置网络接口（需 root 权限）： ifconfig interface ip
  - 常用选项：
    - -a： 查看所有可用网络接口的信息
    - -s： 简短模式（同 netstat -i ）
    - up 、 down： 启用、禁用网络接口
  - 示例：

```
$ ifconfig -a
$ ifconfig eth0 192.168.0.12   # 需要 root 权限
$ ifconfig eth0 down           # 需要 root 权限
$ ifconfig eth0 up             # 需要 root 权限
```

常用 Linux 命令 - netstat

- netstat： 查看网络状态
  - 说明：此命令用来显示网络连接，路由表，接口状态，伪装连接，网络链路信息和组播成员组等
  - 常用选项：
    - `-a`： 所有侦听或不在侦听的 socket 状态
    - `-c`： 每秒刷新一次统计输出结果
    - `-e`： 附加信息，使用 2 次获取更详细的信息
    - `-n`： 数字形式的主机名
    - `-p`： 显示 socket 所属的进程 ID 和进程名
    - `-t`： 只查看 tcp 协议的 socket
    - `-u`： 只查看 udp 协议的 socket
    - `delay`： 一个整数，按指定的时间间隔刷新统计输出结果

常用 Linux 命令 - netstat

- 示例：

```
$ netstat -t -ap 3
(No info could be read for "-p": geteuid()=509 but you should be root.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State       PID/Program
tcp        0      0 *:41024                     *:*                         LISTEN      -

tcp        0      0 *:41312                     *:*                         LISTEN      -

tcp        0      0 *:mysql                     *:*                         LISTEN      -

tcp        0      0 *:23759                     *:*                         LISTEN      -
......
......
tcp        0      0 localhost.localdomain:mysql localhost.localdomain:45177 ESTABLISHED -

......
......
```

常用 Linux 命令 - whosis

- whois： 域名查询工具
  - 执行方式： whois domain

常用 Linux 命令 - talk

- talk：与其他在线用户交谈
  - 执行方式： talk user [tty]
  - 说明：
    - 要使用 talk，系统中必须开启 talk 服务
    - 可以使用 write 命令替代 talk， write 不需配置系统服务

常用 Linux 命令 - write

- write： 向其他在线用户发送消息 (类似 talk)
  - 执行方式： write user [tty]

常用 Linux 命令 - wall

- wall： 向所有在线用户发送消息
  - 执行方式：
    - wall
    - wall files （将文件内容作为消息发送给所有在线用户，需 root 权限）

网络通讯命令 - 小结

- 作为天然的网络操作系统， UNIX/Linux 系统中拥有众多的高效、实用的网络工具，远远不限于我们所列举的这几个，我们所列举的仅仅是作为软件开发人员常用的命令的一小部分。

常用 Linux 命令 - 系统管理

- 常用 Linux 命令
  - 文件管理
  - 打包备份
  - 网络通讯
  - 系统管理
  - 其它
- 深入了解 bash
- 正则表达式基础
- find 、 grep 、 sed 、 awk

- 系统管理（命令列表）

命令	简介
ps	查看进程状态
top	动态监控系统进程状态
free	查看内存使用状况
kill	向指定进程发送一个信号（通常用于终止进程）
who	显示登录到本机的用户
w	显示登录到本机的用户，并且显示他们正在做什
whoami	显示用户的登录名
id	查看用户的 id 及其它信息
last	查看查看最近登录本机的用户
uptime	报告系统连续运行的时间
date	显示或设置系统时间
dmesg	查看开机信息
umask	控制新创建文件的默认权限
ulimit	查看和设置用户的资源限制

常用 Linux 命令 - 系统管理

命令	简介
su	切换到另外的用户（通常是 root 用户）身份
sudo	以另外的用户（通常是 root 用户）身份来执行命令
df	查看文件系统的磁盘使用情况
du	查看指定目录或文件占用磁盘空间
useradd	添加用户
userdel	删除用户
usermod	修改用户
passwd	修改用户的登录口令
chsh	更改用户的登录 shell
groupadd	添加用户组
groupdel	删除用户组
groupmod	修改用户组
env	查看当前用户的环境变量
shutdown、 halt、 reboot、 poweroff	重启或关闭系统

常用 Linux 命令 - ps

- ps： 查看进程状态
  - 执行方式： ps [options]
  - 常用选项：
    - UNIX 风格：
      - -e、 -A： 显示所有进程
      - -f： 显示完整的列表模式
      - -w： 宽格式显示（如显示完整的程序路径和名称，不论行长度）
    - BSD 风格（不带 - ）：
      - a： 显示所有进程
      - u： 以用户为主的格式来显示进程状况
      - x： 显示所有程序，不论是否为终端进程（如 X 进程）
      - w： 宽格式显示（如显示完整的程序路径和名称）

常用 Linux 命令 - ps

- ps： 查看进程状态（续）
  - 示例：

```
#1， UNIX 风格显示所有进程状况
$ ps -e
$ ps -ef
$ ps -eF
$ ps -ely

#2， BSD 风格显示所有进程状况
$ ps ax
$ ps axu
```

常用 Linux 命令 - top

- top： 动态监控系统进程状态
  - 通常执行方式：
    - top
    - top -d < 刷新时间间隔 >
  - 常用指令：
    - d： 设置刷新时间间隔
    - k： 终止进程
    - U、 u： 选择用户
    - q： 退出 top 命令
    - 回车或空格：立即刷新统计
  - 详细指南： man top

常用 Linux 命令 - free

- free： 查看内存使用状况
  - 执行方式： free [-b -k -m -g] [-s <delay-time>]
  - 常用选项：
    - -b、 -k、 -m、 -g： 字节单位
    - -s： 按指定的间隔不断刷新

常用 Linux 命令 - kill

- kill：向指定的进程发送一个信号（通常用于终止进程）
  - 执行方式：
    - kill [signal] process-number （默认信号为 SIGTERM ）
    - kill -l （显示系统所有可用信号）
  - 示例：

```
$ ps -ef | grep apt-get        # 先查找要终止的进程号
kwarph   27517 10039  0 15:48 pts/4    00:00:00 man apt-get
kwarph   27536 21511  0 15:48 pts/0    00:00:00 grep apt-get
$ kill -9 27517     # 9 代表信号 SIGKILL，也可以：kill -KILL 27517
$ ps -ef | grep apt-get
kwarph   27819 21511  0 15:51 pts/0    00:00:00 grep apt-get
```

常用 Linux 命令 - who, w

- who： 显示登录到本机的用户
- w： 显示登录到本机的用户，并显示它们正在做什么

常用 Linux 命令 - whoami, id

- whoami： 查看当前用户的登录名
- id： 查看指定用户的 id 、组 id 等
  - 执行方式：
    - whoami
    - id [user]　　（不指定用户名则是查看当前用户）
  - 示例：

```
$ whoami
kwarph
$ id
uid=1003(kwarph) gid=1003(kwarph) 组=0(root),1003(kwarph)
$ id student
uid=1002(student) gid=1002(student) 组
=1002(student),1001(dba)
```

常用 Linux 命令 - last

- last： 显示最近登录本机的用户
  - last 命令默认读取 /var/log/wtmp

常用 Linux 命令 - uptime

- uptime： 报告系统连续运行时间

常用 Linux 命令 - date

- date：显示或设置系统时间
  - 通常运行方式： date [options] [+fmt-str]
  - 常用选项：
    - -d：按给定字符串的格式显示时间
    - -s：设置时间
  - 示例：

```
$ date -d yesterday
2009 年 11 月 22 日 星期日 16:25:11 CST
$ date -d tomorrow
2009 年 11 月 24 日 星期二 16:25:30 CST
$ date -d last-month
2009 年 10 月 23 日 星期五 16:25:41 CST
$ date "+%Y-%m-%d %H:%M:%S"        # 更多日期格式串见 man date
2009-11-23 16:28:29
$ date -s "2009-11-23 16:26:48"    # 设置时间，需 root 权限
```

常用 Linux 命令 - dmesg

- dmesg： 查看开机信息
  - 示例：

```
$ dmesg | grep USB     # 查看有关 USB 设备的信息
[    2.406108] ehci_hcd: USB 2.0 'Enhanced' Host
Controller (EHCI) Driver
[    2.406222] ehci_hcd 0000:00:1d.7: new USB bus
registered, assigned bus number 1
[    2.424028] ehci_hcd 0000:00:1d.7: USB 2.0 started,
EHCI 1.00
[    2.424167] hub 1-0:1.0: USB hub found
[    2.424298] ohci_hcd: USB 1.1 'Open' Host Controller
(OHCI) Driver
[    2.424314] uhci_hcd: USB Universal Host Controller
Interface driver
[    2.424406] uhci_hcd 0000:00:1d.0: new USB bus
registered, assigned bus number 2
[    2.424534] hub 2-0:1.0: USB hub found
.....
```

常用 Linux 命令 - umask

- umask： 控制新创建文件的默认权限
  - Mask 与文件权限的关系：
    - 新建目录的权限 = 777 - mask （如 mask 为 022， 则目录权限为 755 ）
    - 新建文件（非目录）的权限 =666-mask （如 mask 为 022， 则文件权限为 644 ）
  - 示例：

```
$ umask                          # 查看当前的 umask 值
0022
$ touch tf1; ls -l tf1           # 创建新文件 tf1
-rw-r--r-- 1 kwarph kwarph 0 2009-11-23 16:42 tf1
$ mkdir tfd; ls -ld tfd          # 创建新目录 tfd
drwxr-xr-x 2 kwarph kwarph 4096 2009-11-23 16:43 tfd
$ umask 0027          # 设置新的权限掩码，将其它用户的权限清零
$ touch tf2; ls -l tf2           # 创建新文件 tf2
-rw-r----- 1 kwarph kwarph 0 2009-11-23 16:48 tf2
```

常用 Linux 命令 - ulimit

- ulimit： 查看和设置用户使用系统资源的限制
  - 系统资源举例：如最大打开文件的数量、 core 文件的大小等
  - 常用选项：
    - -a： 列出所有当前资源极限。
    - -c： 以 512 字节块为单位，指定核心转储 (core 文件) 的大小。
    - -d： 以 K 字节为单位指定数据区域的大小。
    - -f： 使用 Limit 参数时设定文件大小极限（以块计），或者在未指定参数时报告文件大小极限。缺省值为 -f 标志。
    - -H： 指定设置某个给定资源的硬极限。如果用户拥有 root 用户权限，可以增大硬极限。任何用户均可减少硬极限。
    - -m： 以 K 字节为单位指定物理存储器的大小。
    - -n： 指定一个进程可以拥有的文件描述符的数量的极限。
    - -s： 以 K 字节为单位指定单个线程的堆栈内存大小。
    - -S： 指定为给定的资源设置软极限。软极限可增大到硬极限的值。如果 -H 和 -S 标志均未指定，极限适用于以上二者。
    - -t： 指定每个进程所使用的秒数（ cpu 时间）。

常用 Linux 命令 - ulimit

- 示例：

```
$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 20
file size               (blocks, -f) unlimited
pending signals                 (-i) 16382
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) unlimited
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
$ ulimit -c 20
```

常用 Linux 命令 - su

- su：切换到其他用户身份
  - 通常的执行的方式：
    - su - [user] 或 su -l [user]
    - su [user]
  - 说明：
    - 上述两种执行方式的区别：前者（带 - 的方式）将使用切换后的用户环境，包括环境变量等，后者仍沿用原用户的的环境
    - 如果没指定用户名，默认切换到 root
  - 示例

```
$ su – student      # 切换到 student
口令:               # 输入 student 的口令
$ su -              # 切换到 root
```

常用 Linux 命令 - sudo

- sudo：以其他用户（通常为 root ）的身份执行命令
  - 执行方式： sudo [options] command
  - 示例：

```
$ sudo rm -ri /opt/tfn
```

常用 Linux 命令 - df

- df：查看文件系统的磁盘使用情况
  - 示例

```
$ df -h          #-h human readable
文件系统              容量    已用    可用 已用% 挂载点
/dev/sda1            9.4G  6.4G  2.6G  72%  /
tmpfs                502M     0  502M   0%  /lib/init/rw
varrun               502M  144K  501M   1%  /var/run
varlock              502M     0  502M   0%  /var/lock
udev                 502M  140K  502M   1%  /dev
tmpfs                502M  112K  502M   1%  /dev/shm
lrm                  502M  2.2M  499M   1%  /lib/modules/2.6.28-
16-generic/volatile
/dev/sda5             26G   16G  8.6G  65%  /home
```

常用 Linux 命令 - du

- du： 查看指定目录或文件占用的磁盘空间
  - 示例：

```
$ du -h ./backups
......
164K    /home/kwarph/backups/workfiles
4.0K    /home/kwarph/backups/Linux
7.3G    /home/kwarph/backups
$ du -h ./*.sql
8.0K    cb.sql
8.0K    cc.sql
32K     um091117.sql
32K     um091118.sql
```

常用 Linux 命令 - useradd

- useradd：添加用户（需 root 权限）
  - 执行方式： useradd [options] user
  - 常用选项：
    - -d：指定新用户的主目录（不指定则在 /home 下创建）
    - -e：失效期（指定日期）
    - -g：指定新用户初始组
    - -G：指定 1 个或多个所属组
    - -m：创建主目录
    - -s：指定登录 shell （如 /bin/bash ）
  - 示例：

```
# 创建用户 tiger，主组 student，从组 dba，创建主目录，shell 为 tcsh
$ sudo useradd -d /data/tiger -g student -G dba -m -s
/bin/tcsh tiger
```

常用 Linux 命令 - userdel

- userdel： 删除用户（需 root 权限）
  - 示例：

```
# 仅删除用户，但不删除用户相关的文件等
$ sudo userdel tiger

# 删除用户，并且删除用户相关的文件（如主目录，其它位置该用户相关的配置文件等）
$ sudo userdel -r tiger
```

常用 Linux 命令 - usermod

- usermod：修改用户信息（需 root 权限）
  - 说明：与 useradd 操作类似
  - 常用选项：
    - -d：改变新用户的主目录，如果有 -m 选项，则将旧目录内容复制到新目录（旧目录不存在则直接创建新目录）
    - -e：改变失效期（指定日期）
    - -g：改变新用户初始组
    - -G：改变从组
    - -l：改变登录名（用户名）
    - -s：改变登录 shell（如 /bin/bash）
  - 示例：

```
$ sudo usermod -s /bin/bash
```

常用 Linux 命令 - passwd

- passwd ： 修改用户登录口令
  - 执行方式： passwd [user]
  - 说明：如果不指定用户名，则更改当前用户（自己）的口令

常用 Linux 命令 - chsh

- chsh ： 更改用户登录 shell
  - 执行方式： chsh[user]
  - 说明：如果不指定用户名，则更改当前用户（自己）的登录 shell
  - 示例：

```
$ sudo chsh student
[sudo] password for kwarph:
正在更改 student  的 shell
请输入新值，或直接敲回车键以使用默认值
     登录 Shell [/bin/bash]: /bin/tcsh
# 注意：必须指定 shell 的绝对路径
```

常用 Linux 命令 -groupadd, groupdel

- groupadd： 添加组
- groupdel： 删除组
- groupmod： 修改组

常用 Linux 命令 - env

- env：（通常用于）查看当前用户的环境变量
  - 几个与环境变量相关的文件：
    - ~/.bash_profile
    - ~/.bashrc
    - /etc/profile
    - /etc/environment
  - 执行方式：
    - env （查看环境变量）
    - env [options] [var=value] command [options 、 args]
  - 示例

```
$ env
$ env PATH="$PATH:." hello
hello, world!
```

常用 Linux 命令 - halt,reboot,poweroff

- halt： 挂起系统（ shutdown -h now ）
- poweroff： 关闭系统（ shutdown -p now ）
- reboot： 重启系统（ shutdown -r now ）
- shutdown： 关闭系统
  - 执行方式： shutdown [options] [delay] [warn-msg]
  - 示例：

```
#3 分钟后重启系统，并向所有登录用户打印一句话
$ shutdown -r +3 'System will reboot after 3 minutes.'
$
# 立即关闭系统
$ shutdown -p now
```

系统管理命令 - 小结

- 一些进程监控、管理命令： ps 、 top 、 kill
- 查看登录用户的信息： w 、 who 、 id 、 whoami 、 last
- 另一些系统信息：
  uptime 、 dmesg 、 free 、 df 、 du 、 ulimit 、 umask
- 管理用户和组：
  useradd 、 userdel 、 usermod 、 chsh 、 passwd 、 groupadd 、 groupdel 、 groupmod
- 取得相应的权限： su 、 sudo
- 系统 shutdown ： halt 、 poweroff 、 reboot

常用 Linux 命令 - 其它

- 常用 Linux 命令
  - 文件管理
  - 打包备份
  - 网络通讯
  - 系统管理
  - 其它
- 深入了解 bash
- 正则表达式基础
- find 、 grep 、 sed 、 awk

常用 Linux 命令 - 其它

- 其它命令列表

命令	简介
clear	终端清屏
echo	回显一行文字（用于显示变量值、字符串等）
history	命令历史
fc	命令历史（find command 或 fix command）
uname	显示系统信息
alias、 unalias	设置、解除命令别名
at	在指定的时间执行任务
crontab	计划任务
nohup	不挂断的执行一个命令

常用 Linux 命令 - clear, echo

- clear： 终端清屏
- echo： 回显一行文字（通常用于显示变量值或字符串）
  - 执行方式:
    - echo $variable
    - echo string
  - 示例:

```
$ echo hello, world!
hello, world!
$ echo $SHELL
/bin/bash
$ v=8
$ echo $v
8
$ echo `date '+%Y-%m-%d'`
2009-11-24
```

常用 Linux 命令 - history

- history： 查看命令历史

常用 Linux 命令 - fc

- fc：查看命令历史
  - 通常执行方式：`fc -l [star-num] [end-num]`
  - 示例：

```
$ fc -l 100 105
100  umask 0024
101  touch tf3
102  ls -l tf3
103  umask 0027
104  touch tf5
105  ls -l tf5
$
$ fc -l              # 将列出最近的 16 个历史命令
$ fc -l 400          # 将列出从历史号为 400 到最近所有历史命令
$ fc                 # 将编辑上一次历史命令
```

常用 Linux 命令 - uname

- uname： 显示系统信息
  - 选项：
    - -a： 显示所有信息
    - -s： 显示内核名称（如不带选项执行 uname， 等同此选项）
    - -n： 主机名称
    - -r： 内核发行名称
    - -v： 内核版本
    - -m： 硬件体系
    - -p： 处理器类型
    - -o： 操作系统名称
  - 示例：

```
$ uname
Linux
$ uname -v
#55-Ubuntu SMP Tue Oct 20 19:48:24 UTC
2009
$ uname -r
2.6.28-16-generic
$ uname -a
Linux xuanyuan-soft 2.6.28-16-generic
#55-Ubuntu SMP Tue Oct 20 19:48:24 UTC
2009 i686 GNU/Linux
```

常用 Linux 命令 - alias, unalias

- alias： 设置命令的别名
  - 执行方式： alias < 别名 >=”command [options]”
  - 将别名加入 ~/.bash_profile ， 永久使用
- unalias： 解除命令的别名
  - 执行方式： unalias < 别名 >
- 示例:

```
$ alias
alias ls='ls --color=auto'
$ alias la="ls -a"          # 现在执行 la 等同执行 ls -a
$ alias
alias la='ls -a'
alias ls='ls --color=auto'
$ unalias la
$ alias
alias ls='ls --color=auto'
```

常用 Linux 命令 - at, atq, atrm

- at：在指定的时间执行任务
  - 常用执行方式： at [-f < 执行文件名 >] [-mldbv] < 指定时间 >
  - 常用选项：
    - -f：从文件读取任务（如无 f 选项，则从键盘输入）
    - -l：（at -l 等同于 atq 命令），查看当前用户未完成的任务
    - -d：（at -d 等同与 atrm 命令），删除指定任务
  - 时间格式： at 接受非常灵活的时间格式
    - HH:MM 格式：如， 04:00 代表 4:00AM。如果时间已过它就会在第二天的这一时间执行。
    - midnight — 代表 12:00AM； noon — 代表 12:00PM； teatime — 代表 4:00PM。
    - 英文月名 日期 年份 格式：如， January 15 2002 代表 2002 年 1 月 15 日。年份可有可无。
    - MMDDYY、 MM/DD/YY、或 MM.DD.YY 格式：如， 011502 代表 2002 年 1 月 15 日。
    - now + 时间：时间以 minutes、 hours、 days、或 weeks 为单位。如， now + 5 days 代表命令应该在 5 天之后的此时此刻执行。

常用 Linux 命令 - at

- 示例：

```
$ at now+2 min        #2 分钟后执行
warning: commands will be executed using /bin/sh
at> echo tiger >> /home/kwarph/at.log    # 输入任务的动作
at>     #ctrl+d 结束输入
job 4 at Mon Nov 23 19:30:00 2009
$ at -l    # 相当于执行 atq
4    Mon Nov 23 19:30:00 2009 a kwarph
$
$ cat monitor.sh
last | grep "^kwarph" >> /home/kwarph/logon.log
$ at -f ~/monitor.sh 18.00
warning: commands will be executed using /bin/sh
job 5 at Tue Nov 24 18:00:00 2009
```

常用 Linux 命令 - crontab

- crontab：计划任务（周期性任务）
  - 创建任务命令：
    - 从键盘输入任务： crontab [-u user] -e
    - 从文件读取任务： crontab [-u user] files
  - 选项：
    - -e：从标准输入（键盘）输入任务
    - -l：列出指定用户的计划任务
    - -r：删除指定用户的计划任务
  - Cronfile 格式
    - 每条任务有 6 个字段组成，字段间用空格隔开：

      分 时 日 月 周几 要执行的命令

常用 Linux 命令 - crontab

- Cronfile 格式（续）
  - 前 5 个字段是时间，时间格式分别如下：
    - 任何时间字段接受的字符：数字、 *（星号）、 ,（逗号）、 -（减号）、 /（正斜杠）
      - * 星号代表任意时间
      - , 逗号表示列举，以第一字段为例： 0,5,10,15 表示 0 分、 5 分、 10 分、 15 分都执行任务
      - - 减号表示范围，以第一字段为例： 5-15 表示 5 到 15 分之间每隔 1 分钟执行一次任务
      - / 正斜杠可以指定时间间隔，以第一字段为例： */5 表示每隔 5 分钟执行一次任务
    - 星期字段格式：可以是数字 0-6 ，也可以是英文缩写，其对应如下：sun(0), mon(1), tue(2), wed(3), thu(4), fri(5), sat(6)
  - 任务所执行的命令必须为绝对路径
  - 一个用户可以计划多个任务
  - # 字符开头的行为注释行

常用 Linux 命令 - crontab

- cronfile 示例

```
#  每晚的 21:30 执行一次 last。
30 21 * * * /usr/bin/last >> /home/kwarph/last.log
#  每月 1、10、22 日的 4:45 执行一次 last。
45 4 1,10,22 * * /usr/bin/last >> /home/kwarph/last.log
#  每周六、周日的 1:10 执行一次 last。
10 1 * * 6,0 /usr/bin/last >> /home/kwarph/last.log
#  在每天 18:00 至 23:00 之间每隔 30 分钟执行一次 last。
0,30 18-23 * * * /usr/bin/last >> /home/kwarph/last.log
#  每星期六的 11:00 pm 执行一次 last。
0 23 * * 6 /usr/bin/last >> /home/kwarph/last.log
#  每一小时执行一次 last
* */1 * * * /usr/bin/last >> /home/kwarph/last.log
#  晚上 11 点到早上 7 点之间，每隔一小时执行一次 last
* 23-7/1 * * * /usr/bin/last >> /home/kwarph/last.log
#  每月的 4 号与每周一到周三的 11 点执行一次 last
0 11 4 * mon-wed /usr/bin/last >> /home/kwarph/last.log
#  一月一号的 4 点执行一次 last
0 4 1 jan * /usr/bin/last >> /home/kwarph/last.log
```

常用 Linux 命令 - crontab

- 执行示例：

```
$ cat ./mytask
#  每月的 4 号与每周一到周三的 11 点执行一次 last
0 11 4 * mon-wed /usr/bin/last >> /home/kwarph/last.log
#  一月一号的 4 点执行一次 last
0 4 1 jan * /usr/bin/last >> /home/kwarph/last.log
$
$ crontab ./mytask
$ crontab -l
#  每月的 4 号与每周一到周三的 11 点执行一次 last
0 11 4 * mon-wed /usr/bin/last >> /home/kwarph/last.log
#  一月一号的 4 点执行一次 last
0 4 1 jan * /usr/bin/last >> /home/kwarph/last.log
$ crontab -r     # 删除当前用户的所有计划任务
$ crontab -l
no crontab for kwarph
```

常用 Linux 命令 - nohup

- nohup：不挂断的执行一个命令（ no hangup ）
  - 说明：我们在终端执行任何一个命令，会新开一个进程，而这个进程则是此终端的子进程，当我们关闭终端（如伪终端）或 logout 后，终端进程（即 shell 进程）结束，其所属的所有子进程随之结束，如果我们想在终端进程结束或 logout 后，仍然不中断在终端开启的进程，则可使用 nohup 。
  - 执行方式： nohup < 命令 [选项、参数]> [&]
  - 示例：

```
$ nohup EchoServer -p 8868 &
```

其它命令 - 小结

- nohup 命令是后台长期运行一个命令的理想方式
- crontab 和 at 命令能为我们的日常工作带来很大的方便，特别的这些命令是系统管理员工具箱中的不可或缺的工具

常用 Linux 命令总结

- 组合不同的命令产生强大的功能，是 UNIX 工具的使用方式。
- UNIX/Linux 系统中有着数量非常庞大的命令集，我们目前只是列出其中与软件开发相关的命令中的一小部分。
- 学习与使用 UNIX/Linux 命令，最佳实践方式是：充分利用 manpage 、 info 这 2 个在线手册，多尝试，多总结。