

# Linux基础

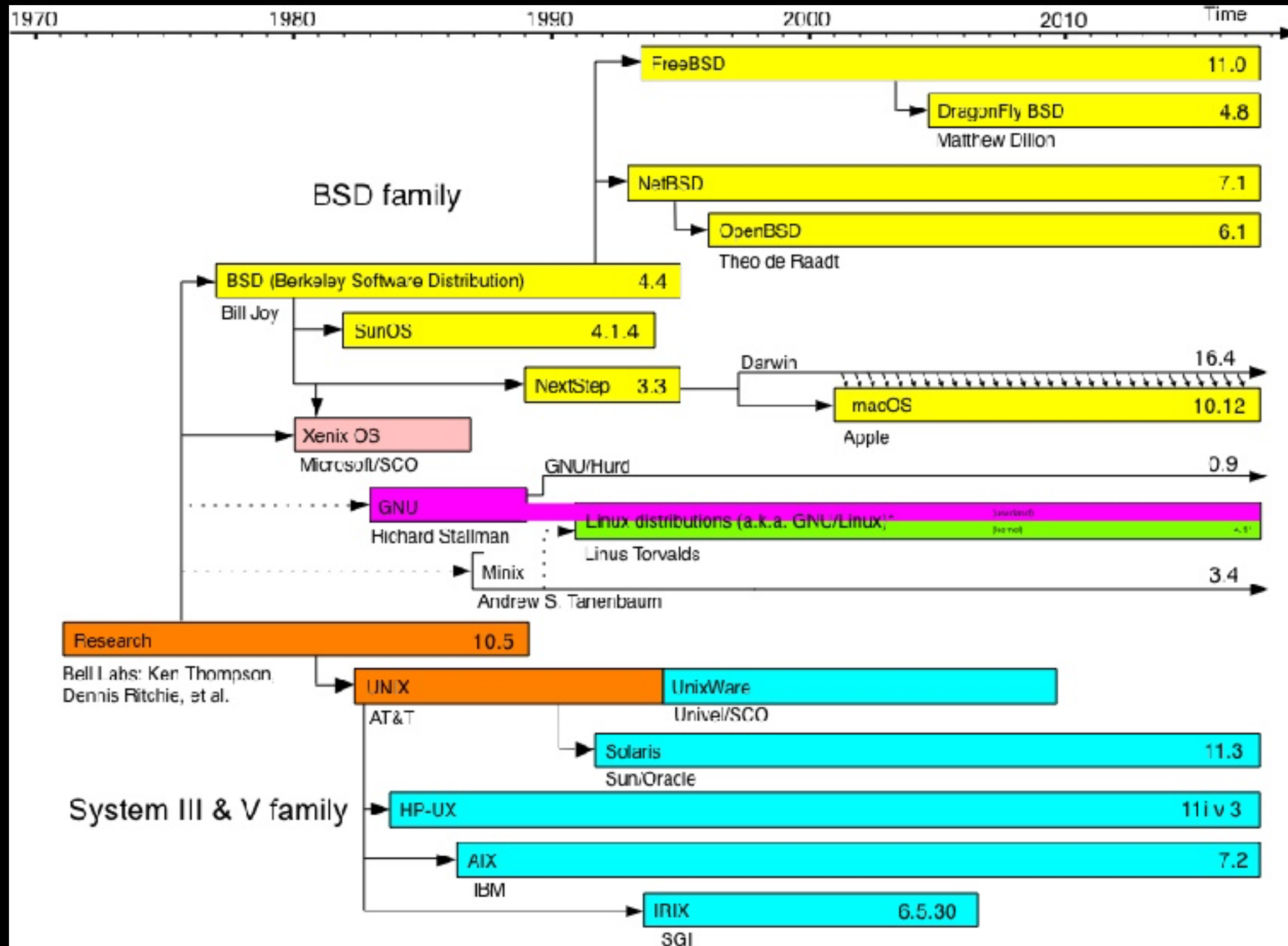
- DevOps的基础知识

2017年8月 朱青玲

# Agenda

- 分支与发行版
- 文件系统
- 权限系统
- 安装软件
- 环境变量
- 进程
- 服务
- Shell命令
- 网络
- 定时任务

# 分支



\*The penetration of GNU utilities varies between distributions, some projects use GNU's implementation of the Linux kernel (Linux-libre). Some operating systems mentioned here include GNU utilities to a lesser degree.

# 主要发布版



- Debian系, Ubuntu
- Fedora系, Red Hat Enterprise Linux, CentOS



# 查看内核发布版

```
uname -a  
cat /etc/issue  
lsb_release -a
```

# 文件系统

"On a UNIX system, everything is a file"

一切皆为文件

# 文件系统

brw-r-----	1	root	operator	1,	0	8	9	08:42	disk0
brw-r-----	1	root	operator	1,	4	8	9	08:42	disk1
lr-xr-xr-x	1	root	wheel		0	8	9	08:42	stderr ->
fd/2									
lr-xr-xr-x	1	root	wheel		0	8	9	08:42	stdin -> fd/
0									
lr-xr-xr-x	1	root	wheel		0	8	9	08:42	stdout ->
fd/1									
crw-rw-rw-	1	root	wheel	3,	2	8	9	11:25	null
crw-rw-rw-	1	root	wheel	20,	0	8	9	08:42	
tty.Bluetooth-Incoming-Port									
crw-rw-rw-	1	root	wheel	2,	0	8	9	08:42	tty
crw-rw-rw-	1	root	wheel	4,	0	8	9	08:42	ttyp0
crw-rw-rw-	1	root	wheel	4,	1	8	9	08:42	ttyp1

# 常用文件夹

/dev	设备
/lib	库和模块
/sbin	系统程序
/usr/bin	用户命令
/etc	配置等
/var/log	日志
/home	用户文件夹
/tmp	临时文件

```
tree -L 2 /
```

文件系统层次结构标准 [http://refspecs.linuxfoundation.org/FHS\\_2.3/fhs-2.3.pdf](http://refspecs.linuxfoundation.org/FHS_2.3/fhs-2.3.pdf)



# 权限系统

当前用户

w, who, who am i # 当前用户

# su和sudo

root用户， 管理员

su user1 # 切换用户

su # 切换到root

sudo adduser user1 # 创建用户， /home/user1

sudo deluser user1 # sudo用于临时切换为root权限操作， 因为命令需要root权限

# 用户组

```
group user1  
cat /etc/group | sort  
cat /etc/group | grep -E "user1"  
sudo usermod -G sudo user1  
sudo deluser user1 --remove-home
```

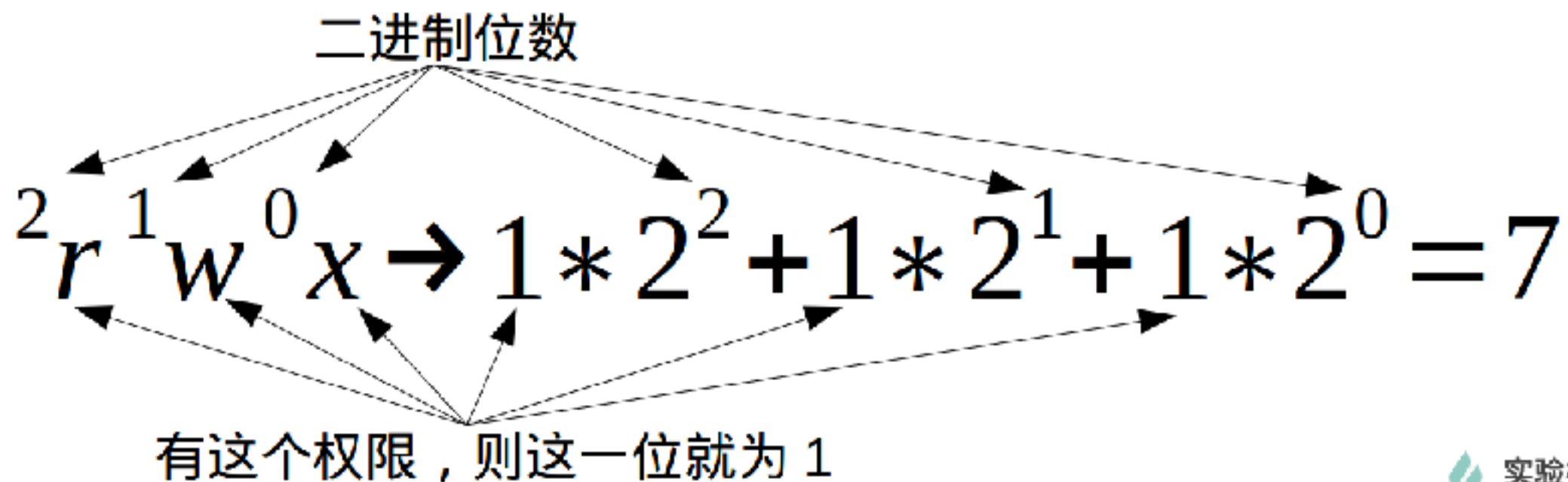
- 加入sudo用户组，才能使用sudo命令

# 文件权限

chown user1 file1 # 变更所有者

chmod 700 file1 # 变更权限

chmod ugo +rw file1 # u、g、o 分别表示 user、group、others, + 和 - 分别表示增加和去掉相应的权限



# 安装软件

- Debian

```
apt-get install nginx  
apt-get update  
apt-get upgrade  
# 升级没有依赖问题的软件包
```

- CentOS

```
yum install nginx  
yum update  
yum upgrade
```

# 手工编译

```
./configure && make && make install
```

# 环境变量

`env` # 所有环境变量

`echo $PATH` # 默认搜索路径

`export temp_env1=studio` # 设置临时变量

# 通过配置文件

```
echo "PATH=$PATH:/home/studio/bin" >> .bashrc  
source .bashrc # 重载马上生效
```

- /etc/bashrc, (有的 Linux 没有这个文件) 和 /etc/profile
- bash环境



# 进程

pid 进程ID

ps aux # 查看所有进程  
kill pid # 结束进程  
kill -signal pid  
kill -l

```
shiyantou:~/ $ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	21884	1608	?	Ss	16:41	0:00	/bin/bas
shiyantou	31	0.0	0.3	198092	28568	?	Sl	16:41	0:00	/usr/bin
shiyantou	40	0.0	0.0	30788	1864	?	S	16:41	0:00	/usr/bin
shiyantou	44	0.0	0.0	4436	648	?	S	16:41	0:00	/bin/sh
root	46	0.0	0.1	56844	14904	?	S	16:41	0:00	/usr/bin
shiyantou	57	0.0	0.0	168680	7464	?	S	16:41	0:00	xfce4-se
shiyantou	60	0.0	0.0	24428	600	?	S	16:41	0:00	/usr/bin
shiyantou	61	0.0	0.0	39208	1112	?	Ss	16:41	0:00	//bin/db
shiyantou	63	0.0	0.0	39444	2488	?	S	16:41	0:00	/usr/lib
shiyantou	67	0.0	0.0	10616	316	?	Ss	16:41	0:00	/usr/bin
shiyantou	69	0.3	0.1	177004	11484	?	S	16:41	0:00	xfwm4
shiyantou	74	0.0	0.1	269720	12336	?	Sl	16:41	0:00	xfce4-pa
shiyantou	78	0.0	0.0	168776	6596	?	S	16:41	0:00	Thunar
shiyantou	80	0.0	0.2	437304	17304	?	Sl	16:41	0:00	xfdeskto
shiyantou	85	0.2	0.1	245160	8264	?	Ssl	16:41	0:00	xfsettin
root	86	0.0	0.0	61372	3064	?	S	16:41	0:00	/usr/sbi

[16:45:43]

信号

- -9 SIGKILL 强制终止该任务
- -15 SIGTERM 正常的方式终止该任务

# 后台进程

ls & # 后面加&进入后台

jobs # 后台任务

fg, bg # 切换前后台

# top命令

top # 实时进程表

```
top - 16:44:18 up 17 days, 10:32, 1 user, load average: 1.01, 0.81, 0.
Tasks: 24 total, 1 running, 23 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.6 us, 0.4 sy, 0.0 ni, 96.5 id, 0.2 wa, 0.0 hi, 0.2 si,
KiB Mem: 8175596 total, 7864724 used, 310872 free, 460484 buffers
KiB Swap: 0 total, 0 used, 0 free. 3768240 cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
108	shiyantl+	20	0	95152	1988	1052	S	0.3	0.0	0:00.05
1	root	20	0	21884	1608	1344	S	0.0	0.0	0:00.02
31	shiyantl+	20	0	198092	28568	9512	S	0.0	0.3	0:00.20
40	shiyantl+	20	0	30788	1864	1512	S	0.0	0.0	0:00.00
44	shiyantl+	20	0	4436	648	548	S	0.0	0.0	0:00.00

切换排序

- P 根据CPU使用百分比大小进行排序
- M 根据驻留内存大小进行排序

# 服务

daemon 守护进程, 后缀通常为d, 如: httpd

- chkconfig
- systemctl

```
service --status-all  
service mysql start  
service mysql restart  
service mysql stop  
service mysql status
```

```
chkconfig --list  
chkconfig --add httpd  
chkconfig httpd on  
chkconfig httpd off
```

```
systemctl start redis.service  
systemctl restart redis.service  
systemctl stop redis.service  
systemctl status redis.service  
systemctl enable redis.service
```

```
systemctl list-units  
systemctl enable httpd.service  
systemctl disable httpd.service
```

# Shell

- sh
- csh
- bash
- zsh



# 文件和文件夹

```
cd .. , cd ../../  
cd /  
cd ~ , cd # 回到用户home
```

```
mkdir dir1  
mkdir -p dir1/dir2/dir3
```

```
cp file1 dir1/dir2/dir3  
cp -r dir1 dir2
```

```
rm file1, rm -f file1  
rm -rf dir1 # 删除文件夹
```

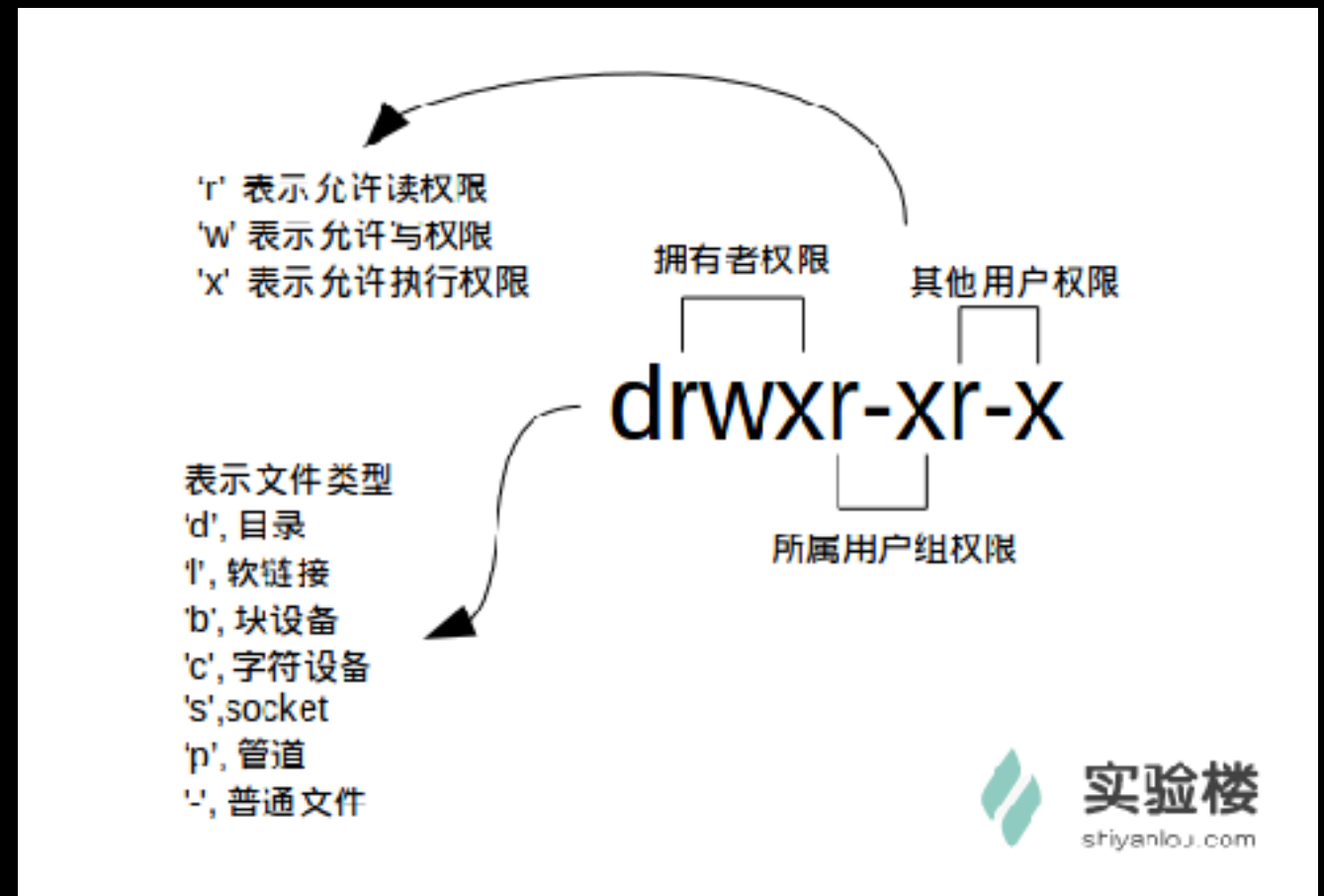
```
mv file1 dir1 # 移动  
mv file1 file2 # 重命名
```

# 列出文件

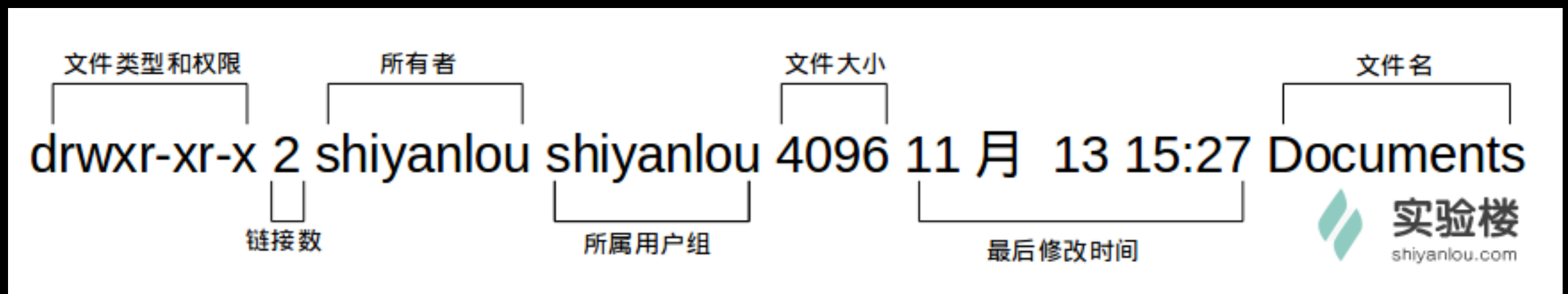
```
ls -la # 文件列表
```

```
ls -dl dir1
```

```
ls -lAsSh
```



# 显示所有文件大小，其中小s为显示文件大小，大S为按文件大小排序



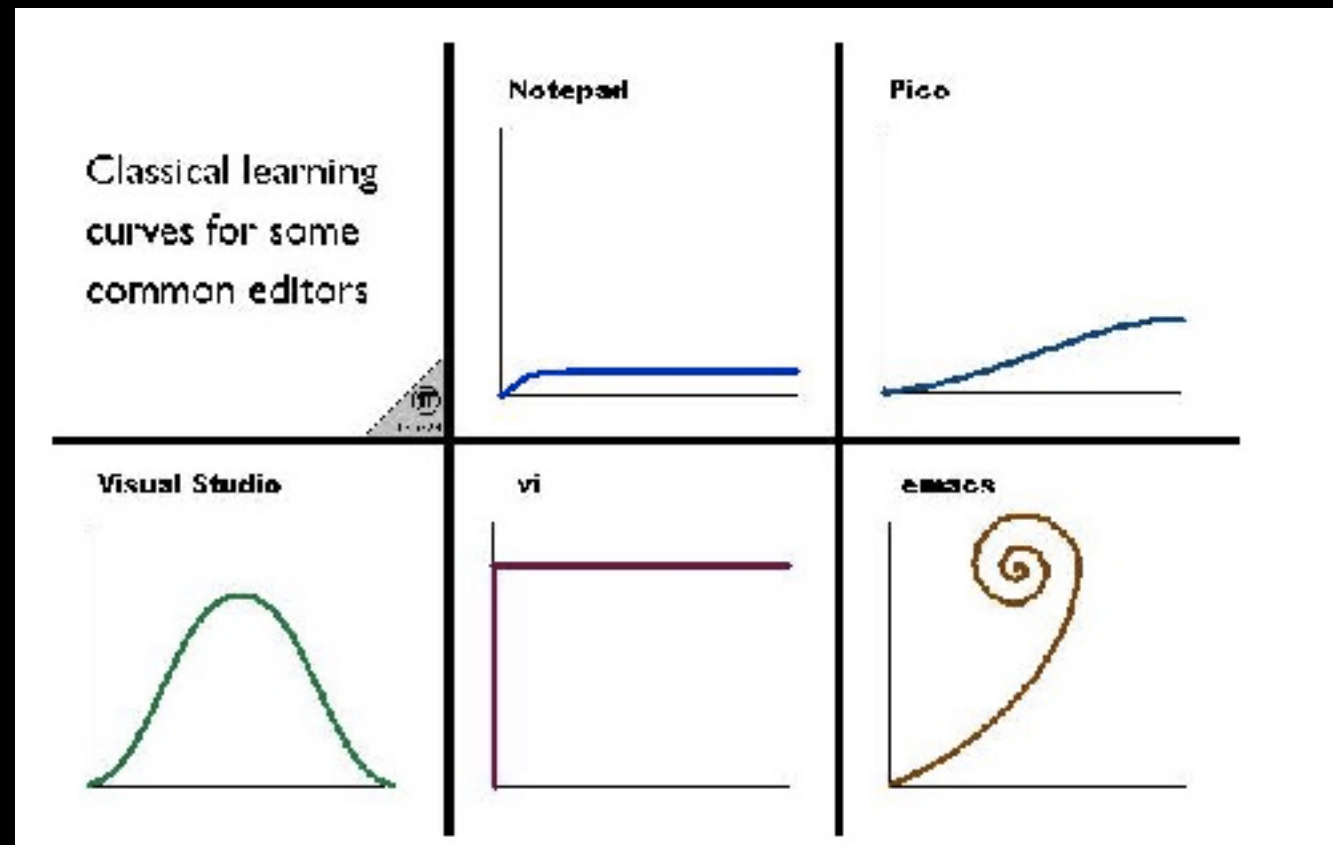
# 查看文件

```
cat -n passwd  
more, less # 支持滚动  
head, tail # 前部、后部  
tail -f file1 # 监控文件变化  
file /bin/ls # 查看文件类型
```

## 编辑器

- emacs, vi/vim, nano

<https://www.zhihu.com/question/19836903>





# 通配符

字符	含义
*	匹配 0 或多个字符
?	匹配任意一个字符
[list]	匹配 list 中的任意单一字符
[!list]	匹配 除list 中的任意单一字符以外的字符
[c1-c2]	匹配 c1-c2 中的任意单一字符 如: [0-9]
{string1,string2,...}	匹配 string1 或 string2 (或更多)其一字符
{c1..c2}	匹配 c1-c2 中全部字符 如{1..10}

```
touch love_{1..10}_linux.txt
```

# 管道

文件描述符	设备文件	说明
0	/dev/stdin	标准输入
1	/dev/stdout	标准输出
2	/dev/stderr	标准错误

```
env|sort>env.txt  
cat nothisfile &> err1.txt
```

# 压缩

```
zip -rqo zip1.zip /home/dir1 # 压缩
```

```
unzip zip1.zip # 解压
```

```
tar czvf file.tar.gz source.c source.h # 压缩 c => create, z => gzip
```

```
tar xzvf file.tar.gz # 解压 x => extract
```

- \*.tar.gz     -z
- \*.tar.bz2    -j

# 技巧

- [Tab] 补全
  - 命令
  - 文件名
- CTRL + C 中断
- 历史命令
  - 向上键
  - history
- 文件系统及程序的限制关系
  - `ulimit -a`
- 帮助
  - `man <command>`
  - `<command> -- help`

# 网络

`netstat -at` # 仅列出TCP/UDP连接

`netstat -nlpt` # 仅列出监听连接

`netstat -rn` # 路由表

- nc Netcat

- telnet

<http://www.binarytides.com/linux-netstat-command-examples/>

# 数据传输

- SSH

`ssh user@ip:port`

- scp 基于ssh协议的文件拷贝

`scp local_file remote_username@remote_ip:remote_folder`

`scp local_file remote_username@remote_ip:remote_file`

`scp -r local_folder remote_username@remote_ip:remote_folder`

`scp /home/file1.tar.gz root@192.168.1.1:/home/root`

`scp remote_username@remote_ip:remote_file local_file`

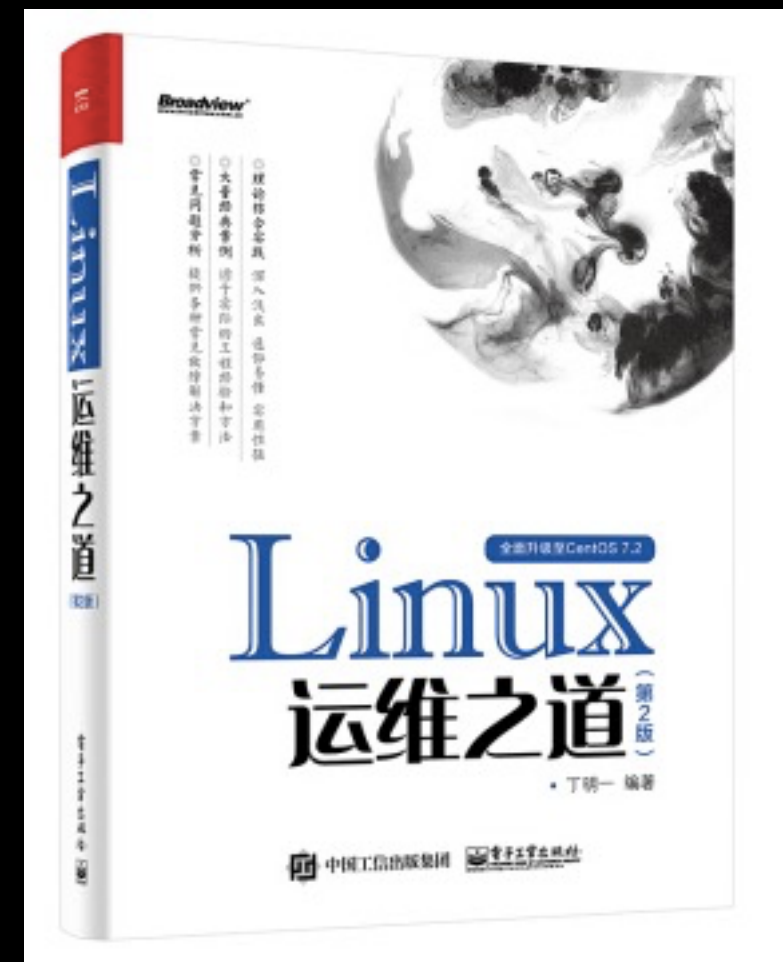
`scp -r remote_username@remote_ip:remote_folder local_folder`

`scp root@192.168.1.1:/home/root/file1.tar.gz /home/file1.tar.gz`

# rsync

rsync 是一款高效的远程数据备份和镜像工具，可快速地同步多台主机间的文件

<http://roclinux.cn/?p=2643>



# 下载和请求

- wget 支持代理服务器以及断点续传功能, wget可实现递归下载

```
wget ftp://ftp.gnu.org/pub/gnu/wget/wget-latest.tar.gz
```

- curl 支持 FTP, FTPS, HTTP, HTTPS, TFTP, SFTP, TELNET等多种协议, 功能强大

```
curl -d "name=Rafael%20Sagula&phone=3320780" http://www.where.com/guest.cgi # POST数据
```

```
curl -F "coolfiles=@fil1.gif;type=image/gif" http://www.post.com/postit.cgi # 上传文件
```

```
curl -b "name=Daniel" www.sillypage.com # 设置cookie
```

```
curl -c cookies.txt www.example.com # 完整cookie
```

```
curl -H "X-you-and-me: yes" www.love.com # 自定义请求头
```

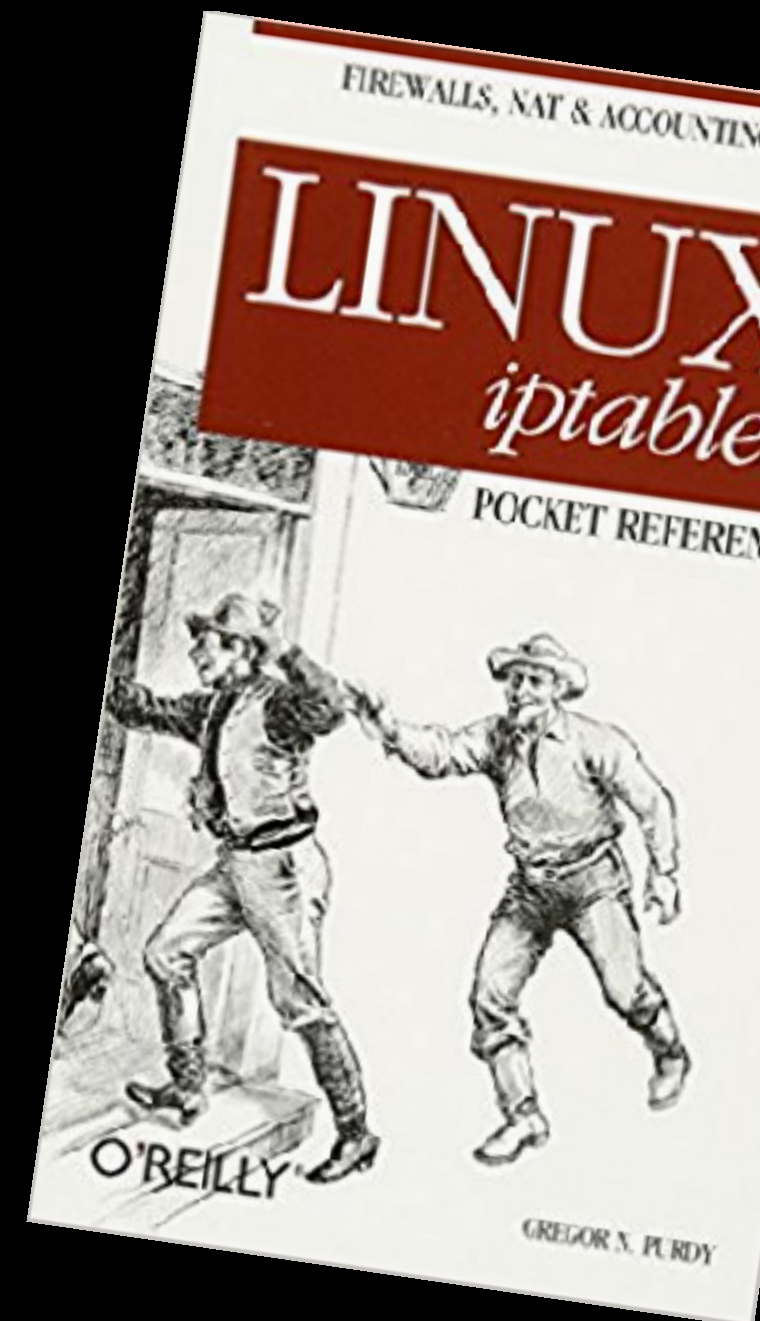
<https://curl.haxx.se/docs/manual.html>



# 防火墙

iptables防火墙

<https://wiki.centos.org/zh/HowTos/Network/IPTables>



# 定时任务

crontab

用户级

crontab -l # 列表

crontab -e # 编辑

# crontab配置

```
vi /etc/crontab
```

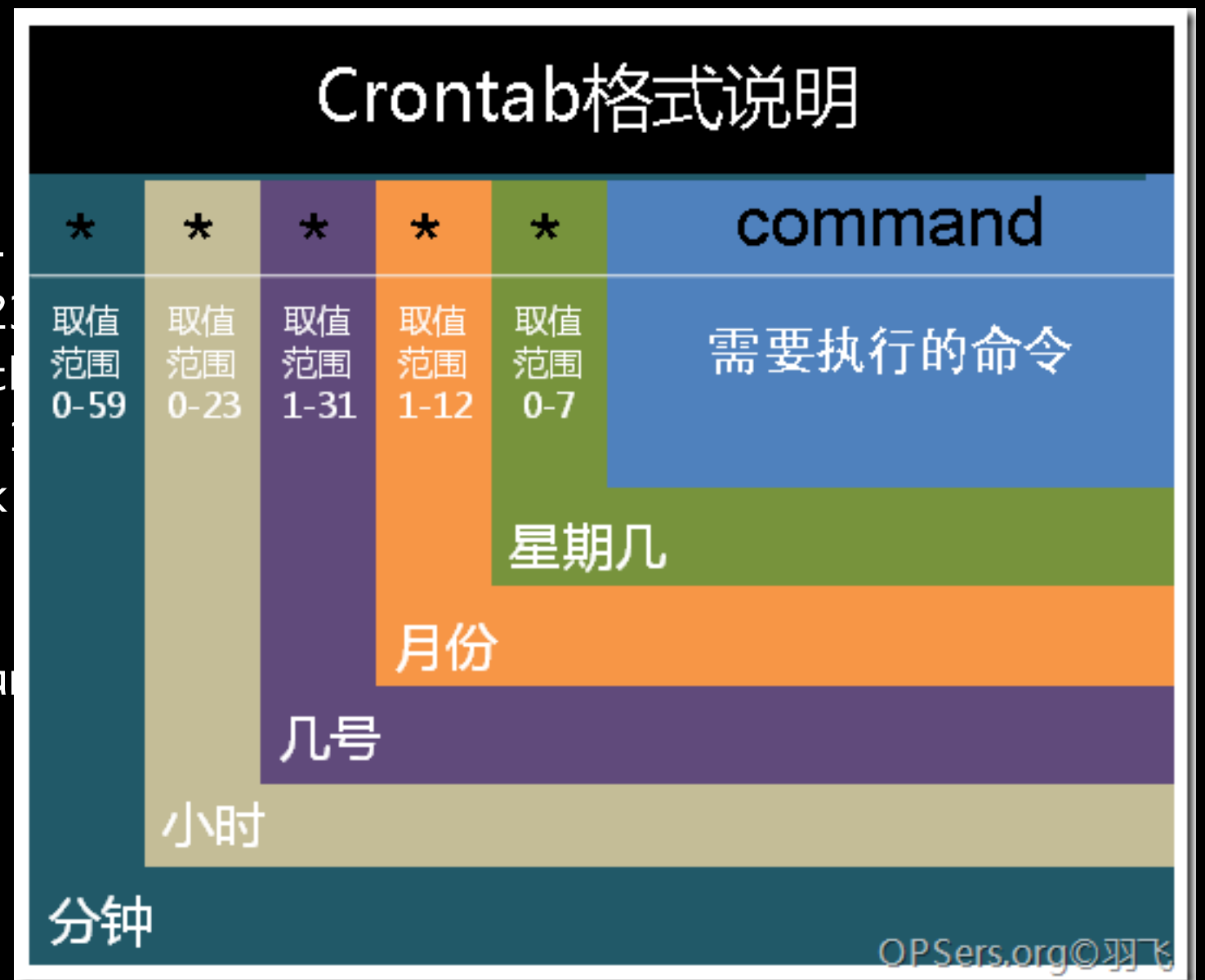
```
# Example of job definition:  
# .----- minute (0 - 59)  
# | .----- hour (0 - 23)  
# | | .----- day of month (1 - 31)  
# | | | .----- month (1 - 12)  
# | | | | .---- day of week (0 - 6)  
sun,mon,tue,wed,thu,fri,sat  
# | | | | |  
# * * * * * user-name command
```

星号 \* 任意

逗号 , 列表范围

中杠 - 连续范围

正斜线 / 间隔频率



- 需要在配置里自定义参数、路径，否则需要使用完整路径或cd path && command
- 用户级环境变量不可用

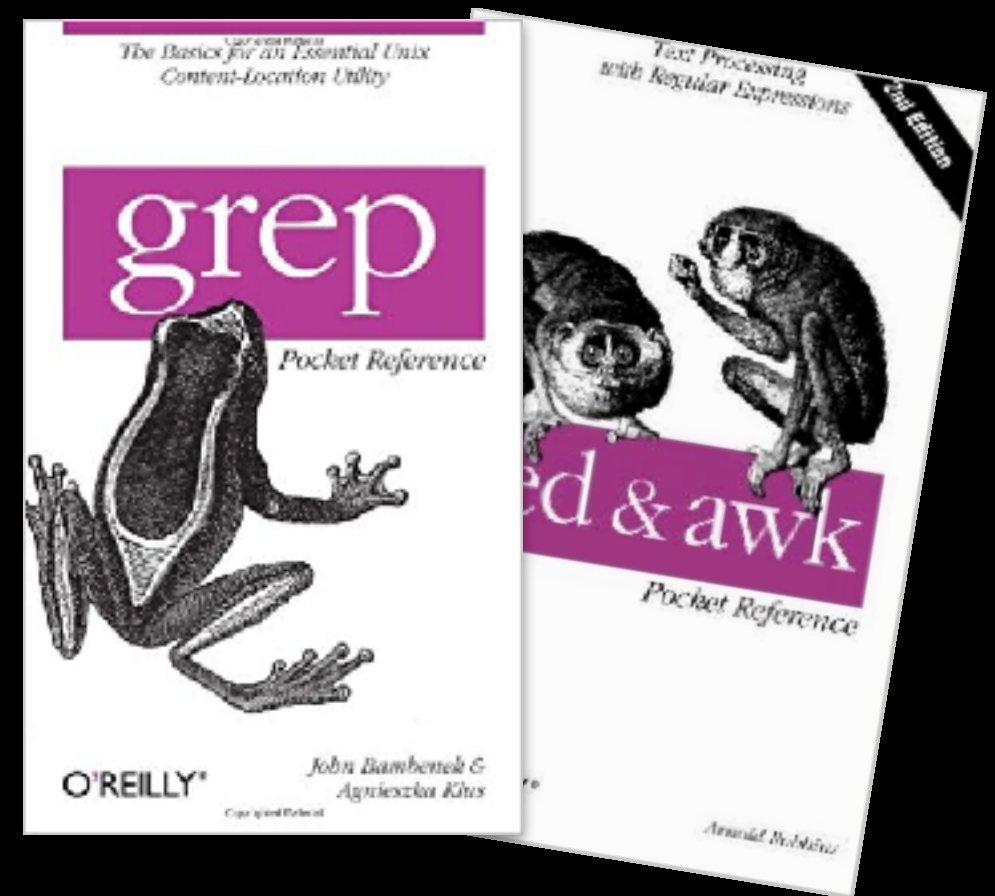
# anacron

/etc/cron.d  
/etc/cron.daily  
/etc/cron.hourly  
/etc/cron.weekly  
/etc/cron.monthly

```
# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.monthly )
```

# 扩展阅读

- grep 模式匹配
- sed 流编辑器
- awk 文本处理语言
- 实验楼 [shiyancelou.com](http://shiyancelou.com)



**Thanks**