

在linux中, &和&&, |和||介绍如下:

& 表示任务在后台执行, 如要在后台运行redis-server, 则有 `redis-server &`
&& 表示前一条命令执行成功时, 才执行后一条命令, 如 `echo '1' && echo '2'`
| 表示管道, 上一条命令的输出, 作为下一条命令参数, 如 `echo 'yes' | wc -l`
|| 表示上一条命令执行失败后, 才执行下一条命令, 如 `cat nofile || echo "fail"`

1. du命令 查看文件大小

`du -sh *` -s 仅展示一级文件夹 h 可读方式 m Mb

2. df命令 查看磁盘剩余空间

`df -h`

3.xargs 和 管道 区别

`echo '--help' | cat`

输出:

`--help`

`echo '--help' | xargs cat`

可以看到 `echo '--help' | cat` 该命令输出的是echo的内容, 也就是说将echo的内容当作cat处理的文件内容了, 实际上就是echo命令的输出通过管道定向到cat的输入了。然后cat从其标准输入中读取待处理的文本内容。这等价于在test.txt文件中有一行字符 '--help' 然后运行 `cat test.txt` 的效果。

而 `echo '--help' | xargs cat` 等价于 `cat --help` 什么意思呢, 就是xargs将其接受的字符串 --help 做成cat的一个命令参数来运行cat命令, 同样 `echo 'test.c test.cpp' | xargs cat` 等价于 `cat test.c test.cpp` 此时会将test.c和test.cpp的内容都显示出来。

xargs 能够捕获一个命令的输出, 然后传递给另外一个命令。

之所以能用到这个命令, 关键是由于很多命令不支持管道来传递参数, 而日常工作中有这个必要, 所以就有了 xargs 命令, 例如:

`find /sbin -perm +700 |ls -l` #这个命令是错误的

`find /sbin -perm +700 |xargs ls -l` #这样才是正确的

`ps aux|grep php|grep -v grep | awk '{print $2}'|xargs kill -9`