**前言**

Linux中的一些小技巧可以大大提高你的工作效率，本文就细数那些提高效率或者简单却有效的Linux技巧。

**命令编辑及光标移动**

这里有很多快捷键可以帮我们修正自己的命令。接下来使用光标二字代替光标的位置。

**删除从开头到光标处的命令文本**

ctrl + u，例如：

$ cd /proc/tty;ls -al光标如果此时使用ctrl + u快捷键，那么该条命令都会被清除，而不需要长按backspace键。

**删除从光标到结尾处的命令文本**

ctrl+k，例如：

$ cd /proc/tty光标;ls -al如果此时使用ctrl + k快捷键，那么从光标开始处到结尾的命令文本将会被删除。

还有其他的操作，不再举例，例如：

* ctrl + a:光标移动到命令开头
* ctrl + e：光标移动到命令结尾
* alt  f:光标向前移动一个单词
* alt  b：光标向前移动一个单词
* ctrl w：删除一个词（以空格隔开的字符串）

**历史命令快速执行**

我们都知道history记录了执行的历史命令，而使用!＋历史命令前的数字，可快速执行历史命令。另外，还可以使用ctrl+r搜索执行过的命令。

**部分历史命令查看**

history会显示大量的历史命令，而fs -l只会显示部分。

**实时查看日志**

$ tail -f filename.log

tail -f 加文件名，可以实时显示日志文件内容。当然，使用less命令查看文件内容，并且使用shift+f键，也可达到类似的效果。

**磁盘或内存情况查看**

**怎么知道当前磁盘是否满了呢？**

$ df -h  
/dev/sda14      4.6G   10M  4.4G   1% /tmp  
/dev/sda11      454M  366M   61M  86% /boot  
/dev/sda15       55G   18G   35G  35% /home  
/dev/sda1       256M   31M  226M  12% /boot/efi  
tmpfs           786M   64K  786M   1% /run/user/1000

使用df命令可以快速查看各挂载路径磁盘占用情况。

**当前目录各个子目录占用空间大小**

如果你已经知道home目录占用空间较大了，你想知道home目录下各个目录占用情况：

$ du -h –-max-depth=1 /home(或者-d 1)  
18G    /home/hyb  
16K    /home/lost+found  
18G    /home/

这里指定了目录深度，否则的话，它会递归统计子目录占用空间大小，可自行尝试。

**当前内存使用情况**

$ free -h  
              total        used        free      shared  buff/cache   available  
Mem:           7.7G        3.5G        452M        345M        3.7G        3.5G  
Swap:          7.6G          0B        7.6G

通过free的结果，很容易看到当前总共内存多少，剩余可用内存多少等等。

**使用-h参数**

不知道你是否注意到，我们在前面几个命令中，都使用了-h参数，它的作用是使得结果以人类可读的方式呈现，所以我们看到它呈现的单位是G,M等，如果不使用-h参数，可以自己尝试一下会是什么样的结果呈现。

**根据名称查找进程id**

想快速直接查找进程id，可以使用：

$ pgrep hello  
22692

或者：

$ pidof hello  
22692

其中，hello是进程名称

**根据名称杀死进程**

一般我们可以使用kill　-9 pid方式杀死一个进程，但是这样就需要先找到这个进程的进程id，实际上我们也可以直接根据名称杀死进程，例如：$ killall hello

或者：$ pkill hello

**查看进程运行时间**

可以使用下面的命令查看进程已运行时间：

$ ps -p 24525 -o lstart,etime   
                 STARTED     ELAPSED  
Sat Mar 23 20:52:08 2019       02:45

其中24525是你要查看进程的进程id。

**快速目录切换**

* cd -　回到上一个目录
* cd  回到用户家目录

**多条命令执行**

我们知道使用分号隔开可以执行多条命令，例如：

$ cd /temp/log/;rm -rf \*

但是如果当前目录是/目录，并且/temp/log目录不存在，那么就会发生激动人心的一幕：

bash: cd: /temp/log: No such file or directory  
（突然陷入沉默）

因为;可以执行多条命令，但是不会因为前一条命令失败，而导致后面的不会执行，因此，cd执行失败后，仍然会继续执行rm -rf \*，由于处于/目录下，结果可想而知。

所以你还以为这种事故是对rf -rf \*的力量一无所知的情况下产生的吗？

如果解决呢？很简单，使用&&，例如:

$ cd /temp/log/&&rm -rf \*

这样就会确保前一条命令执行成功，才会执行后面一条。

**查看压缩日志文件**

有时候日志文件是压缩的，那么能不能偷懒一下，不解压查看呢？当然可以啦。例如：

$ zcat test.gz  
test log

或者：

$ zless test.gz  
test log

**删除乱码文件**

无论是自己意外创建还是程序异常创建，难免会出现一些命名奇怪或者乱码文件，可以选择删除Linux中特殊名称文件的多种方式。

对于特殊字符，如<>!\*等组成的文件如何删除呢？例如：

$ rm <>\!\*  
rm: missing operand  
Try 'rm --help' for more information.

对于这种类型的文件，我们需要用引号将文件名包含起来：

$ rm "<>\!\*"  
$

**清空文件内容**

比如有一个大文件，你想快速删除，或者不想删除，但是想清空内容：

>filename

**将日志同时记录文件并打印到控制台**

在执行shell脚本，常常会将日志重定向，但是这样的话，控制台就没有打印了，如何使得既能记录日志文件，又能将日志输出到控制台呢？

$ ./test.sh |tee test.log

**终止并恢复进程执行**

我们使用ctrl+z 暂停一个进程的执行，也可以使用fg恢复执行。例如我们使用

$ cat filename

当我们发现文件内容可能很多时，使用ctrl+z暂停程序，而如果又想要从刚才的地方继续执行，则只需要使用fg命令即可恢复执行。或者使用bg使得进程继续在后台执行。

**计算程序运行时间**

我们可能会进程写一些小程序，并且想要知道它的运行时间，实际上我们可以很好的利用time命令帮我们计算，例如：

$ time ./fibo 30  
the 30 result is 832040  
  
real    0m0.088s  
user    0m0.084s  
sys    0m0.004s

它会显示系统时间，用户时间以及实际使用的总时间。

**查看内存占用前10的进程**

$ ps -aux|sort -k4nr |head -n 10

**快速查找你需要的命令**

我们都知道man可以查看命令的帮助手册，但是如果我们想要某个功能却不知道使用哪个命令呢？别着急，还是可以使用man：

$ man -k "copy files"  
cp (1)               - copy files and directories  
cpio (1)             - copy files to and from archives  
git-checkout-index (1) - Copy files from the index to the working tree  
gvfs-copy (1)        - Copy files  
gvfs-move (1)        - Copy files  
install (1)          - copy files and set attributes

使用-k参数，使得与copy files相关的帮助手册都显示出来了。

**命令行下的复制粘贴**

我们知道，在命令行下，复制不能再是ctrl + c了，因为它表示终止当前进程，而控制台下的复制粘贴需要使用下面的快捷键：

* ctrl +  insert
* shift + insert

**搜索包含某个字符串的文件**

例如，要在当前目录下查找包含test字符串的文件：

$ grep -rn "test"  
test2.txt:1:test

它便可以找到该字符串在哪个文件的第几行。

**屏幕冻结**

程序运行时，终端可能输出大量的日志，你想简单查看一下，又不想记录日志文件，此时可以使用ctrl+s键，冻结屏幕，使得日志不再继续输出，而如果想要恢复，可使用ctrl+q退出冻结。

**无编辑器情况下编辑文本文件**

如果在某些系统上连基本的vi编辑器都没有，那么可以使用下面的方式进行编辑内容：

$ cat >file.txt  
some words  
(ctrl+d)

编辑完成后，ctrl+d即可保存。

**查看elf文件**

**查看elf文件头信息**

例如：

$ readelf -h filename

我们在显示结果中，可以看到运行的平台，elf文件类型，大小端情况等。

**查看库中是否包含某个接口**

$ nm filename |grep interface

这里是从文件filename中查看是否包含interface接口，前提是该文件包含符号表。

**Linux命令全称总结**

pwd:print work directory 打印当前目录 显示出当前工作目录的绝对路径

ps: process status(进程状态，类似于windows的任务管理器)

常用参数：－auxf

ps -auxf 显示进程状态

df: disk free 其功能是显示磁盘可用空间数目信息及空间结点信息。换句话说，就是报告在任何安装的设备或目录中，还剩多少自由的空间。

du: Disk usage

rpm：即RedHat Package Management，是RedHat的发明之一

rmdir：Remove Directory（删除目录）

rm：Remove（删除目录或文件）

cat: concatenate 连锁

cat file1file2>>file3 把文件1和文件2的内容联合起来放到file3中

insmod: install module,载入模块

ln -s : link -soft 创建一个软链接，相当于创建一个快捷方式

mkdir：Make Directory(创建目录)

touch

man: Manual

su：Swith user(切换用户)

cd：Change directory

ls：List files

ps：Process Status

mkdir：Make directory

rmdir：Remove directory

mkfs: Make file system

fsck：File system check

uname: Unix name

lsmod: List modules

mv: Move file

rm: Remove file

cp: Copy file

ln: Link files

fg: Foreground

bg: Background

chown: Change owner

chgrp: Change group

chmod: Change mode

umount: Unmount

dd: 本来应根据其功能描述“Convert an copy”命名为“cc”，但“cc”已经被用以代表“CComplier”，所以命名为“dd”

tar：Tape archive （磁带档案）

ldd：List dynamic dependencies

insmod：Install module

rmmod：Remove module

lsmod：List module

文件结尾的"rc"（如.bashrc、.xinitrc等）：Resource configuration

Knnxxx /Snnxxx（位于rcx.d目录下）：K（Kill）；S(Service)；

nn（执行顺序号）；xxx（服务标识）

.a（扩展名a）：Archive，static library

.so（扩展名so）：Shared object，dynamically linked library

.o（扩展名o）：Object file，complied result of C/C++ source file

RPM：Red hat package manager

dpkg：Debian package manager

apt：Advanced package tool（Debian或基于Debian的发行版中提供部分Linux命令缩写）

bin = Binaries (二进制文件)

/dev = Devices (设备)

/etc = Etcetera (等等)

/lib = LIBrary

/proc = Processes

/sbin = Superuser Binaries (超级用户的二进制文件)

/tmp = Temporary (临时)

/usr = Unix Shared Resources

/var = Variable (变量)

FIFO = First In, First Out

GRUB = GRand Unified Bootloader

IFS= Internal Field Seperators

LILO = LInux LOader

MySQL = My 是最初作者女儿的名字，

SQL = Structured QueryLanguage

PHP = Personal Home Page Tools = PHP HypertextPreprocessor

PS = Prompt String

Perl = "Pratical Extraction and Report Language"(实际的抽取和报告语言) ="Pathologically Eclectic Rubbish Lister"

Python 得名于电视剧Monty Python's Flying Circus

Tcl = Tool Command Language

Tk = ToolKit

VT = Video Terminal

YaST = Yet Another Setup Tool

apache = "a patchy" server

apt = Advanced Packaging Tool

ar = archiver

as = assembler

awk = "Aho Weiberger and Kernighan"三个作者的姓的第一个字母

bash = Bourne Again SHell

bc = Basic (Better) Calculator

bg = BackGround

biff = 作者HeidiStettner在U.C.Berkely养的一条狗,喜欢对邮递员汪汪叫。

cal = Calendar (日历)

cat = Catenate (链接)

cd = Change Directory

chgrp = Change Group

chmod = Change Mode

chown = Change Owner

chsh = Change Shell

cmp = compare

cobra = Common Object Request BrokerArchitecture

comm = common

cp = Copy

cpio = CoPy In and Out

cpp = C Pre Processor

cron = Chronos 希腊文时间

cups = Common Unix Printing System

cvs = Current Version System

daemon = Disk And Execution MONitor

dc = Desk Calculator

dd = Disk Dump (磁盘转储)

df = Disk Free

diff = Difference

dmesg = diagnostic message

du = Disk Usage

ed = editor

egrep = Extended GREP

elf = Extensible Linking Format

elm = ELectronic Mail

emacs = Editor MACroS

eval = EVALuate

ex = EXtended

exec = EXECute (执行)

fd = file descriptors

fg = ForeGround

fgrep = Fixed GREP

fmt = format

fsck = File System ChecK

fstab = FileSystem TABle

fvwm = F\*\*\* Virtual Window Manager

gawk = GNU AWK

gpg = GNU Privacy Guard

groff = GNU troff

hal = Hardware Abstraction Layer

joe = Joe's Own Editor

ksh = Korn SHell

lame = Lame Ain't an MP3 Encoder

lex = LEXical analyser

lisp = LISt Processing = Lots of IrritatingSuperfluous Parentheses

ln = Link

lpr = Line PRint

ls = list

lsof = LiSt Open Files

m4 = Macro processor Version 4

man = MANual pages

mawk = Mike Brennan's AWK

mc = Midnight Commander

mkfs = MaKe FileSystem

mknod = Make Node

motd = Message of The Day

mozilla = MOsaic GodZILLa

mtab = Mount TABle

mv = Move

nano = Nano's ANOther editor

nawk = New AWK

nl = Number of Lines

nm = names

nohup = No HangUP

nroff = New ROFF

od = Octal Dump

passwd = Passwd

pg = pager

pico = PIne's message COmposition editor

pine = "Program for Internet News &Email" = "Pine is not Elm"

ping = 拟声 又 = Packet Internet Grouper

pirntcap = PRINTer CAPability

popd = POP Directory

pr = pre

printf = Print Formatted

ps = Processes Status

pty = pseudo tty

pushd = PUSH Directory

pwd = Print Working Directory

rc = runcom = run command, rc还是plan9的shell

rev = REVerse

rm = ReMove

rn = Read News

roff = RunOFF

rpm = RPM Package Manager = RedHat PackageManager

rsh, rlogin, rvim中的

r = Remote

rxvt = ouR XVT

seamoneky = 我

sed = Stream Editor

seq = SEQuence

shar = Shell ARchive

slrn = S-Lang rn

ssh = Secure Shell

ssl = Secure Sockets Layer

stty = Set TTY

su = Substitute User

svn = SubVersion

tar = Tape ARchive

tcsh = TENEX C shell

tee = T (T形水管接口)

telnet = TEminaL over Network

termcap = terminal capability

terminfo = terminal information

tex = τέχνη的缩写，希腊文art

tr = traslate

troff = Typesetter new ROFF

tsort = Topological SORT

tty = TeleTypewriter

twm = Tom's Window Manager

tz = TimeZone

udev = Userspace DEV

ulimit = User's LIMIT

umask = User's MASK

uniq = UNIQue

i = VIsual = Very Inconvenient

vim = Vi IMproved

wall = write all

wc = Word Count

wine = WINE Is Not an Emulator

xargs = eXtended ARGuments

xdm = X Display Manager

xlfd = X Logical Font Description

xmms = X Multimedia System

xrdb = X Resources DataBase

xwd = X Window Dump

yacc = yet another compiler compiler

Fish = the Friendly Interactive SHell

su = Switch User

MIME = Multipurpose Internet Mail Extensions

ECMA = European Computer ManufacturersAssociation