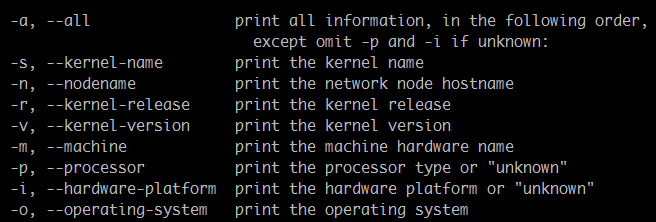
**uname -a**

列印當前電腦和作業系統的名稱、版本及其他細節



**cat /etc/issue**

想知道它是 Fedora, Debian, 還是Ubuntu 等等的 Linux distribution version 可以用此command



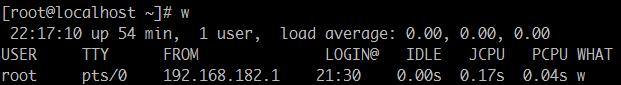
**uptime**

觀察系統啟動時間與工作負載



**w**

顯示目前登入使用者相關資訊



USER | 登入的使用者

TTY | 表示終端機

FROM | 表示來源位址

LOGIN@ | 表登入時間

IDLE | 表閒置時間

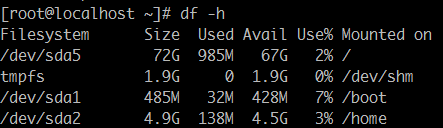
JCPU | 表示對於同一個終端機下所有程序累績的執行時間

PCPU | 表示 WHAT 欄位所顯示的程序所顯示的程序累積的執行時間

WHAT | 表示目前正執行的工作

**df -h**

文件系統 容量 已用 可用 已用% 掛載點



**cd 、mkdir、pwd**

• cd /tmp/

• mkdir -p aa/bb/cc

• cd aa

• cd bb/cc

• pwd

• cd ~

• cd - //返回進入此目錄之前所在的目錄

• pwd

**ctime, atime，mtime**

atime(Access time)：檔案上次被讀取的時間

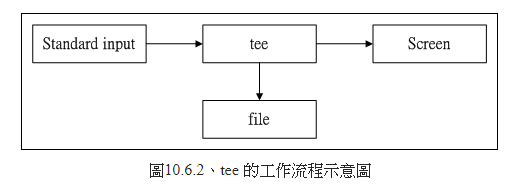
ctime(status Change time)：檔案的屬性或內容上次被修改的時間

mtime(Modified time)：檔案的內容上次被修改的時間

* ls -l所列出的長資訊，該時間是mtime
* 如果利用touch某檔案，則三種時間屬性均被更新
* 如果你修改了某一檔案內容，則ctime與mtime會更新
* 如果某檔案僅被讀取或另存新檔，則只有atime會被改變
* 如果你只改變檔案屬性(如權限、檔名、所有人或是suid、acl、chattr等)，則只有ctime被更新。
* atime相較之下是最常變動的時間標記，可以將之關閉以增進I/O，特別是多人多工，小檔案多以及laptop環境。filesystem設定裡可以增加noatime屬性，或是chattr +A固定住atime(chattr僅適用於ext2/3/4系列的filesystem)

**tee 雙向重導向**

tee 會同時將資料流分送到檔案去與螢幕 (screen)



pls use this command

• cd /etc/

• ls -A | tee /tmp/aa/bb/cc/file (要先有此file檔 資料才有辦法導入)

**tail 從指定點開始將檔寫到標準輸出**

[root@localhost ~]# ping 192.168.120.204 > test.log &

[1] 11891[root@localhost ~]# tail -f test.log  //迴圈查看檔內容，在後臺ping遠端主機。並輸出檔到test.log；這種做法也使用於一個以上的檔案監視。用Ctrl＋c來終止。

PING 192.168.120.204 (192.168.120.204) 56(84) bytes of data.

64 bytes from 192.168.120.204: icmp\_seq=1 ttl=64 time=0.038 ms

64 bytes from 192.168.120.204: icmp\_seq=2 ttl=64 time=0.036 ms

64 bytes from 192.168.120.204: icmp\_seq=3 ttl=64 time=0.033 ms

只列列出 file 的第 50 ⾏行行內容？

root@localhost:~/aa/bb/cc$ cat -n file | head -n 50 | tail -n 1

50 init

**cp 複製檔案或目錄**

-p：保留來源檔案或目錄的屬性，包括擁有者、群組、權限與時間

-R, -r：遞迴處理，針對目錄下的所有檔案與子目錄一併處理，可以利用它進行目錄的copy

-A：保留來源檔案或目錄的屬性，包括擁有者、群組、權限與時間&遞迴處理 //等於cp -Rp (cp -A)

**rm 用來刪除目錄/檔案**

-f, --force    忽略不存在的文件，從不給出提示。

-i, --interactive 進行互動式刪除 //二度確認是否要刪除

-r, -R, --recursive   指示rm將參數中列出的全部目錄和子目錄均遞迴地刪除。

ln **建立一個同步的鏈結「類似Windows捷徑」 工具**

分軟鏈結和硬鏈結兩種

軟鏈結:

[root@localhost test]# ln -s log2013.log link2013

[root@localhost test]# ll

lrwxrwxrwx 1 root root     11 12-07 16:01 link2013 -> log2013.log

-rw-r--r-- 1 root bin      61 11-13 06:03 log2013.log

為log2013.log檔案創建軟鏈結link2013，如果log2013.log丟失，link2013將失效

硬鏈結:

[root@localhost test]# ln log2013.log ln2013

[root@localhost test]# ll

lrwxrwxrwx 1 root root     11 12-07 16:01 link2013 -> log2013.log

-rw-r--r-- 2 root bin      61 11-13 06:03 ln2013

-rw-r--r-- 2 root bin      61 11-13 06:03 log2013.log

為log2013.log創建硬鏈結ln2013，log2013.log與ln2013的各項屬性相同

1.原始檔案案被刪除後，並沒有影響硬鏈結檔案；軟鏈結檔案在centos系統下不斷的閃爍，提示原始檔案案已經不存在

2.重建原始檔案案後，軟鏈結不在閃爍提示，說明已經鏈結成功，找到了鏈結檔案系統；重建後，硬鏈結檔案並沒有受到原始檔案案影響，硬鏈結檔案的內容還是保留了刪除前原始檔案案的內容，說明硬鏈結已經失效

3.目錄只能創建軟鏈結

4.目錄創建鏈結必須用絕對路徑，相對路徑創建會不成功，會提示：符號連接的層數過多這樣的錯誤

5.在鏈結目標目錄中修改檔案都會在原始檔案案目錄中同步變化

可參考 http://jashliao.pixnet.net/blog/post/164632275-%E6%AF%8F%E5%A4%A9%E4%B8%80%E5%80%8Blinux%E6%8C%87%E4%BB%A4--ln-%E6%8C%87%E4%BB%A4(%E5%BB%BA%E7%AB%8B%E4%B8%80%E5%80%8B%E5%90%8C%E6%AD%A5%E7%9A%84%E9%8F%88

**gzip 檔案解壓縮**

命令參數：

-d或--decompress或----uncompress 　解開壓縮檔。

-f或--force 　強行壓縮檔。不理會檔案名稱或硬連接是否存在以及該檔是否為符號連接。

-l或--list 　列出壓縮檔的相關資訊。

-n或--no-name 　壓縮檔時，不保存原來的檔案名稱及時間戳記記。

-N或--name 　壓縮檔時，保存原來的檔案名稱及時間戳記記。

-q或--quiet 　不顯示警告資訊。

-r或--recursive 　遞迴處理，將指定目錄下的所有檔及子目錄一併處理。

-v或--verbose 　顯示指令執行過程。

壓縮

root@localhost:~/aa/bb/cc$ gzip file

root@localhost:~/aa/bb/cc$ la

-rw-r--r--. 1 root root 2886 Aug 28 23:52 file.cp

-rw-r--r--. 1 root root 785 Aug 29 01:40 file.gz

解壓縮

gzip -d (gunzip)

root@localhost:~/aa/bb/cc$ gzip -vd file.gz //等同gunzip file.gz

file.gz: 74.1% -- replaced with file

root@localhost:~/aa/bb/cc$ la

-rw-r--r--. 1 root root 2906 Aug 29 01:40 file

**file用來探測文件的類型**

[root@localhost ~]# file install.log

install.log: UTF-8 Unicode text

[root@localhost ~]# ls -l /var/mail lrwxrwxrwx 1 root root 10 08-13 00:11 /var/mail -> spool/mail

[root@localhost ~]# file /var/mail /var/mail: symbolic link to `spool/mail'

**zcat 無法解壓縮 只能顯示內容**

root@localhost:~/aa/bb/cc$ zcat file.gz

**tar 檔案解壓縮**

命令參數：

-c 打包一個 tar 檔案

-x 解開一個 tar 檔案

-t 檢視 tar 檔案的內容

-z 使用 gzip 壓縮

-v 顯示建立 tar 檔案的過程

-f 指定 tar 檔案的檔案名稱。此參數的後面要接檔案名稱，因此要注意參數的順序 (通常是把 f 參數寫在最後一個，或者是與其它參數拆開使用)

壓縮：

[root@localhost ~]# tar zcvf FileName.tar.gz DirName

解壓縮：

[root@localhost ~]# tar zxvf FileName.tar.gz

可參考

http://note.drx.tw/2008/04/command.html

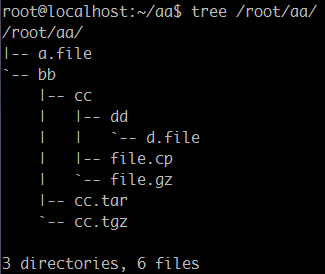
包出只有 file.clone、file.cp 的 tgz 檔？

[root@localhost dd]# tar -zcvf test.tgz file.clone file.cp

file.clone

file.cp

**tree快速列出(製作)樹狀目錄結構清單(tree)**



**find目錄中搜索文件，它的使用權限是所有用戶**

可搭配

－maxdepth levels：表示至多查找到開始目錄的第level層子目錄。level是一個非負數，如果level是0的話表示僅在當前目錄中查找。  
－mindepth levels：表示至少查找到開始目錄的第level層子目錄。

-name：支持系統配符\*和?

檔案名選項是find命令最常用的選項，要麼單獨使用該選項，要麼和其他選項一起使用。

可以使用某種檔案名模式來匹配檔，記住要用引號將檔案名模式引起來。

不管當前路徑是什麼，如果想要在自己的根目錄$HOME中查找檔案名符合\*.log的檔，使用~作為 'pathname'參數，波浪號~代表了你的$HOME目錄。

find ~ -name "\*.log" -print

想要在目前的目錄及子目錄中查找所有的‘ \*.log‘檔，可以用：

find . -name "\*.log" -print

想要的目前的目錄及子目錄中查找檔案名以一個大寫字母開頭的檔，可以用：

find . -name "[A-Z]\*" -print

想要在/etc目錄中查找檔案名以host開頭的檔，可以用：

find /etc -name "host\*" -print

想要查找$HOME目錄中的檔，可以用：

find ~ -name "\*" -print 或find . -print

要想讓系統高負荷運行，就從根目錄開始查找所有的檔。

find / -name "\*" -print

如果想在目前的目錄查找檔案名以一個個小寫字母開頭，最後是4到9加上.log結束的文件：

命令：

find . -name "[a-z]\*[4-9].log" -print

用 “find /tmp/aa …”只找出 aa 目錄下第三層的檔案但不包含第一、二層及第四層之後的檔案

[root@localhost aa]# find . -maxdepth 3 -mindepth 3

./bb/cc/file.cp

./bb/cc/dd

./bb/cc/file.gz

**grep是一種強大的文本搜索工具，它能使用正則表達式搜索文本，並把匹配的行打印出來**

命令參數：

-a不要忽略二進制數據。

-A <顯示列數>除了顯示符合範本樣式的那一行之外，並顯示該行之後的內容。

-b在顯示符合範本樣式的那一行之外，並顯示該行之前的內容。

-c計算符合範本樣式的列數。

-C<顯示列數>或-<顯示列數> 除了顯示符合範本樣式的那一列之外，並顯示該列之前後的內容。

-d<進行動作>當指定要查找的是目錄而非文件時，必須使用這項參數，否則grep命令將回報信息並停止動作。

-e<範本樣式>指定字符串作為查找文件內容的範本樣式。

-E將範本樣式為延伸的普通表示法來使用，意味著使用能使用擴展正則表達式。

-f<範本文件>指定範本文件，其內容有一個或多個範本樣式，讓grep查找符合範本條件的文件內容，格式為每一列的範本樣式。

-F將範本樣式視為固定字符串的列表。

-G將範本樣式視為普通的表示法來使用。

-h在顯示符合範本樣式的那一列之前，不標示該列所屬的文件名稱。

-H在顯示符合範本樣式的那一列之前，標示該列的文件名稱。

-i胡列字符大小寫的差別。

-l列出文件內容符合指定的範本樣式的文件名稱。

-L列出文件內容不符合指定的範本樣式的文件名稱。

-n在顯示符合範本樣式的那一列之前，標示出該列的編號。

-q不顯示任何信息。

-R/-r此參數的效果和指定“-d recurse”參數相同。

-s不顯示錯誤信息。

-v反轉查找。

- w只顯示全字符合的列。

-x只顯示全列符合的列。

-o只輸出文件中匹配到的部分。

[root@localhost cc]# grep **^**rc file.cp //grep rc開頭的字 (正規表示法^)

rc

rc0.d

rc1.d

rc2.d

rc3.d

[root@localhost cc]# grep -l yu file.cp file.cp1 file.cp2 //搜尋哪些file 有"yu"

**file.cp**

**file.cp1**

[root@localhost cc]# grep ^yu -n file.cp //列出行數

**156**:yum

**157**:yum.conf

**158**:yum.repos.d

**314**:yum

**315**:yum.conf

**316**:yum.repos.d

**sort 將檔的每一行作為一個單位，相互比較，比較原則是從首字元向後，依次按ASCII碼值進行比較，最後將他們按昇冪輸出。**

命令參數：

-b：忽略每行前面開始出的空格字符

-c：檢查文件是否已經按照順序排序

-d：排序時，處理英文字母、數字及空格字符外，忽略其他的字符

-f：排序時，將小寫字母視為大寫字母

-i：排序時，除了040至176之間的ASCII字符外，忽略其他的字符

-m：將幾個排序號的文件進行合併

-M：將前面3個字母依照月份的縮寫進行排序

-n：依照數值的大小排序

-o<輸出文件>：將排序後的結果存入製定的文件

-r：以相反的順序來排序 // sort預設的排序方式是昇冪，如果想改成降冪，就加個-r

-t<分隔字符>：指定排序時所用的欄位分隔字符； +<起始欄位>-<結束欄位>：以指定的欄位來排序，範圍由起始欄位到結束欄位的前一欄位。  
-u：刪除重複行

可參考

http://charleslin74.pixnet.net/blog/post/419884702-%5Blinux%5D-%E6%8E%92%E5%BA%8F-sort%E7%9A%84%E7%94%A8%E6%B3%95

**uniq命令用於報告或忽略文件中的重複行，一般與sort命令結合使用**命令參數：

-c或——count：在每列旁邊顯示該行重複出現的次數

-d或--repeated：僅顯示重複出現的行列

-f<欄位>或--skip-fields=<欄位>：忽略比較指定的欄位

-s<字符位置>或--skip-chars=<字符位置>：忽略比較指定的字符

-u或——unique：僅顯示出一次的行列

- w <字符位置>或--check-chars=<字符位置>：指定要比較的字符

刪除重複行

uniq file .txt

sort file.txt | uniq

sort -u file.txt

排序後刪除重複再算出行數

sort file.cp | uniq | wc -l

[root@localhost cc]# sort -u file.cp | wc -l

158

找出 **file** 檔案中包含 **“g” d**開頭的資料 並去重複，但不能用 **uniq**

[root@localhost cc]# grep ^g file.cp1 | sort -u

**gai.conf**

**gcrypt**

**gnupg**

**group**

**group-**

**grub.conf**

**gshadow**

**gshadow-**

**we統計指定檔案中的位元組數、字數、行數 工具**

命令參數：

-c 統計位元組數

-l 統計行數

-m 統計字元數。這個標誌不能與 -c 標誌一起使用

-w 統計字數。一個字被定義為由空白、跳格或換行字元分隔的字串

-L 列印最長行的長度

查看檔案的位元組數、字數、行數

[root@localhost cc]# wc file.cp

316 316 2886 file.cp

[root@localhost cc]# grep ^g file.cp1 | sort -u | wc -l //統計行數

8

[root@localhost cc]# grep ^g file.cp1 | sort -u | wc -m //統計字元數

62

[root@localhost cc]# grep ^g file.cp1 | sort -u | wc -c //統計位元組數

62

**awk將每一列資料視為一筆記錄（record），每筆記錄以欄位分 隔符號分成若干欄位，然後，輸出各個欄位的值**

awk「樣式」檔案：把符合樣式的資料列顯示出來

awk '{動作}' 檔案：對每一列都執行{}中的動作

awk '樣式{動作}' 檔案：對符合樣式的資料列，執行{}中的動作

awk ‘{print $NF}’ test //$ NF僅打印記錄中的最後一列

awk 的用法 1： awk '/La/' dataf3

顯示含 La 的資料列

awk 的用法 2： awk '{ print $1, $2 }' dataf3

顯示 dataf3 每一列的第 1 和第 2 個欄位。 $1 代表第 1 個欄位，$2 代表第二欄位，其他依此類推

awk 的用法 3： awk '/La/{ print $1, $2 }' dataf3

將含有 La 關鍵字的資料列的第 1 及第 2 個欄位顯示出來

awk 的用法 4： awk -F: '/^ols3/{ print $3, $4 }' /etc/passwd

使用選項 -F，指定:為分隔字元，帳號 ols3 的 uid（第 3 欄位）及 gid（第 4 欄位）顯示出來

[root@localhost ~]# cat awk.txt | head -n 5

[2017/08/03 01:48:56] [INF][20325][imap4d:userlog]# goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 03:48:57] [INF][31269][imap4d:userlog]# goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 04:49:11] [INF][22250][imap4d:userlog]# goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 07:18:47] [INF][31888][imap4d:userlog]# goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 08:19:10] [INF][24932][imap4d:userlog]# goldalex@alt.idv.tw 5 114.136.41.23

//只取紅字地方

[root@localhost ~]# awk '{print $1 ,$2 ,$(NF-2),$(NF-1),$(NF)}' awk.txt | head -5

[2017/08/03 01:48:56] goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 03:48:57] goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 04:49:11] goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 07:18:47] goldalex@alt.idv.tw 5 60.248.35.228

[2017/08/03 08:19:10] goldalex@alt.idv.tw 5 114.136.41.23

// $NF代表 ：最后一個Field(列) 要框起來

[root@localhost ~]# awk '{printf "%-10s %-10s %-20s %-3s %-10s\n" ,$1,$2,$(NF-2),$(NF-1),$NF}' awk.txt | head -10 | tail -5

//調整固定欄寬

[2017/08/03 10:09:24] alex@alt.idv.tw 3 125.227.53.48

[2017/08/03 10:18:40] goldalex@alt.idv.tw 5 125.227.53.46

[2017/08/03 10:18:41] goldalex@alt.idv.tw 5 125.227.53.46

[2017/08/03 10:19:39] alex@alt.idv.tw 3 125.227.53.48

[2017/08/03 10:56:51] goldalex@alt.idv.tw 5 125.227.53.46

**sed 比對每一資料列，若符合樣式， 就執行指定的動作。**

sed 的語法如下： sed '樣式命令' 檔案 它的意思是說：如果檔案中某一列符合「樣式」，就執行指定的 sed 命令，例如 刪除（d）或取代（s）

這裡的「樣式」使用一對//含括，表示尋找之意；也可以指定資料列的範圍，例 如：1,6 表作用範圍是由第 1 列到第 6 列；/AAA/,/DDD/表作用範圍是從含有 AAA 的資料列，到含有 DDD 的資料列

特別注意：sed 並不會更改檔案內容

sed 的工作方式是讀取檔案內容，經串流 編輯之後，把結果顯示到標準輸出。因此，如果想要儲存 sed 的處理結果，得自行運用轉向輸出將結果存成其他檔案

範例

sed -n '/La/p' dataf3

-n 會抑制 sed 秀出其他資料列的預設動作，只顯示符合樣式的資料列

sed -n 's/La/Oo/p' dataf3

這裡的 s 是取代的意思，第一對 //中含括的字串（La）是搜尋的目標，第二對 //含括的是置換的字串（Oo）。它會把資料列中的字串 La 換成 Oo

請注意：上面這個指令，只會更換第一個出現的 La ，全部置換，要加上全域的命令 g

sed -n 's/La/Oo/gp' dataf3

這樣就會把所有找到的 La 全換成 Oo 了。

//延伸取代方法 可以看pdf <sed 和 awk 入門>

awk & sed 可參考

http://www.delightpress.com.tw/bookRead/skns00004\_read.pdf

http://wanggen.myweb.hinet.net/ach3/ach3.html?MywebPageId=2017311504159794084#sed\_and\_awk

http://netkiller.sourceforge.net/shell/awk.html

//或是資料夾的sed 和 awk 入門