Grep

Les coses que no siguem capaços de recordar les hem de guardar al nostre ordinador, diu una màxima de l'usuari informàtic. Aquesta no és una mala idea, però, a diferència del que passa amb la memòria humana, que normalment recuperarà la informació emmagatzemada de forma fidedigna (excepte en els exàmens, és clar), no sempre és tan fàcil trobar informació al nostre disc dur. Podem perdre molt temps a trobar un arxiu del que hem oblidat el seu nom o on va ser emmagatzemat. Fins i tot el saber exactament un arxiu conté la informació que busquem pot ser de poca ajuda en cas d'arxius de text grans.

L'ordre del shell **grep**, que localitza cadenes de text en fitxers, és útil en tots dos casos. En la situació més simple, podem executar grep amb la tecla de cerca i l'arxiu a buscar. **grep** ens mostrarà totes les línies a l'arxiu especificat que continguin el text buscat. Imaginem que volem buscar a l'arxiu llibre.txt el text "Edèn". Hem escriure

```
$ grep Edèn llibre.txt
```

a la línia d'ordres i grep ens mostrarà els passatges apropiats de l'arxiu.

Si la cerca conté espai hem d'utilitzar cometes. Per exemple:

```
$ grep "Jardí de l'Edèn" llibre.txt
```

Hem de parar atenció i esperar errors quan usem caràcters especiales: *****,**?** i **!** tenen un significat especial per a la línia d'ordres. Un altre grup de caràcters **(.**, *****, **^**, **$** i \) no s'interpreta per grep tal qual. En el seu lloc, l'eina suposarà una expressió regular. El costat positiu és que això ens permet construir cerques molt potents i complexes, si bé és possible que preferim evitar aquests caràcters amb grep fins que ens sentim a gust amb l'eina.

Si no sabem quin arxiu conté el text que busquem, executem grep amb un comodí. L'obra "Moby Dick" de Herman Melville està composta d'una col·lecció d'arxius de text.

```
$ grep blanc moby.*
```

Ens mostrarà totes les coincidències amb la paraula **blanc** que apareixen en aquesta obra de Melville (veure Figura 1).

L'asterisc significa "qualsevol grup de lletres". La interfície canviarà aquesta expressió pel nom de qualsevol arxiu en el directori actual que comenceu amb els caràcters moby.

Si els arxius en els que volem cercar estan distribuïts per diversos directoris pel disc durs, hem d'afegir a la línia l'opció **-r** (recursive) per indicar a grep que busqui en una carpeta completa:

```
$ grep -r blanc Melville/obres/
```

buscarà blanc al directori obres i en tots els subdirectoris que pengin d'ell.

Grep de un Vistazo

Comando	Acción
grep patrón fichero	búsca un patron en un fichero
*grep patrón *.htm*	busca en todos los archivos en un directorio que acaban con el sufijo .htm
grep -r patrón dir	realiza una búsqueda recursiva en un directorio y sus subdirectorios.
grep -i patrón fichero	ignora la diferencia entre fichero mayúsculas y minúsculas.
grep -A n	muestra las siguientes n líneas tras la línea que contiene la búsqueda.

El trio: ps, grep i kill.

grep no és només útil per a recerques de text filosòfics i teològics, sinó que pot ser combinat amb altres ordres del shell. Si la resposta d'un comandament produeix molt de text, grep pot ser utilitzat escrivint després de la instrucció el caràcter "|" i **grep text_a_buscar** per a filtrar el resultat i mantenir només les parts en què estem interessats. Un cas típic és aquell en el qual faríem servir grep per a tancar un programa que s'ha bloquejat.

La comanda **ps ax** ens mostra els processos actius. Podem utilitzar grep per aplicar un filtre i trobar només el programa que estem buscant, Mozilla per exemple:

$ **ps ax | grep mozilla**
2500? S 1:40 / usr/lib/mozilla-1.3/mozilla-bin
5645 pts / 4 S 0:00 grep mozilla

grep ens mostra dues coincidències que contenen Mozilla: el cercador i el propi grep. La part que ens interessa per a tancar el programa apareix al principi de cada línia: l'ID del procés. Ara podem escriure **kill -9 2500** per a tancar el programa desitjat.

Com que els gurús del shell solen ser notablement mandrosos, necessitem trobar un mètode per no haver d'escriure aquesta comanda cada vegada que ho requerim: en altres paraules, necessitem un àlies. Els àlies definits han de ser guardats a l'arxiu **.bashrc** al directori d'usuari. Aquest arxiu s'executa cada vegada que obrim un shell de comandaments interactiu. Farem servir el nostre editor preferit per obrir el fitxer, per exemple

$ nano ~/.bashrc

Ara podem introduir el nostre àlies en l'última línia de **.bashrc**. En lloc de PSS podem utilitzar qualsevol altre nom fàcil de recordar, però evitant usar el nom d'una comanda existent:

alias PSS = "ps ax | grep"

Observa que PSS és unes ordre inacabada, al "grep" li falta la cadena a buscar. L'afegirem quan executem l'alias afegint al final la cadena: **PSS elquesigui** executarà **ps ax | grep elquesigui**

Per a utilitzar el nostre nou àlies al shell necessitem executar el fitxer de configuració. Per a fer-ho hem d'escriure:.

$ ~/.bashrc

o simplement tancant i obrint el terminal (des d'entorn gràfic)

Ara podem usar la comanda **PSS nomprograma** per a buscar un programa actiu.

Exemple. Llibre d'Adreces

grep és una eina tremendament flexible. Una aplicació interessant per a grep és un senzill llibre d'adreces.

Tot el que necessitem per al nostre "programa" són els comandaments **grep**, **alias** i **cat** i un fitxer de text on hem anat emmagatzemant els noms, números de telèfon, adreces postals i les adreces de correu electrònic dels nostres amics, coneguts i parents. Una entrada podria tenir el següent aspecte:

Charly Pingüí
+12345 678
tux@linux.org
C/ del Pol Sud
Vilatux, Antàrtica

Ara hem de guardar l'arxiu amb el nom adreces al nostre directori arrel i afegir el següent àlies a **.bashrc**:

alias tel="cat ~/direccions | grep -i -A 4"
o
alias tel="grep -i -A 4 <direccions"

La comanda **cat** ens mostra el contingut del fitxer adreces. El caràcter | envia aquest resultat a grep. L'opció **-i** garanteix que la recerca no diferenciarà entre majúscules i minúscules.

Finalment, **-A 4** indica a grep que es mostrin les 4 línies següents a la primera coincidència amb la recerca.

Novament. ~/.bashrc executarà el nostre arxiu de configuració. En el futur, només necessitarem escriure **tel** nom a la shell per a recuperar la direcció de la persona que desitgem buscar.

Exemple pràctic

Emprant combinacions d'ordres anem a extreure la informació de la xarxa. Suposem que tenim una màquina amb dues targetes de xarxa: eth0 i eth1

```
armand@ubuntu:/home/linux$
armand@ubuntu:/home/linux$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:99:2c:4b
          inet addr:192.168.1.149  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe99:2c4b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:137 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17328 (17.3 KB)  TX bytes:8306 (8.3 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:b0:9c:77
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb0:9c77/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:94 errors:0 dropped:0 overruns:0 frame:0
          TX packets:128 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12164 (12.1 KB)  TX bytes:16440 (16.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:165 errors:0 dropped:0 overruns:0 frame:0
          TX packets:165 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11525 (11.5 KB)  TX bytes:11525 (11.5 KB)
```

Volem crear un alias que ens extregui la línia que contingui l'adreça ip de la targeta indicada.

Obtenim tota la informació. Si emprem grep podem extreure un dispositiu. Per exemple

```
armand@ubuntu:/home/linux$ ifconfig | grep -A7 eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:b0:9c:77
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb0:9c77/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:119 errors:0 dropped:0 overruns:0 frame:0
          TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13900 (13.9 KB)  TX bytes:18160 (18.1 KB)
armand@ubuntu:/home/linux$ _
```

Extreurà les 7 línies següents a la línia on troba "eth1". Observa que això ja ho fa "ifconfig eth1". Aquest exemple és simplement il·lustratiu

Creem un alias

```
armand@ubuntu:/home/linux$ alias net="ifconfig | grep -A7"
armand@ubuntu:/home/linux$
armand@ubuntu:/home/linux$ net eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:99:2c:4b
          inet addr:192.168.1.149  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe99:2c4b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:217 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26377 (26.3 KB)  TX bytes:8888 (8.8 KB)
armand@ubuntu:/home/linux$
armand@ubuntu:/home/linux$ net eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:b0:9c:77
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb0:9c77/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:168 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19629 (19.6 KB)  TX bytes:19142 (19.1 KB)
armand@ubuntu:/home/linux$ _
```

Ara emprant head i tail podríem extreure únicament la línia que te l'adreça IP.

```
armand@ubuntu:/home/linux$ net eth1 | head -n2
eth1      Link encap:Ethernet  HWaddr 08:00:27:b0:9c:77
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
armand@ubuntu:/home/linux$
armand@ubuntu:/home/linux$ net eth1 | head -n2 | tail -n1
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
armand@ubuntu:/home/linux$
```

I si volem crear un alias amb el head i tail

```
armand@ubuntu:/home/linux$ alias ip="head -n2 | tail -n1"
armand@ubuntu:/home/linux$
armand@ubuntu:/home/linux$ net eth1 | ip
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
armand@ubuntu:/home/linux$ _
```

Un altre exemple

Volem construir un detector de preus d'una botiga online. Amb les eines que coneixem no podem fer massa cosa, però si ens podem aproximar. Per exemple volem control·lar el preu d'un producte concret:

```
armand@IJMSERVER:~/proves$ wget http://www.pccomponentes.com/leotec_titanium_t355_5_5__4g_negro_libre.html  >/dev/null
--2016-01-13 14:03:54--  http://www.pccomponentes.com/leotec_titanium_t355_5_5__4g_negro_libre.html
Resolviendo www.pccomponentes.com (www.pccomponentes.com)... 162.159.255.66, 162.159.254.66
Conectando con www.pccomponentes.com (www.pccomponentes.com)[162.159.255.66]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [text/html]
Grabando a: “leotec_titanium_t355_5_5__4g_negro_libre.html”

    [ <=>                                                                                ] 238.520     --.-K/s   en 0,1s

2016-01-13 14:03:54 (2,14 MB/s) - “leotec_titanium_t355_5_5__4g_negro_libre.html” guardado [238520]

armand@IJMSERVER:~/proves$
```

Aquesta primera ordre desa un fitxer amb el resultat de la pàgina a control·lar. I acontinuació , amb l'ajuda del grep, extreiem la línia amb el preu del producte

```
armand@IJMSERVER:~/proves$ cat *html | grep "importe_articulo"
"importe_articulo":119,
armand@IJMSERVER:~/proves$
```

I si volem apurar un pel més, extreient el valor numèric:

```
armand@IJMSERVER:~/proves$ cat *html | grep "importe_articulo" | tail -c7
:119,
armand@IJMSERVER:~/proves$ cat *html | grep "importe_articulo" | tail -c6
119,
armand@IJMSERVER:~/proves$ cat *html | grep "importe_articulo" | tail -c6 | head -c3
119armand@IJMSERVER:~/proves$
```

Queda enganxat, però això ho podem desar en una variable

```
armand@IJMSERVER:~/proves$
armand@IJMSERVER:~/proves$ preu=`cat *html | grep "importe_articulo" | tail -c6 | head -c3`
armand@IJMSERVER:~/proves$
armand@IJMSERVER:~/proves$ echo $preu
119
armand@IJMSERVER:~/proves$
```

Ara un alias:

```
armand@IJMSERVER:~/proves$ alias elmeupreu="preu=`cat *html | grep 'importe_articulo' | tail -c6 | head -c3`;echo $preu"
armand@IJMSERVER:~/proves$ elmeupreu
119
armand@IJMSERVER:~/proves$
```

Expressions regulars

Les expressions regular ens permeten afinar en el procés de cerca amb l'ordre grep (bé, les expressions regulars poden emprar-se amb altres ordres, però de moment ens centrarem amb grep)

Veiem una llista d'exemples:

Patró	**Què representa**
gato	La cadena gato
^gato	La cadena gato al començament de la línia
gato$	La cadena gato al final de la línia
^gato$	La cadena gato formant una única línia
gat[ao]	La paraula gata o gato
ga[^aeiou]o	La tercera lletra no és una vocal minúscula
ga.o	La tercera lletra és qualsevol caràcter
^....$	Qualsevol línia que contingui 4 caràcters qualsevols
^\.	Qualsevol línia que comenci per punt
^[^.]	Qualsevol línia que no comenci per punt.
gatos*	gato gatos gatoss gatosss , etc ..
"gato"	gato entre cometes dobles.
"*gato"*	gatos amb o sense comentes dobles
[a-z][a-z]*	línia que contingui una o més lletres minúscules
[^0-9A-Z]	Qualsevol caràcter que no sigui ni número ni lletra majúscula
[Ax5]	Qualsevol caràcter que sigui A , x o 5
gato|gota|gata	Qualsevol línia que contingui gato, gota o gata
(s|arb)usto	Que contingui susto o arbusto
ga?t [oa]	gato, gata, gasto, gaita, etc.
\<ga	Qualsevol paraula que comenci per ga.
to\>	Qualsevol paraula que acabi per to.
\<gato\>	La paraula gato.
o\{2, \}	Dos o més 'o' consecutius en una mateixa línia.

A continuació posarem una sèrie d'exemples fent us de grep i d'expressions regulars conjuntament. Amb això pretenem deixar més clar l'ús d'expressions regulars Treballarem amb un arxiu anomenat datos, que conté el següent:

```
gato      libro     atunn     gotas     atas
pez       gaita     ##%%      dado      oso
.exrc     expreso   atún      gota      loco
Gato      tierra    Gata      nada      ratón
gata      canica    atunnn    fuente    gatos
fin
```

Busca la paraula gato

$ grep gato datos

```
gato    libro    atunn   gotas   atas
gata    canica   atunnn  fuente  gatos
$
```

Línies que comencen per gato

$ grep "^gato" datos
gato libro atunn gotas atas
$

Que continguin gata o gato

$ grep "gat[ao]" datos
gato libro atunn gotas atas
gata canica atunnn fuente gatos
$

Línies que continguin únicament tres caràcters

$ grep "^. . .$" datos
fin $

Que continguin una o més lletres majúscules

$ grep "[A-Z][A-Z]*" datos
GAto tierra Gata nada ratón

Línies que comencin per punt

$ grep "^\." datos
.exrc expreso atún gota loco
$

Si ara volem les línies que no comencin per punt

$ grep "^[^.]"· datos

gato	libro	atunn	gotas	atas
pez	gaita	##%%	dado	oso
Gato	tierra	Gata	nada	raton
gata	canica	atunnn	fuente	gatos
fin				

Línies que acabin per n

$ grep "n$" datos
Gato tierra Gata nada ratón
fin

Que continguin 3 o mes 'n' seguides:

$ grep n\{3,\} datos

gata canica atunnn fuente gatos

Que continguin,exactament, 3 'n' seguides

$ grep n\{3\} datos
gata canica atunnn fuente gatos

Que continguin una a, seguida de qualsevol caràcter i a continuació una o

$ grep "a.o" datos

gato libro atunn gotas atas
pez gaita ##%% dado oso
Gato tierra Gata nada raton
gata canica atunnn fuente gatos

Mostrar únicament els directoris de /usr, fent ús de | (pipe o canonada)

$ ls -l /usr | grep "^d"
drwxr-xr-x 2 root root 65536 2011-03-24 12:08 bin
drwxr-xr-x 2 root root 4096 2010-10-26 11:29 games
drwxr-xr-x 43 root root 20480 2011-02-14 09:45 include
drwxr-xr-x 259 root root 86016 2011-03-24 12:08 lib
drwxr-xr-x 41 root root 36864 2011-01-28 10:37 lib32
drwxr-xr-x 13 root root 4096 2010-10-24 18:54 local
drwxr-xr-x 2 root root 12288 2011-03-20 12:13 sbin
drwxr-xr-x 377 root root 12288 2011-03-24 12:08 share
...

Un altre exemple, intenta esbrinar que fa!!!

$ ls -lF /bin | grep 's*$'