LPI 103.7 Search text files using regular expressions

Curs 2021 - 2022

ASIX M01-ISO LPI 103-GNU_and_unix_commands

Regular expressions

Description

Key Concepts:
- ❏ basic regular expressions
- ❏ exteded regular expressions

Commands:
- ❏ grep, grep -i, grep -v, grep -E, egrep
- ❏ cut
- ❏ sort
- ❏ tr
- ❏ expand/unexpand
- ❏ sed
- ❏ paste
- ❏ join

Regular expressions

Using grep with regular expressions

grep

- ● grep pattern file...
- ● -i ignore case
- ● -v not match
- ● -w word
- ● -n num
- ● -c count
- ● -A [N] after lines
- ● -B [N] before lines
- ● -E extended
- ● -q no output
- ● -l list filename only

```
$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
```

```
$ grep 100 /etc/passwd
games:x:12:100:games:/usr/games:/sbin/nologin
guest:x:1000:1000:guest:/home/guest:/bin/bash
ecanet:x:1001:1001::/home/ecanet:/bin/bash
```

```
$ cat > noms.txt
pere
Anna
MARTA
pau
joan
Julio
jordi
miquel
^d
```

```
$ grep "anna" noms.txt

$ grep "Anna" noms.txt
Anna

$ grep -i "Anna" noms.txt
Anna
```

```
$ grep -v  j noms.txt
pere
Anna
MARTA
pau
Julio
miquel

$ grep -v -i j noms.txt
pere
Anna
MARTA
pau
miquel
```

```
$ grep guest /etc/* 2> /dev/null
/etc/group:guest:x:1000:
/etc/group-:guest:x:1000:
Binary file /etc/ld.so.cache matches
/etc/libguestfs-tools.conf:# /etc/libguestfs-tools.conf or .libguestfs-tools.rc
/etc/libguestfs-tools.conf:# See libguestfs-tools.conf(5) man page for documentation.
/etc/libguestfs-tools.conf:# Uncomment the following line to make guestfish, guestmount,
/etc/passwd:guest:x:1000:1000:guest:/home/guest:/bin/bash
/etc/passwd-:guest:x:1000:1000:guest:/home/guest:/bin/bash
/etc/subgid:guest:100000:65536
/etc/subgid-:guest:100000:65536
/etc/subuid:guest:100000:65536
/etc/subuid-:guest:100000:65536
```

```
$ grep -c bin /etc/passwd
51

$ grep -w bin /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
guest:x:1000:1000:guest:/home/guest:/bin/bash
ecanet:x:1001:1001::/home/ecanet:/bin/bash
```

```
$ grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
10:operator:x:11:0:operator:/root:/sbin/nologin

$ grep -c root /etc/passwd
2
```

```
$ grep operator /etc/passwd
operator:x:11:0:operator:/root:/sbin/nologin

$ grep -A 2 operator /etc/passwd
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

$ grep -B 3 operator /etc/passwd
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

Basic regular expressions

- [char] 1 char from the charset
- ^   begin line
- $ end line
- .   1 character
- _*   0-n occurrences precedent char

```
$ grep "home" /etc/passwd
guest:x:1000:1000:guest:/home/guest:/bin/bash
ecanet:x:1001:1001::/home/ecanet:/bin/bash

$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

$ grep ^root /etc/passwd
root:x:0:0:root:/root:/bin/bash

$ grep bash$ /etc/passwd
root:x:0:0:root:/root:/bin/bash
guest:x:1000:1000:guest:/home/guest:/bin/bash
ecanet:x:1001:1001::/home/ecanet:/bin/bash
$ grep [jJm]oan noms.txt
joan

$ echo "pere pare pire pore pure" | grep "p[aou]re"
pere pare pire pore pure

$ echo "nom1 nom2 nom3 nom8 nom9 noma nomb nomc" | grep "nom[0-8]"
nom1 nom2 nom3 nom8 nom9 noma nomb nomc

$ grep [0-3] /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```

```
$ echo pim pam pum repim | grep p.m
pim pam pum repim
```

```
$ echo piim paamp pumpum repiim | grep p.m
piim paamp pumpum repiim

$ echo piim paamp pumpum repiim | grep p..m
piim paamp pumpum repiim

$ echo "name-23 name.45 name/34" | grep name.
name-23 name.45 name/34
```

```
$ echo namea:nameaa:nameaaa:nameaaa: | grep "a"
namea:nameaa:nameaaa:nameaaa:

$ echo namea:nameaa:nameaaa:nameaaa: | grep "aa"
namea:nameaa:nameaaa:nameaaa:

$ echo namea:nameaa:nameaaa:nameaaa: | grep "a."
namea:nameaa:nameaaa:nameaaa:

$ echo namea:nameaa:nameaaa:nameaaa: | grep "aa*"
namea:nameaa:nameaaa:nameaaa:

$ echo namea:nameaa:nameaaa:nameaaa: | grep "a*"
```

```
$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep "a."
aa:bb:ab:a:bb:aaa:bb:aaaa

$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep "aa"
aa:bb:ab:a:bb:aaa:bb:aaaa

$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep "aa*"
aa:bb:ab:a:bb:aaa:bb:aaaa

$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep "aaa*"
aa:bb:ab:a:bb:aaa:bb:aaaa
```

```
$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep ":a:"
aa:bb:ab:a:bb:aaa:bb:aaaa

$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep ":a.:"
aa:bb:ab:a:bb:aaa:bb:aaaa

$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep ":aa:"

$ echo aa:bb:ab:a:bb:aaa:bb:aaaa | grep ":aa*:"
aa:bb:ab:a:bb:aaa:bb:aaaa
```

Extended regular expressions

- _+   1 or + precedent char
- _?   0 or 1 precedent char
- {} {n} {n,} {1,m} {n,m}
- |
- ()

- equivalences:
  - char{0,n} char*
  - char{1,n} char+
  - char{0,1] char?

```
## 1
$ echo :user:user1:user2:user11:user221: | grep ":user[0-9]:"
:user:user1:user2:user11:user221:

$ echo :user:user1:user2:user11:user221: | grep -E ":user[0-9]+"
:user:user1:user2:user11:user221:

$ echo :user:user1:user2:user11:user221: | grep -E ":user[0-9]?"
:user:user1:user2:user11:user221:
```

```
## 2
$ cat > users.txt
user
pere
user1
marta
user2
user11
ramon
user221
usera
userb
```

```
##3
$ grep "user" users.txt
user
user1
user2
user11
user221
usera
userb

$ grep "user[0-9]$" users.txt
user1
user2

$ grep -E "user[0-9]?$" users.txt
user
user1
user2

$ grep -E "user[0-9]+$" users.txt
user1
user2
user11
user221
```

```
##4
$ grep -E "user[0-9][0-9]$" users.txt
user11

$ grep -E "user[0-9]{2}$" users.txt
user11

$ grep -E "user[0-9]{2,}$" users.txt
user11
user221

$ grep -E "user[0-9]{,2}$" users.txt
user
user1
user2
user11

$ grep -E "user[0-9]{2,3}$" users.txt
user11
user221
```

```
#5
$ echo "dni:66454753S" | grep -E ":[0-9]{8}"
dni:66454753S

$ echo "dni:664547530000S" | grep -E ":[0-9]{8}"
dni:664547530000S
$ echo "dni:66454753S" | grep -E ":[0-9]{8}$"

$ echo "dni:66454753S" | grep -E ":[0-9]{8}[A-Z]$"
dni:66454753S

$ echo "dni:664547530000S" | grep -E ":[0-9]{8}$"
```

```
#6
$ grep -E "root|operator" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin

$ grep -E "1$|2$" users.txt
user1
user2
user11
user221
```

```
#7
# validate data format dd-mm-aaaa
$ echo "22-11-2021" | grep -E "^[0-9]{2}-[0-9]{2}-[0-9]{4}$"
22-11-2021

# Validate car format
$ echo "7774MM" | grep -E "[0-9]{4}[A-Z]{2}"
7774MM

$ echo "77749999MMaaaa" | grep -E "[0-9]{4}[A-Z]{2}"
77749999MMaaaa

$ echo "7774MM" | grep -E "^[0-9]{4}[A-Z]{2}$"
7774MM

# validate data format dd-mm-aaaa and dd/mm/aaaa
...
```

```
# 8
$ echo "Mis Miss Mississ tresMiss" | grep -E "M(iss)*"
Mis Miss Mississ tresMiss

$ echo "192.168.12.15" | grep -E "^([0-9]{1,3}\.){3}[0-9]{1,3}"
192.168.12.15
```

Basic text filters

- head
- tail
- expand/unexpand
- cut
- sort
- tr

head

```
#1  head

$ cat noms.txt
pere
Anna
MARTA
pau
joan
Julio
jordi
miquel

$ wc -l noms.txt
8 noms.txt
$
$ head -n 2 noms.txt
pere
Anna

$ head -n -3 noms.txt
pere
Anna
MARTA
pau
joan
```

tail

```
# 2
$ tail -n 2 noms.txt
jordi
miquel
```

expand/unexpand

```
#3
$ echo "text  with  two spaces" | unexpand -a -t2
text    with    two spaces

$ echo -e "text\twith\ttabulations"  | expand  -t 2
text  with  tabulations
```

cut

```
#4
$ cut -d: -f1,3,4 /etc/passwd | head -n5
root:0:0
bin:1:1
daemon:2:2
adm:3:4
lp:4:7

$ cut -d: -f1-2,4,6- /etc/passwd | head -n5
root:x:0:/root:/bin/bash
bin:x:1:/bin:/sbin/nologin
daemon:x:2:/sbin:/sbin/nologin
adm:x:4:/var/adm:/sbin/nologin
lp:x:7:/var/spool/lpd:/sbin/nologin
```

```
#5
```

```
$ ls -l | cut -c2-9
otal 8
rw-rw-r-
rw-rw-r-
```

sort | uniq

- lexicographic order (default)
- -r reverse
- -n -g -H numeric order
- -t_ delimiter
- -knº key field
- -u (uniq)

```
#6
$ sort /etc/passwd | head -n5
abrt:x:173:173::/etc/abrt:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
akmods:x:977:969:User is used by akmods to build akmod
packages:/var/cache/akmods/:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin

$ sort -r noms.txt
pere
pau
miquel
MARTA
Julio
jordi
joan
Anna
```

```
#7
$ sort -t: -k3 /etc/passwd | head -n5
root:x:0:0:root:/root:/bin/bash
guest:x:1000:1000:guest:/home/guest:/bin/bash
ecanet:x:1001:1001::/home/ecanet:/bin/bash
qemu:x:107:107:qemu user:/:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin

$ sort -t: -k3n /etc/passwd | head -n5
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

$ sort -t: -k4n -k3nr /etc/passwd | head -n10
operator:x:11:0:operator:/root:/sbin/nologin
halt:x:7:0:halt:/sbin:/sbin/halt
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
sync:x:5:0:sync:/sbin:/bin/sync
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```

```
#8
$ cat > bdades.txt
anna:12:barcelona
pere:5:girona
```

```
pau:33:girona
julia:22:barcelona
marta:22:girona

$ sort -t: -k2n bdades.txt
pere:5:girona
anna:12:barcelona
julia:22:barcelona
marta:22:girona
pau:33:girona

$ sort -t: -k3 -k2n bdades.txt
anna:12:barcelona
julia:22:barcelona
pere:5:girona
marta:22:girona
pau:33:girona
```

```
#9
$ cut -d: -f4 /etc/passwd | head -n 5
0
1
2
4
7

$ cut -d: -f4 /etc/passwd | wc -l
51

$ cut -d: -f4 /etc/passwd | sort | uniq -c | head -n 5
      5 0
      1 1
      1 100
      1 1000
      1 1001

$ cut -d: -f4 /etc/passwd | sort -u | head -n 5
0
1
100
1000
1001
```

```
#10
$ cat bdades.txt
anna:12:barcelona
pere:5:girona
pau:33:girona
julia:22:barcelona
marta:22:girona

$ cut -d: -f2  bdades.txt | sort -n
5
12
22
22
33

$ cut -d: -f2  bdades.txt | sort -n -u
5
12
22
33

$ cut -d: -f2  bdades.txt | sort -n | uniq -c
      1 5
      1 12
      2 22
      1 33
```

tr | normalize

- char to char (no text)
- delete
- squeeze
- normalize

```
#11
$ echo "supersecretpassword" | tr 'aeiou' '43701'
s1p3rs3cr3tp4ssw0rd

$ echo "mama meva maravella" | tr 'mama' 'papa'
papa peva paravella

$ echo "text in lower case" | tr [a-z] [A-Z]
TEXT IN LOWER CASE

$ echo "text in lower case" | tr 'ex' '9X'
t9Xt in low9r cas9

$ echo "la mr estava salada, salada estava la mar" | tr a i
li mr estivi silidi, silidi estivi li mir

$ echo "secret number is 1111" | tr [0-9] X
secret number is XXXX

$ echo "pere,marta,anna"
pere,marta,anna

$ echo "pere,marta,anna" | tr ',' '\n'
pere
marta
anna

$ head -n1 /etc/passwd
root:x:0:0:root:/root:/bin/bash

$ head -n1 /etc/passwd | tr ':' '\t'
root    x       0       0       root    /root   /bin/bash
```

```
#12
$ tr [a-z] [A-Z] < noms.txt
PERE
ANNA
MARTA
PAU
JOAN
JULIO
JORDI
MIQUEL

$ tr [a-z] [A-Z] < noms.txt  > upper.txt

$ tr [A-Z] [a-z] < upper.txt > upper.txt
$ cat upper.txt
```

```
#13
$ echo "deleted text witout letter e" | tr -d e
dltd txt witout lttr

$ tr -d 'aeiou' < noms.txt
pr
Ann
MARTA
p
jn
Jl
```

11

```
jrd
mql
```

```
#14
$ echo "text  with  unniform    espaces in it  " | tr -s ' '
text with unniform espaces in it

$ echo "text  with  unniform    espaces in it  " | tr ' ' ':'
text::with::unniform::::::espaces:in:it::

$ echo "text  with  unniform    espaces in it  " | tr --squeeze ' ' ':'
text:with:unniform:espaces:in:it:
```

```
$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs"
text     with                  uniform  spaces and              tabs

$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs" | tr ' \t' ':'
text:::with:::::::uniform::spaces:and::::tabs

$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs" | tr -s ' \t' ':'
text:with:uniform:spaces:and:tabs

$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs" | tr -s ' \t' ' '
text with uniform spaces and tabs

$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs" | tr -s ' \t' '\t'
text    with    uniform spaces and    tabs

$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs" | tr -s '[:blank:]' '\t'
text    with    uniform spaces and    tabs

$ echo -e "text \t with \t\t\t  uniform  spaces and \t\t tabs" | tr -s '[:blank:]' ' '
text with uniform spaces and tabs
```

```
#15
$ grep -v "#"  /etc/fstab
/dev/mapper/fedora-root /                     ext4    defaults       1 1
UUID=027d32d8-de03-40d5-ad80-453618bd71a8 /boot              ext4    defaults
1 2
UUID=7F70-2AC1          /boot/efi            vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/fedora-home /home                ext4    defaults       1 2
/dev/mapper/fedora-swap swap                 swap    defaults       0 0

$ grep -v "#"  /etc/fstab | tr -s [:blank:] ' '
/dev/mapper/fedora-root / ext4 defaults 1 1
UUID=027d32d8-de03-40d5-ad80-453618bd71a8 /boot ext4 defaults 1 2
UUID=7F70-2AC1 /boot/efi vfat umask=0077,shortname=winnt 0 2
/dev/mapper/fedora-home /home ext4 defaults 1 2
/dev/mapper/fedora-swap swap swap defaults 0 0
```

```
#16
$ ls / -l | head -n5
total 68
drwxr-xr-x.  2 root root  4096 Apr 16  2021 backups
lrwxrwxrwx.  1 root root     7 Jan 28  2020 bin -> usr/bin
dr-xr-xr-x.  7 root root  4096 Jul  5 20:01 boot
drwxr-xr-x. 23 root root  4320 Oct 21 15:53 dev

$ ls / -l | head -n5 | tr -s [:blank:] '\t'
total  68
drwxr-xr-x.   2     root   root   4096   Apr   16    2021   backups
lrwxrwxrwx.   1     root   root   7      Jan   28    2020   bin    ->    usr/bin
dr-xr-xr-x.   7     root   root   4096   Jul   5     20:01  boot
drwxr-xr-x.   23    root   root   4320   Oct   21    15:53  dev
```

sed: editing text

- -n silent
- -i in place
- s search & replace
- d delete
- p print
- line
- start,end
- /regex/

```
#1
$ head -n4 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin

$ head -n4 /etc/passwd | sed 's/adm/ADM/'
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
ADM:x:3:4:adm:/var/adm:/sbin/nologin

$ head -n4 /etc/passwd | sed 's/adm/ADM/g'
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
ADM:x:3:4:ADM:/var/ADM:/sbin/nologin
```

```
#2
$ echo "hola bon dia, bonet" | sed 's/bon/MOLT BON/'
hola MOLT BON dia, bonet

$ echo "hola bon dia, bonet" | sed 's/bon/MOLT BON/g'
hola MOLT BON dia, MOLT BONet
```

```
#3
$ head -n 4 /etc/passwd | sed 's/bin//g'
root:x:0:0:root:/root://bash
:x:1:1::/:/s/nologin
daemon:x:2:2:daemon:/s:/s/nologin
adm:x:3:4:adm:/var/adm:/s/nologin

$ echo "some words may not be words woth to say" | sed 's/word//g'
some s may not be s woth to say
```

```
#4
$ cat noms.txt
pere
Anna
MARTA
pau
joan
Julio
jordi
miquel

$ sed '2,4''d' noms.txt
pere
joan
```

```
Julio
jordi
miquel

$ sed '1,4''d' noms.txt
joan
Julio
jordi
miquel

$ sed '4,$''d' noms.txt
pere
Anna
MARTA
```

```
#5
$ sed '1''p' noms.txt
pere
pere
Anna
MARTA
pau
joan
Julio
jordi
miquel

$ sed '1,4 p' noms.txt
pere
pere
Anna
Anna
MARTA
MARTA
pau
pau
joan
Julio
jordi
miquel

$ sed -n  '1 p' noms.txt
pere

$ sed -n '1,4 p' noms.txt
pere
Anna
MARTA
pau
```

```
#6
$ sed  '1 s/e/XXX/' noms.txt
pXXXre
Anna
MARTA
pau
joan
Julio
jordi
miquel

$ sed -e '1,4 s/a/A/g' noms.txt
pere
AnnA
MARTA
pAu
joan
Julio
jordi
miquel
```

```
$ sed -e '5,$ s/o/000/g' noms.txt
pere
Anna
MARTA
pau
j000an
Juli000
j000rdi
miquel
```

```
#7
$ sed '/e/ d' noms.txt
Anna
MARTA
pau
joan
Julio
jordi

$ sed '/^[jJ]/ d' noms.txt
pere
Anna
MARTA
pau
miquel

$ sed '/^[jJ]/ s/o/000/' noms.txt
pere
Anna
MARTA
pau
j000an
Juli000
j000rdi
miquel
```

```
#8
$ sed -i s/MARTA/marta/ noms.txt

$ cat noms.txt
pere
Anna
marta
pau
joan
Julio
jordi
miquel
```

```
#9
$ echo "935550055"
935550055

$ echo "935550055" | sed -r 's/^[0-9]{2}//'
5550055

$ echo "935550055" | sed -r 's/^[0-9]{2}/(93) /'
(93) 5550055

$ echo "915550055" | sed -r 's/^[0-9]{2}/(93) /'
(93) 5550055

$ echo "935550055" | sed -r 's/^([0-9]{2})/(\1) /'
(93) 5550055

$ echo "935550055" | sed -r 's/^([0-9]{2})([0-9]{3})([0-9]{2})([0-9]{2})/(\1) \2 \3 \4 /'
(93) 555 00 55
```

```
#10
$ echo "pere:15" | sed -r 's/^(.*):.*$/\1/'
pere

$ echo "pere:15" | sed -r 's/^(.*):/--\1--/'
--pere--15

$ echo "pere:15" | sed -r 's/^(.*):(.*)$/\2:\1/'
15:pere
```

```
$ grep "^root" /etc/passwd | cut -d: -f1,3
root:0

$ grep "^root" /etc/passwd | cut -d: -f1,3 | sed -r 's/^(.*):(.*)$/\2:\1/'
0:root

$ grep "^root" /etc/passwd | cut -d: -f1,3 | sed -r 's/^(.*):(.*)$/\1(\2)/'
root(0)
```

Paste and join

paste

```
#1
$ cat bdades.txt
anna:12:barcelona
pere:5:girona
pau:33:girona
julia:22:barcelona
marta:22:girona

$ cut -f1 -d:  bdades.txt > name.txt
$ cut -f3 -d:  bdades.txt > city.txt
$ cut -f2 -d:  bdades.txt > age.txt
```

```
#2
$ paste name.txt edat.txt
anna    12
pere    5
pau     33
julia   22
marta   22

$ paste name.txt city.txt
anna    barcelona
pere    girona
pau     girona
julia   barcelona
marta   girona

$ paste city.txt age.txt -d:
barcelona:12
girona:5
girona:33
barcelona:22
girona:22

$ paste city.txt name.txt age.txt  -d' '
barcelona anna 12
girona pere 5
girona pau 33
barcelona julia 22
girona marta 22
```

join

```
#1
$ paste name.txt age.txt > join1
$ paste name.txt city.txt > join2

$ join join1 join2
anna 12 barcelona
pere 5 girona
pau 33 girona
julia 22 barcelona
marta 22 girona

# sort join1 > join1s
# sort join2 > join2s
```

```
#2
$ cat join1s
anna    12
doe     17
julia   22
marta   35
pau     33
pere    5
unknown 35

$ cat join2s
anna    barcelona
jesus   lleida
julia   barcelona
lluis   lleida
marta   girona
pau     girona
pere    girona


# join
$ join join1s join2s
anna 12 barcelona
julia 22 barcelona
marta 35 girona
pau 33 girona
pere 5 girona

# left join
$ join -a1 join1s join2s
anna 12 barcelona
doe 17
julia 22 barcelona
marta 35 girona
pau 33 girona
pere 5 girona
unknown 35

# right join
$ join -a2 join1s join2s
anna 12 barcelona
jesus lleida
julia 22 barcelona
lluis lleida
marta 35 girona
pau 33 girona
pere 5 girona

# full outer join
$ join -a1 -a2 join1s join2s
anna 12 barcelona
doe 17
jesus lleida
julia 22 barcelona
lluis lleida
marta 35 girona
```

```
pau 33 girona
pere 5 girona
unknown 35
```

```
#3
$ cat status
menor    12
menor    17
major    22
major    35
major    33
menor     5
major    35

$ join  -j1 2  -j2 2 join1s status
12 anna menor
17 doe menor
22 julia major
35 marta major
33 pau major
5 pere menor
35 unknown major
```

Exercicis d'exemple

1. Github LPIC-1 103.7-Exercices.md
2. Github LPIC-1 103.7-Exercices_alternate_2.pdf
3. Github LPIC-1 103.2-Exercices.md
4. LPI Exercices 103.7 Search text files using regular expressions

Exercices

[Grep]

1. Search all the lines in the file /etc/passwd containing the string root.
2. Search in the first 12 lines of the file /etc/passwd the lines starting with the word root.
3. Search in the last 16 lines of the file /etc/passwd the lines ending with nologin.
4. Search in the file /etc/group for a group with the GID 42.
5. Search in the file /etc/passwd all the groups with GID from 20 to 29.
6. Using the echo and grep commands, validate if a DNI has an appropiate value (echo "12345688A" | grep ...).
7. Using the echo and grep commands, validate if a date has the format dd-mm-aa.
8. Filter the file /etc/services showing all the lines containing http or HTTP.
9. Filter the lines of the file /etc/fstab showing al the lines without the char #.

[tr]

10. Show the file /etc/group using one tabulation as a delimiter.
11. Show the lines of the file /etc/passwd ending with /bin/bash in upper case.
12. Show the last 3 lines of the file /etc/group transforming the vowels in numbers.
13. Filter the output of the command uname -a deleting all the numbers.
14. Filter the first 5 lines of the file /etc/passwd deleting all the vowels.
15. Squeeze all the spaces in the output of the ls -l / command.
16. Show the last 5 lines of the file /etc/fstab in a normalized format (using only one space or one tabulation as a delimiter).

[cut]

17. List (long list) the root directory and show only the characters from 2 to 10.
18. Show the login, uid and gid of the users accounts.
19. List the fields from the gid to the end, of all users using the /bin/bash shell.
20. List the gname, gid and user-list of the last 15 groups.
21. (*difficult*) Long list the root directory and show the fields: type/permissions, owner and name. It is necessary to normalize the blanks before cutting the fields.

[sort / uniq]

22. List by gname order all the system groups.
23. List the system groups sorted by gid in descendant order.
24. List the users using gid as a primary order key and uid as a secondary.
25. List the users accounts grouped by gid and sorted by login name in descendant order.
26. List all the different gids used in the /etc/passwd.
27. List all the different shells used by users.

[sed]

28. Change the /etc/passwd delimiter to --.
29. Delete all lines in the /etc/passwd containing the word root.
30. Change all occurrences of the string bin for the word BIN in the /etc/passwd file.
31. Do the command echo "935550055" and transform the output to (93) 555 00 55.
32. Cut the login and uid of the first 10 users in the password accounts file and show it in the format login(uid)

[paste]

33. Create the two next files and paste the contents to a new file.

```
cat <<EOF > file1.txt
IBM
```

```
INTC
SAP
VMW
EOF
```

```
cat <<EOF > file2.txt
174.99
22.99
59.37
102.92
EOF
```

34. Paste the two previous files using the -s option.

35. Create a file named fit1.txt containing the login, passwd and uid of the then first system users. Create a file named fit2 containing he gid, home and shell of the first then system users. Show the two files pasted.

[join]

36. Create the files noms.txt and cognoms.txt and join them. Observe the difference.

```
1 pere
3 marta
4 borrell
5 pau
9 rosa
```

```
1 puig
2 vilada
3 mas
5 pou
9 roure
```

37. Repeat the previous exercise doing a left join. And then, again doing a full outer join.

38. Create the file cognoms2.txt and join noms.txt with cognoms2.txt using the id field as key.

```
puig pedroś 1
vilada vendrell 2
mas morell 3
pou prat 5
roure ribalta 9
```

39. Repeat the previous exercise doing a full outer join.
    join -a1 -a2 -j1 1 -j2 3 noms.txt cognoms2.txt