Introduzione all'interfaccia utente di Linux

E.Mumolo, mumolo@units.it

Generalità

- All’accesso nel sistema:
  - Operazioni predefinite dall’amministratore del sistema
  - Operazioni predefinite dall’utente
- All’accesso nel sistema:
  - Vengono assegnati gli identificatori dell’utente e del gruppo utenti alla Shell
  - Vengono assegnati gli identificatori PID, PPID del processo di Shell
- Sintassi di un comando Unix: `comando [opzioni o flags] [argomenti]`

- **SHELL**: programma di interfacciamento con l’utente e linguaggio di programmazione. Alcune Shell:
  - BOURNE (Steve Bourne, AT&T),
  - BASH (FSF, `bash`),
  - C (BSD, `csh`),
  - KORN (Dave Korn, AT&T, `ksh`),
  - TC (`csh`) ,
  - Z (`zsh`),
  - TCSH (`tcsh`)

Generalità

- Primi comandi Unix: man <comando> --> per avere informazioni sul <comando>
- Esempio: comando uname

```
$ man uname
DESCRIPTION
       Print certain system information.  With no OPTION, same as -s.
       -a, --all                print  all  information
       -s, --kernel-name        print the kernel name
       -n, --nodename           print the network node hostname
       -r, --kernel-release     print the kernel release
       -v, --kernel-version     print the kernel version
       -m, --machine            print the machine hardware name
       -p, --processor          print the processor type or "unknown"
       -i, --hardware-platform  print the hardware platform or "unknown"
       -o, --operating-system   print the operating system
       --help                   display this help and exit
       --version                output version information and exit
```

Generalità

- `id` → visualizza il real UID e il real GID

```
$ id
uid=21132(i3101464) gid=10000(studenti)
```

- `id –u` → visualizza l'effective used ID

```
$ id -u
21132
```

- `id –g` → visualizza l'effective group ID

```
$ id -g
10000
```

- `ps` → mostra i processi che stanno eseguendo per un utente

```
$ ps -f
UID         PID   PPID    C STIME    TTY              TIME CMD
i3101464   2328   2327  0.0 16:07:15 ttyp6        0:00.42 -ksh (ksh)
```

Generalità

- `who` → mostra gli utenti attualmente collegati

```
ingstud.univ.trieste.it> who
i3100602    ttyp0       Mar  25 15:10
i3100602    ttyp1       Mar  25 14:10
...
ingstud.univ.trieste.it>
```

- `Passwd` → modifica la password
  - file /etc/passwd: nome_utente::user-id:group-id:commento:home_directory:shell
- `logout, ^d, exit, shutdown (solo superuser)` → Sconnessione

- Formato dei comandi:
  - comandi minuscolo
  - variabili d'ambiente maiuscolo

- Standard input, standard output, standard error

- Redirezione (>, >>, <, <<)

Generalità

- Processi in foreground, in background, operatore &

- Segnali da tastiera:
  `erase (^h - cancella un carattere), kill (^u – cancella la riga), stop (^s), start(^q), susp (^z/^y), intr (^c), quit (^\ - core dump), eof (^d)`

- Qualche altro comando utile:
- `date` → visualizza la data

$ date
Wed Mar 25 18:20:49 MET 2003
$

- `Uptime` → tempo di vita di un sistema, nr. utenti collegati, carico del sistema negli ultimi 1, 5, 15'

$ uptime
18:21 up 4:21, 5 users, load average: 0.00, 0.06, 0.05
$

- `hostname` → nome dell'host

$ hostname
ubuntu

Altri comandi utili

- `users` → visualizza gli utenti collegati attualmente

```
ingstud.univ.trieste.it> u
i3101251 i3101251 i3101464
```

- `w` → visualizza cosa stanno facendo gli utenti.

Qualche opzione:

-h sopprime l'intestazione

-l formato lungo d'uscita (default)

```
$ w
18:57  up  4:56,  3 users,  load average: 0.24, 0.10, 0.07
User     tty        from             login@    idle   JCPU   PCPU what
i3101251 p1         192.168.172.140  18:46        1                telnet
i3101251 p4         192.168.172.56   18:42        2                telnet
i3101464 p6         caesar.univ.trie 16:07              11        w
```

- `finger <username>` → informazioni sugli utenti locali e remoti leggendo il file password

```
$ finger viola
Login       Name                TTY         Idle    When    Where
viola    Massimiliano VIOLA     pts/1        <Feb 12 13:29> presing01.univ.ts.it
$ finger viola@ingstud
Login name: viola                           In real life: Massimiliano VIOLA
Directory: /home/viola                      Shell: /bin/ksh
Never logged in. No Plan.
```

Altri comandi utili

- `ping <dominio>` → controlla se un host è vivo

```
$ /usr/sbin/ping uts
PING uts.univ.trieste.it (140.105.48.1): 56 data bytes
64 bytes from 140.105.48.1: icmp_seq=0 ttl=62 time=1 ms
64 bytes from 140.105.48.1: icmp_seq=1 ttl=62 time=0 ms

----uts.univ.trieste.it PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/1 ms
```

- `whereis <comando>` → cerca il file ‘comando’ nelle directories

/etc, /etc/nls, /sbin, /usr/bin, /usr/lbin, /usr/lbin/spell,
/usr/ccs/lib, /usr/lib, /usr/local, /usr/hosts, /usr/sbin

- `pwd` → visualizza il nome della corrente directory di lavoro

Uno sguardo alla struttura logica del file system

/	→	radice
/bin	→	file eseguibili del sistema – ls, pwd, cp, mv ....
/boot	→	file necessari per l'avvio del sistema, boot loader ...
/dev	→	file speciali che descrivono i dispositivi – dischi, scheda audio, porte seriali ...
/etc	→	file eseguibili, script, inizializzazione, configurazione sistema, file password, ...
/home	→	directory delle home directory degli utenti
/lib	→	librerie di sistema
/lost+found	→	contiene i file danneggiati
/mnt	→	punto di montaggio file system (mount point)
/proc	→	file system virtuale che contiene informazioni sui programmi in esecuzione
/sys	→	programmi di sistema
/tmp	→	direttorio temporaneo
/usr	→	file relativi alle applicazioni installate
/usr/include	→	header file libreria standard C
/usr/bin	→	file binari disponibili agli utenti
/var	→	file di sistema che variano con frequenza elevata
/var/spool	→	aree temporanee di spooling

Comandi per il file system

- NOTA: TUTTI GLI OGGETTI IN UNIX SONO FILE (comandi, archivi, programmi utente, programmi applicativi, dispositivi..)

- `rm [-i] [filename]` → rimuove il/i file ordinari selezionati

QUALCHE OPZIONE

-r rimozione ricorsiva del contenuto delle directories. I link simbolici incontrati non vengono considerati.

Una rimozione di una directory non vuota e protetta in scrittura fallisce sempre!!

-f rimozione di tutti i file (anche protetti in scrittura) senza avvisare. Se la directory e' protetta in scrittura i file non vengono mai rimossi ma non viene mostrato nessun avviso.

-i con questa opzione rm chiede conferma

ESEMPIO

rm –r prova cancella tutto dalla directory prova in giu'

rm –r prova/* cancella tutti i file e directory da prova in giu' ma non prova

- `rmdir [directory]` → rimuove una directory vuota

Comandi per il file system

- `ls [-1ltsaRn] [filename]` → lista il contenuto della directory

QUALCHE OPZIONE:

-1 stampa su una colonna

-l formato lungo

-n come –l ma visualizza gli ID al posto del nome del proprietario e del gruppo

-t ordina per data

-s mostra la dimensione dei file in blocchi

-a mostra tutti i file compresi . e ..

-R elenca il contenuto in modo ricorsivo

-n mostra UID e GID

-i mostra il nr. di i-node

NB: il primo carattere puo' essere: d (dir), l(sym.link), b(block) ,c(char), p(pipe),s(socket), -

Es.:

$ls -1

10

11

1q

2

…

Comandi per il file system (Sistema Operativo Solaris)

```
$ ls -l
total 17354
-rw-r--r-- 1 mumolo calcolat 12249 May 8 1997 10
-rw-r--r-- 1 mumolo calcolat 7781 May 8 1997 11
-rw-r--r-- 1 mumolo calcolat 775 Jan 23 11:38 1q
-rw-r--r-- 1 mumolo calcolat 1 Nov 16 1995 2
-rw-r--r-- 1 mumolo calcolat 8689 Nov 25 1999 20db.eps
-rw-r--r-- 1 mumolo calcolat 21175 Apr 13 1996 3
-rw-r--r-- 1 mumolo calcolat 1545 Jun 5 1998 38
-rw-r--r-- 1 mumolo calcolat 7780 Feb 15 1999 40n_1.ps
-rw-r--r-- 1 mumolo calcolat 7787 Feb 15 1999 40n_2.ps
-rw-r--r-- 1 mumolo calcolat 7798 Feb 15 1999 80n4b_1.ps
-rw-r--r-- 1 mumolo calcolat 7795 Feb 15 1999 80n4b_2.ps
-rw-r--r-- 1 mumolo calcolat 11146 Apr 3 1998 Array.c
drwxr-xr-x 2 mumolo calcolat 1024 Nov 26 1995 source
lrwxrwxrwx 1 mumolo calcolat 3 Jan 27 2000 ss -> pse
srwxrwxrwx 1 mumolo calcolat 0 Jan 25 17:05 nomesocket
-rw-r--r-- 1 mumolo calcolat 232 Feb 2 1999 np.mat
```

Comandi per il file system

```
$ ls -n
total 17354
-rw-r--r-- 1 3281 15006 12249 May 8 1997 10
-rw-r--r-- 1 3281 15006 7781 May 8 1997 11
-rw-r--r-- 1 3281 15006 775 Jan 23 11:38 1q
-rw-r--r-- 1 3281 15006 1 Nov 16 1995 2
-rw-r--r-- 1 3281 15006 8689 Nov 25 1999 20db.eps
-rw-r--r-- 1 3281 15006 21175 Apr 13 1996 3
-rw-r--r-- 1 3281 15006 1545 Jun 5 1998 38
-rw-r--r-- 1 3281 15006 7780 Feb 15 1999 40n_1.ps
…
$ ls –il
total 17354
49828 -rw-r--r-- 1 mumolo calcolat 12249 May 8 1997 10
49815 -rw-r--r-- 1 mumolo calcolat 7781 May 8 1997 11
49695 -rw-r--r-- 1 mumolo calcolat 775 Jan 23 11:38 1q
49877 -rw-r--r-- 1 mumolo calcolat 1 Nov 16 1995 2
50220 -rw-r--r-- 1 mumolo calcolat 8689 Nov 25 1999 20db.eps
50017 -rw-r--r-- 1 mumolo calcolat 21175 Apr 13 1996 3
50145 -rw-r--r-- 1 mumolo calcolat 1545 Jun 5 1998 38
```

Comandi per il file system

- `cd <dir>` → cambia directory. Punto (.) directory corrente. Doppio punto (..)

  radice della directory corrente

NOTA

Il carattere ~ significa home directory

Il permesso di scrittura in un directory significa ricerca!

$ ls -l prova

total 4

drwxr-xr-x 2 mumolo calcolat 512 Mar 5 14:09 sub1

drwxr-xr-x 2 mumolo calcolat 512 Mar 5 14:09 sub2

$ chmod 600 prova

$ ls prova

sub1 sub2

$ ls -l prova

prova/sub1: Permission denied

prova/sub2: Permission denied

total 0

- `mkdir (crea sub-directory)`

Comandi per il file system

- `cp <file1> <file2>` → copia file

QUALCHE OPZIONE

-f Unlink. Se il file di destinazione non puo' essere sovrascritto, lo rimuove e continua

-i Interattivo. cp chiede conferma nel caso che la copia sovrascriva il file di destinazione

-r Ricorsivo. cp copia la directory e tutti I suoi file, incluso le sottodirectory ed i loro file alla destinazione

-R uguale a -r, eccetto che le pipe sono duplicate senza leggerle

Es. copia di una gerarchia di file:

$cp –r dirA/* dirB → tutta la gerarchia dei file da dirA in giu' si ritrova da dirB in giu'

$cp –r dirA dirB → sotto dirB si ritrova dirA con tutti i file e le directories

- `mv <file1> <file2>` → muove file

QUALCHE OPZIONE

-f mv muove i files senza chiedere conferma anche se sovrascrive una destinazione esistente

-i mv chiede conferma nel caso che il movimento sovrascriva la destinazione.

Comandi per il file system

- `ln <file> <nuovo file>` → Hard-Link. Aggiunge una coppia nome file – nr. inode senza aggiungere un file: lo stesso file avra' piu' nomi. La cancellazione decrementa il nr. di link.

ESEMPIO:

```
$ ls -il miofile.txt
49671 -rw-r--r-- 1 mumolo calcolat 12 Mar 5 16:26 miofile.txt
$ ln miofile.txt mio
$ ls -il mio
49671 -rw-r--r-- 2 mumolo calcolat 12 Mar 5 16:26 mio
$ ls -il miofile.txt
49671 -rw-r--r-- 2 mumolo calcolat 12 Mar 5 16:26 miofile.txt
$ rm -i miofile.txt
rm: remove miofile.txt (yes/no)? y
$ ls -il miofile.txt
miofile.txt: No such file or directory
$ ls -il mio
49671 -rw-r--r-- 1 mumolo calcolat 12 Mar 5 16:26 mio
```

NOTA: il file linkato appare come file regolare!

Comandi per il file system

- `ln –s <file> <nuovo file>` → Symbolic Link.

ESEMPIO:

```
$ ln -s mio altro
$ ln -s mio suo
$ ls -il mio
49671 -rw-r--r-- 1 mumolo calcolat 12 Mar 5 16:26 mio
$ ls -il altro
49760 lrwxrwxrwx 1 mumolo calcolat 3 Mar 5 16:44 altro -> mio
$ ls -il suo
49762 lrwxrwxrwx 1 mumolo calcolat 3 Mar 5 16:44 suo -> mio
$
```

NOTA: i file linkati appaiono come link

- `chmod <perm> file` → cambia i permessi:
  - chmod nmt <nomefile> dove n, m e t sono i permessi per proprietario, gruppo e tutti gli altri in ottale.
  - chmod classe [+/-] permessi <nome file> dove classe puo' essere: u (proprietario), g (gruppo), a (altri)
  - chmod ug+x file

Comandi per il file system

Esempi di chmod:

```
$ ls -l mio
-rw-r--r-- 1 mumolo calcolat 12 Mar 5 16:26 mio
$ chmod 666 mio
$ ls -l mio
-rw-rw-rw- 1 mumolo calcolat 12 Mar 5 16:26 mio
$ chmod uga-w+r-x mio
$ ls -l mio
-r--r--r-- 1 mumolo calcolat 12 Mar 5 16:26 mio
```

- `find [path] [-n nome] [-print]` → ricerca ricorsiva di directories

Esempio piu' semplice (mostra tutti i file utente):

```
$ find . -print
./.wastebasket
./sub
./sub/dati
./sub/dato
./pl
./NewFolder
./NewFolder/bar1.c
```

./NewFolder/customer1.cio

…

Comandi per il file system

Altro esempio:

```
$ find . -name "*.c"
/net/cli-ser/gc.c
./net/cli-ser/ech.c
./net/cli-ser/gech.c
./net/echo-cli/echoclient.c
./net/finger/finger.c
./net/finger/lprint.c
find: cannot read dir ./qq/: Permission denied
./phase/PVC-3.0-linux/PVC-3.0-linux/CMUSIC_GEN/gen/cspline.c
./phase/PVC-3.0-linux/PVC-3.0-linux/CMUSIC_GEN/gen/gen0.c
```

…

NOTA: find METTE A DISPOSIZIONE ALTRI PREDICATI:

-perm (selezione sui permessi), -type (selezione sul tipo di file), -user (selezione sull'utente), -group (selezione sul gruppo), -size (selezione sulla dimensione: esatta con =, inferiore con <, superiore con >

ESEMPIO:

find . –mtime 0 → file modificati 0 giorni fa

find . –mtime 1 → file modificati ieri

find . –mtime –2 → file modificati oggi e ieri

Comandi per il file system

- `tail [+numero [lbc]] [file]` →stampa le ultime numero oggetti dal fondo del file)

-b unita' in blocchi.

-c unita' in byte

-l unita' in linee

-r copia le linee dal punto specificato in ordine inverso.

- `touch [file]` → aggiorna la data di ultima modifica del file alla data corrente

ESEMPIO:

```
$ ls -l a*.c
-rw-r--r-- 1 mumolo calcolat 11146 May 12 1998 array.c
-rw-rw-rw- 1 mumolo calcolat 400 Jan 14 1997 assign.c
ingsun2/home/mumolo $ ls -l b*.c
-rw-r--r-- 1 mumolo calcolat 905 Jan 14 1997 back.c
-rw-rw-rw- 1 mumolo calcolat 1033 Jan 14 1997 boh.c
-rw-rw-rw- 1 mumolo calcolat 796 Jan 14 1997 builtin.c
$ touch b*.c
$ ls -l b*.c
-rw-r--r-- 1 mumolo calcolat 905 Mar 5 14:24 back.c
-rw-rw-rw- 1 mumolo calcolat 1033 Mar 5 14:24 boh.c
-rw-rw-rw- 1 mumolo calcolat 796 Mar 5 14:24 builtin.c
$
```

Comandi per il file system

- `nice` →abbassa la priorita'. Es. nice –8 processo

- `at [time] [comandi] [processo]` (attiva il processo all'ora specificata)

es. at 3pm <comandi

- `env` → stampa le variabili d'ambiente. Es.

$ env
LANG=C
PATH=/usr/bin:/usr/ucb:/etc:.
LOGNAME=mumolo
MAIL=/var/mail/mumolo
PS1=ingsun1$
SHELL=/bin/ksh
HOME=/home/mumolo
FCEDIT=vi
TERM=vt100
PWD=/home/mumolo

Comandi per il file system

- `ps [opzioni]` → mostra informazioni sui processi attivi

QUALCHE OPZIONE:

-a informazioni su tutti i processi piu' utilizzati
-d informazioni su tutti i processi tranne i processi leader
-e info su tutti I processi che girano correntemente
-g grplist mostra solo i processi che appartengono ai gruppi listati (in termini di ID dei leader)
-G gidlist mostra solo i processi con real-group-ID elencati nella lista separata da virgola o spazio
-f formato completo
-l mostra una lista con formato lungo
-L mostra informazioni sui thread attivi nei processi selezionati
-o format mostra informazioni secondo un formato specificato
-p proclist mostra solo i processi con PID elencati
-t term mostra solo i processi associati con il terminale indicato
-u uidlist mostra solo i processi con l'effective-user-ID elencato
-U uidlist mostra solo i processi con i real-user-ID elencati

Comandi per il file system

FORMATI D'USCITA
S (l) lo stato del processo:
O running
S il processo e' in attesa di un evento
R pronto in memoria
Z stato di Zombie: il processo e' terminato ma il padre non lo sta
aspettando
T stoppato
UID (f,l) effective-user-ID del processo
PID (all) il process-ID
PPID (f,l) il process-ID del padre
PRI (l) la priorita' del processo
SZ (l) dimensione totale (virtuale) del processo
STIME (f) l'istante di partenza del processo, in ore, minuti, secondi
TTY (all) il terminale che controlla il processo. Se non c'e' nessun terminale Å ?
TIME (all) il tempo cumulative di esecuzione
Se e' specificato –j:
PGID il PID del leader del gruppo al quale appartiene il processo
Se e' stato specificato –L:
LWP l'ID del thread
NLWP il numero di thread del processo

Comandi per il file system

Esempi di ps

```
$ ps
PID TTY TIME CMD
15298 pts/0 0:01 ksh
$ ps -alf
F S      UID   PID  PPID C PRI NI     ADDR  SZ WCHAN       STIME   TTY TIME CMD
8 O   mumolo 15386 15298 1  61 20 f60c86e0 203          09:27:18 pts/0 0:00 ps -alf
8 S pediroda 13411  3706 0  41 20 f6189030 982 f593b4e6   Mar 03 pts/2 0:03 pine
8 S  toffoli 15380 15369 0  51 20 f6235db8 886 f616a08e 09:25:42 pts/3 0:00 pine
```

- `kill -s PID` → manda il segnale s (numerico o mnemonico) al processo PID, che deve appartenere all'utente (oppure root)

ESEMPIO:

`kill -9 100 –165` → manda il segnale 9 al processo con PID=100 e a tutti i processi del gruppo 165

`kill -s kill 100 -165` → stessa cosa, segnale mnemonico

`kill -s KILL 100 -165`

Dispositivi

- NOTA:

  FILE SPECIALI IN /dev/nomedevice Esempio ls –l /dev/tty0

  DISCHI IN /dev/dsk

- `df [-tv]`: visualizza il nome del file system, nome dispositivo, nr. Blocchi liberi, i-node disponibili. Blocchi di 512 byte.

-t : nr totale di blocchi e i-node liberi

-v: percentuale di blocchi e i-node

- `du [-arsu] [directory]` (visualizza per ogni subdirectory, il numero di blocchi usati)

-a: nr blocchi del file

-r: visualizza messaggio se le directory o i file non possono essere letti

-s: nr totale di blocchi relativi a tutte le directory

-u: ignora i file che hanno + di un link

- `file <nomefile>` (indica il tipo di file)

ESEMPIO:

```
$ file prova
prova: directory
$ file a.out
a.out: ELF 32-bit MSB executable SPARC Version 1, dynamically
   linked, not stripped
```

Visualizzazione dei file

- `head [-count] [file]` count e' il nr. di linee da visualizzare (default 10)
- `tail [-count] [file]` (vedi sopra)
- `more [-cs] [+startline] [+/pattern] [file]` Visualizza il file una schermata alla volta fornendo il prompt dopo ogni schermata.

  Per avanzare di schermata -> spazio.

  Startline e' il numero di linea da cui si vuole iniziare. Pattern e' il pattern iniziale da cercare.

  Al prompt si possono dare alcuni comandi, per es. -> h

  L'opzione –c visualizza ogni riga sovrascrivendo dall'alto verso il basso lo schermo.

  L'opzione –s sostituisce piu' linee bianche con una sola linea bianca.

- `pg [-cn] [+startline] [+/pattern] [file]` Simile a more, solo che si preme return, non spazio.

  L'opzione –n dice di eseguire i comandi di una sola lettera senza premere return. Opzione –c come more.

- `Look [-df] pattern [file]` Cerca nel file che deve essere ordinato e seleziona le righe che iniziano con pattern.

  Con -d si considerano solole lettere maiuscole, minuscole, numeri, tabulatori e spazi.

  -f tratta maiuscole come minuscole.

Filtri in Unix

DEF: UN FILTRO E' UN PROGRAMMA CHE LEGGE DALLO STANDARD INPUT E SCRIVE UN QUALCHE RISULTATO SULLO STANDARD OUTPUT.

IL MODO PIU' OVVIO DI COMUNICAZIONE TRA FILTRI E' MEDIANTE UNA PIPELINE

- `cat` (concatenate) Non fa' nulla: copia dallo standard input allo standard output.

Legge dalla tastiera fino ad EOF (^d). Usi di cat:

1) per creare un file: cat > file

2) per aggiungere dati ad un file esistente: cat >>file

3) per visualizzare un file: cat <file

4) per copiare due file: cat < file1 > file2

Estensioni: concatenazione di file. cat file1 file2 file3 > file4

Opzioni:

cat [-bns] file

-n numera le righe

-nb non numera le righe bianche

-s sostituisce piu' linee bianche con una linea

Filtri in Unix

- `cut` → Estrae colonne o campi di dati dal file.

Possibili sintassi:

cut –blista [file] o cut –bn1-n2 [file] dove –b specifica la posizione in byte all'interno di ogni riga

cut –clista [file] dove –c specifica la posizione in caratteri

cut –flista [–d<carattere>] [file] dove –f specifica la posizione in campi e –d il delimitatore

QUALCHE OPZIONE

list lista separata da virgole o spazi. Esempio: 1,4,7 oppure 3-

-b list esempio –b1-72 sono I primi 72 byte

-c list esempio: -c1-72 individua I primi 72 caratteri

-d delim il carattere che segue –d e' il delimitatore di campo (solo con opzione –f) default e' Tab

ESEMPI:

```
$ cat > a.txt
Questo e' un file di testo
preso come esempio per
realizzare alcune funzioni
di elaborazioni di testo
in Unix.
```

Filtri in Unix

```
$ cut -c1-5 a.txt
Quest
preso
reali
di el
in Un
$ cut -d' ' -f1-2 a.txt
Questo e'
preso come
realizzare alcune
di elaborazioni
in Unix.
$ cut -d' ' -f3-4 a.txt
un file
esempio per
funzioni
di testo
```

Filtri in Unix

- `paste [-d char]` file combina colonne di dati.

Cioe', con paste file1 file2 file 3 > out si concatenano i file per riga.

L'opzione –dchar serve per mettere il char tra i campi

ESEMPIO:

```
$ cut -c1-5 a.txt > a1.txt
$ cut -c6-7 a.txt > a2.txt
$ cut -c8- a.txt > a3.txt
$ cat a1.txt
Quest
preso
reali
di el
in Un
$ cat a2.txt
o
c
zz
ab
ix
```

Filtri in Unix

```
$ cat a3.txt
e’ un file di testo
ome esempio per
are alcune funzioni
orazioni di testo

$ paste a1.txt a2.txt a3.txt
Quest o e’ un file di testo
preso c ome esempio per
reali zz are alcune funzioni
di el ab orazioni di testo
in Un ix .

$ paste -d’ ’ a1.txt a2.txt a3.txt
Quest o e’ un file di testo
preso c ome esempio per
reali zz are alcune funzioni
di el ab orazioni di testo
in Un ix .
```

Filtri in Unix

- `crypt [key]` codifica dati mediante una chiave. Uso: crypt > file e crypt < file

Esempio

```
$ crypt < a.txt > acrypt.txt
Enter key:
$ cat acrypt.txt
fÅ¾Ã"µrÄ„` ‡ú|ÎÈ5×´ ^ÁÚ¼§!¸ZØÎV
ÌÈ
u,ê…q»617„Fæ[ïøúY¾v›¬“½aa•¾-ÿJLK
¨òE¤•ƒ L ;áœÊî:9éØÇ'š ‡Qkú‡ü?0z]
$
$ crypt < acrypt.txt
Enter key:
Questo e' un file di testo
preso come esempio per
realizzare alcune funzioni
di elaborazioni di testo
in Unix.
$
```

Altri comandi importanti

- `sort [-dfnru] [-o outfile] [file…]`

Ordina i dati del file.

–d considera solo lettere, numeri e spazi.

–f tratta maiuscole come minuscole.

–n riconosce i numeri e li ordina in modo numerico.

–r ordina i dati in modo inverso.

–u cerca le linee uguali e ne lascia una sola.

sort –m [-o outfile] sortedfile.. Legge file gia' ordinati e li fonde.

- `spell [-b] [file]` Legge dei dati e genera una lista della parole scritte in modo sbagliato. –b seleziona americano o britannico

- `tr [-cds] [set1] [set2]` Legge dei dati e sostituisce i caratteri specificati con altri caratteri. Esempio `tr a A < file1 > file2`

Se il secondo set e' piu' corto del primo, l'ultimo carattere e' ripetuto.

-c e' il complemento del primo insieme

-d cancella tutti i caratteri specificati

-s sostituisce le ripetizioni del carattere specificato con un solo carattere

- `uniq [-cdu] [infile] [outfile]` Esamina i dati linea per linea cercando linee duplicate e puo': tenere solo le linee duplicate (-d), tenere solo le linee uniche (-u), eliminare le linee duplicate e contare quante volte le linee sono duplicate (-c).

- `wc [-lwc] [file]` Conta linee (l), parole(w) e caratteri(c) dello standard input o del file