

Linux

marco.falda@unipd.it

Origini

- Ideato da Linus Torvalds nel 1991
 - a partire dal S.O. Minix
 - si appoggia allo *standard* POSIX

- Appartiene alla famiglia UNIX
 - multi-utente, *multitasking*
 - dispositivi, non unità
 - comandi diversi

Funzionalità

- Funzioni principali
 - gestione di processi e memoria
 - gestione delle periferiche
 - gestione dei *file*

 - Linux è un S.O. che fa largo uso dell'interfaccia a caratteri
 - la sola *standard*
-

Window Manager

- L'interfaccia grafica di Linux non è *standard*

 - È gestita da Window Manager
 - K Desktop Environment (KDE)
 - GNOME
 - ...
-

KDE

Le console in Linux

- Le *console* permettono di interagire tramite comandi

 - *Un comando è formato da*
 - *un nome*
 - *opzioni*
 - *argomenti*
-

Gestione dei processi

- Elenco dei processi
 - ps: elenca i processi ed esce
 - top: mostra i processi più attivi (C-c per terminare)
- Ogni processo ha un num. (PID)
- Terminazione dei processi
 - kill *n* (termina il processo *n*)
 - kill -9 *n* (per forzare)

Esercizio

1. Avviare una *console*
2. avviare emacs dalla *console*
 - scrivendo "emacs &"
3. elencare i processi
4. terminare emacs

Gestione delle periferiche

- Non esistono unità, solo dispositivi
- I dispositivi devono essere caricati (*mount*) nel *filesystem*
 - `mount origine destinazione`
 - `df` (-h per dimensioni in *byte*)
- Nelle distribuzioni più moderne c'è il caricamento automatico
 - CD, USB card

I dispositivi più comuni

/dev/fd0	unità a dischetti
/dev/hda1	1 ^a partizione
/dev/hda2	2 ^a partizione
/dev/cdrom	unità ottica
/dev/sdb1	2 ^a HD SCSI (USB)
/dev/lp0	porta parallela
/dev/ttyS0	porta seriale

Filesystem

- Tipi diversi (in Windows NTFS o FAT)
 - ext2, ext3, ReiserFS, ...
- Alcune differenze
 - maiuscole/minuscole
 - senza estensioni (non rilevanti)
 - "/" invece di "\" nei percorsi

I permessi

- *File e directory* hanno permessi associati
 - lettura, scrittura, esecuzione

```
chmod [ $\pm$ r $\pm$ w| $\pm$ x] nomefile
```
- Permesso di esecuzione
 - eseguire i programmi
 - esplorare le *directory*

Elencare file e directory

- Il comando `ls` permette di elencare *file* e *directory*

 - Opzioni
 - `-l`: mostra i permessi
 - `-a`: mostra *file* nascosti
 - `-R`: entra nelle *subdirectory*

 - Il comando `pwd` stampa la *directory* di lavoro
-

Gestire le directory

- Creazione: `mkdir`
 - `mkdir ~/prova`

 - Spostarsi tra le *directory*: `cd`
 - `cd ~/prova`
 - `cd ..` (livello precedente)

 - Eliminazione: `rmdir`
 - `rmdir ~/prova`
-

Esercizio

1. Elencare i *file* della propria *home directory* (indicata con ~)
2. creare una directory
3. entrarci
4. tornare al livello precedente
5. eliminarla

Gestire i file

- Copia: *cp origine destinazione*
 - `cp ~/prova.doc ..`
- Spostamento / ridenominazione: *mv vecchionome nuovonome*
 - `mv ~/prova.doc prova1.doc`
- Eliminazione: *rm origine dest*
 - `rm ../prova.doc`

Altri comandi utili

- `man`: mostra la guida per un determinato comando
 - `less`: visualizza il contenuto di un *file* pagina per pagina
 - `lpr`: stampa un *file*
 - `lpr stampante nomefile`
-

I collegamenti

- In Linux esistono collegamenti veri e propri
 - fisici: creano copie indistinguibili (meglio evitare)
 - simbolici: conservano la differenza con l'originale
 - Si usa il comando `ln`
 - Per i collegamenti simbolici: `-s`
 - `ln -s origine destinazione`
-

Cercare in un file

- Per cercare in un *file* si può usare il comando `grep`
 - `grep [opzioni] pattern file`
- Le opzioni più utili sono
 - `-i`: ignora maiuscole/minuscole
 - `-w`: cerca parole intere
 - `-C`: stampa 2 righe di contesto
 - `-n`: stampa il numero di riga
 - `-H`: stampa il nome del *file*

Le espressioni regolari

- Anche in Linux esistono i caratteri *jolly*
- Grep invece usa le espressioni regolari
 - `*` e `?` ripetono l'espressione precedente zero / al più una volta
 - `.` indica un carattere qualsiasi
 - `[i-jz]` un intervallo di caratteri da i a j più z (se si vuole il "-" si mette in fondo)
 - `^` e `$` indicano l'inizio e la fine del testo da cercare

Esercizi

- Cercare nel file “prova”
 - le righe con parole che contengono “que”
 - le righe con parole che contengono “questo”
 - le righe con parole che contengono “questa” o “quella”

Pipe e redirection

- I comandi possono essere concatenati usando “|”
 - `ls -la | less`
- *Output* e *input* possono essere reindirizzati a / da *file*
 - `ls -la > elenco.txt`

Esercizi

- Elencare i *file* in “~” e ordinarli con `sort`
- Elencare i *file* in “/” una pagina per volta
- Salvare un elenco dettagliato dei *file* in ~ in “elenco.txt”

Cercare un file

- Per cercare un *file* si può usare il comando `grep` e il *piping*
 - `ls | grep tesi.*`
 - `ls -la | grep ^d.*`
- Il risultato può essere salvato
 - `ls -la -R / | grep ^l.* > ris.txt`

Gli editor di testi

- Per creare e modificare i testi vi sono programmi più o meno diffusi
 - vi, ed
 - emacs, nedit, mcedit
 - kate, kedit (KDE)
 - gedit (GNOME)
-

Emacs

Funzionalità

- Posizionamento
- Modifica
- Annullamento
- Gestione dei file
- Ricerca
- ...

Posizionamento

Pagina precedente / successiva	M-v / C-v
Riga precedente / successiva	C-p / C-n
Indietro / Avanti	C-b / C-f
Centra il testo sul cursore	C-l
Parola precedente / successiva	C-b / C-f
Inizio / fine riga	C-a / C-e
Frase precedente / successiva	M-a / M-e
Inizio /fine testo	M-< / M->

Modifica

- Cancellare
 - un carattere a sx: BACKSPACE
 - un carattere a dx: C-d
 - una parola a sx: M- BACKSPACE
 - una parola a dx: M-d
 - una riga: C-k
 - una frase: M-k

- Sostituire: M-% e poi y / n

Annullamento

- Annullare un comando: C-g

- Annullare le modifiche: C-x u

- Per ripetere *n* annullamenti:

C-u *n* C-x u

 - vale per tutti i comandi

Gestione dei file

- Ogni *file* è gestito da un *buffer*
 - aprire / creare un *file*, C-x C-f
 - salvare un *file*, C-x C-s
 - elenco dei *buffer*, C-x C-b
 - scorrere tra i *buffer*. C-x o
 - aprire *n* *buffer*. C-x n
 - chiudere il *buffer*, C-x k
 - uscire, C-x C-c

Ricerca incrementale

- La ricerca è incrementale
 - cerca mentre si digita
- Procedura:
 1. C-s (o C-r) per iniziare una ricerca in avanti (all'indietro)
 2. digitare il testo
 3. premere C-s (o C-r) per cercare la corrispondenza successiva (precedente)

Esercizio

1. Eseguire emacs
 2. aprire “prova” e creare un *file*
 3. modificare il nuovo *file*
 4. cercare “que” in “prova”
 5. salvarli
 6. chiuderli
 7. uscire da emacs
-