

Corso GNU/ Linux - 3

- Shell
 - Comandi base
 - Struttura filesystem
 - Permessi
- Netstudent <http://netstudent.polito.it>
 - by err err@netstudent.polito.it
 - novembre 2007
 - CC- by
<http://creativecommons.org/licenses/by/2.5/it/>

Shell - 1

- L'interfaccia grafica in GNU/ Linux e` considerata come un applicativo qualsiasi
- L'interazione di base con il kernel ed il sistema in generale avviene storicamente tramite comandi testuali
- Questo deriva dall'implementazione in linguaggio C del sistema
- Il termine 'shell' indica genericamente il programma testuale che permette di comunicare con il kernel ed il sistema

Shell - 2

- Esistono diverse implementazioni di shell, tra cui una basilare sempre fornita assieme al kernel
- Sh
- Ksh
- Bash
- Csh, tcsh
- In parte equivalente al command.com del DOS

Shell - 3

- Da shell e` possibile eseguire applicazioni per qualsiasi scopo
- Ogni shell dispone inoltre di comandi 'nativi', che permettono di interagire con il sistema (esempio 'cd' oppure 'echo')
- Vi sono poi altri comandi di base aggiuntivi
- I comandi di base sono nell'ordine delle centinaia e uguali in tutte le distro
- E` importante conoscere un minimo questi comandi perche` permettono di gestire totalmente il sistema

Comandi: man - 1

- Un comando molto utile e` 'man'
- Esso permette di consultare la documentazione presente nel sistema
- La documentazione e` divisa in varie sezioni
- Vi si accede tramite 'man TERMINE'
- La sezione 1 di solito e` relativa ai comandi di base, 2 librerie di base, 5 sui file di configurazione, e cosi` via
- 'man 1 man' apre le informazioni su man in sezione 1

Comandi: man - 2

- Oltre a 'man', per trovare l'argomento cercato e` a volte comodo il comando 'apropos TERMINE', che elenca i documenti correlati
- 'whatis TERMINE' cerca invece il termine nel nome stesso dei documenti
- Esiste anche il comando 'info TERMINE', simile con una formattazione migliore, disponibile di solito per i programmi GNU

Comandi: ls

- 'ls' permette di visualizzare i file e directory sul filesystem
- Lanciato senza argomenti presenta i file nella directory corrente
- Lanciato come 'ls NOMEPERCORSO' elenca i file siti nella directory indicata
- Sono disponibili innumerevoli opzioni, per presentare informazioni aggiuntive
- 'ls -lha' elenca tutti i file con indicazione della dimensione, se è una directory e i permessi

Comandi: pwd cd

- Il comando 'pwd' informa sull'attuale directory di lavoro
- Il comando 'cd' permette di cambiare directory attuale
- Eseguito da solo ritorna alla directory dell'utente (detta anche home)
- 'cd NOMEPERCORSO' ci porta nella directory indicata
- 'cd -' ritorna alla directory precedente
- 'cd ..' si sposta alla directory di livello superiore

Comandi: mv, cp

- Con il comando 'mv' e` possibile rinominare o spostare un file
- 'mv FILEORIGINE FILEDESTINAZIONE'
- Il file di destinazione e di origine possono essere indicati con il percorso
- Il comando 'cp' permette la copia di un file
- 'cp FILEORIGINE FILEDESTINAZIONE'
- In entrambi e` possibile utilizzare i caratteri jolly '*' e '?', che sostituiscono una stringa ed un carattere qualsiasi

Comandi: cp, rm, rmdir

- Il comando 'cp -r ' e` ricorsivo nel caso di directory, in quanto copia a destinazione tutti i file o sotto-directory presenti
- 'rm' permette di cancellare un file (non le directory pero`)
- 'rmdir' permette di cancellare una directory
- Il comando 'rm -r' permette di cancellare in modo ricorsivo tutti i file e sotto-directory

Comandi: mkdir, touch

- 'mkdir' permette la creazione di una directory
- 'mkdir -p' serve per creare in un colpo solo una gerarchia di directory
- 'touch NOMEFILE' permette creare un file vuoto
- Se il file esiste, non lo modifica e aggiorna la data di ultimo accesso
- ':> NOMEFILE' crea un file vuoto
- Se il file esiste, lo svuota del suo contenuto

Comandi: file, cat, more

- Il comando 'file NOMEFILE' ci indica il tipo di file con cui abbiamo a che fare
- Sostituisce l'informazione fornita dall'estensione in DOS e Windows
- 'cat' permette di concatenare i file ('cat FILE1 FILE2 > FILE3')
- Utilizzando 'cat NOMEFILE' esso presenta interamente il suo contenuto
- 'more' e` piu comodo in quando presenta il contenuto del file con impaginazione sulla shell

Comandi: head, tail, less

- 'head' visualizza l'inizio di un file, di default 10 righe
- 'tail' visualizza la fine di un file
- 'tail -f NOMEFILE' permette di vedere il contenuto del file durante eventuali modifiche al file
- 'less' e` un programma simile a 'more' ma non nativo, che permette una maggiore flessibilita` nella visualizzazione del file

Comandi: who, whoami, id, last

- Il comando 'whoami' o 'id' riporta informazioni sull'utente corrente
- Il comando 'who' elenca gli utenti attualmente presenti nel sistema
- 'last' riporta l'elenco degli utenti che in passato hanno fatto accesso al sistema

Comandi: su, sudo

- Il comando 'su' permette di cambiare utente senza effettuare login/ logout
- Senza l'indicazione dell'utente si intende passaggio a root
- Con l'opzione 'su - ' si forza all'aggiornamento delle variabili di ambiente
- 'sudo' permette di eseguire un comando come se fosse un'altro utente a farlo
- In ubuntu 'sudo - i' permette di aprire una shell con utenza root

Comandi: uptime, top

- 'uptime' informa brevemente sullo stato del sistema
- 'top' monitorizza in tempo reale il sistema
- informazioni sulla memoria, sul carico del processore, e sui processi attivi
- È molto utile per avere un'idea immediata di cosa sta facendo GNU/Linux

Comandi: vmstat, free

- Il comando 'free' illustra brevemente la memoria totale utilizzata dal sistema, sia essa principale (la RAM) che quella aggiuntiva scritta sul disco (lo swap)
- È più dettagliato ma simile il comando 'vmstat'

Comandi: alias - 1

- Il comando 'alias' permette di definire dei comandi personalizzati
- Eseguito senza argomenti elenca i comandi definiti
- È per esempio possibile aggiungere un comando apposta al posto di 'rm - r' con:
- `alias cancellatutto='rm - r'`
- Di solito ls è impostato per elencare i file a colori, ottenuto con:
- `alias ls='ls -- color= auto'`

Comandi: alias, echo

- Attenzione che il comando 'alias' ha la priorit a rispetto ai comandi di base
- Per esempio con 'alias ls=uptime', eseguendo 'ls' si avr  l'output di 'uptime'
- Il comando 'echo' permette di stampare a video una stringa di testo
- Per esempio 'echo "ciao"'

Comandi: ps, pstree

- Il comando 'ps' permette di elencare tutti i programmi in esecuzione nel sistema
- 'ps - eF' elenca tutti i processi attivi
- 'ps fax' elenca i processi organizzati ad albero
- I processi sono indicati con il loro pid, un numero univoco del processo
- Il comando 'pstree' dispone i principali processi ad albero partendo da init, il primo del sistema

Comandi: kill

- Il comando 'kill' permette inviare un segnale ad un processo in esecuzione
- Il segnale di default e` SIGTERM, indicato con il numero 15, che chiede al programma di terminare
- Il segnale SIGKILL, indicato con 9, obbliga il programma a terminare
- Il programma deve essere indicato con il suo pid
- 'kill - 91234'

Comandi: killall, uname

- Il comando 'killall' consente di terminare l'esecuzione di un programma dato il suo nome
- Non è detto che sia solo uno il programma con quel nome (ecco perché 'all')
- Il comando 'uname' fornisce informazioni sul kernel attuale del sistema
- 'uname - a' elenca tutte le informazioni

Comandi: env

- Nella shell e` presente una serie di variabili impostate
- Si accede all'elenco di queste con il comando 'env'
- Per esempio la directory home e` indicata dalla variabile 'HOME'
- 'SHELL' indica l'attuale shell in utilizzo
- 'PATH' contiene l'elenco delle directory dove possono essere trovati i programmi
- Queste variabili possono essere consultate con echo ('echo \$PATH')

Comandi: export

- Per poter impostare una variabile d'ambiente si utilizza 'export'
- 'export INIZIO= "18:30"', 'echo \$INIZIO'
- È spesso utile aggiungere un percorso da cui poter eseguire le applicazioni:
- 'export PATH= \$PATH"/var/bin"'

Comandi: shutdown

- Il comando standard per eseguire lo spegnimento del sistema e' 'shutdown'
- 'shutdown - h now' spegne subito il sistema
- 'shutdown - r now' riavvia ora il sistema
- Esistono poi alcuni comandi piu' specifici:
- 'halt' spegne di forza il sistema
- 'reboot' riavvia di forza il sistema
- 'sync' scrive i dati ancora in memoria sul filesystem per non avere inconsistenze

Filesystem - 1

- Il disco del computer e` diviso in uno o piu` contenitori di dati
- Il contenitore viene chiamato filesystem, su GNU/Linux si utilizza 'ext2', 'ext3', 'reiserfs', etc
- Il filesystem imposta il modo con il quale i dati sono scritti sul disco, e come vengono riportati all'utente
- I filesystem GNU/Linux organizzano i dati in file e directory

Filesystem - 2

- I file sono unita` che contengono dati i piu` svariati, per esempio testuali quale un documento, oppure binari, come un'immagine o un eseguibile
- I file sono raggruppati in directory
- La directory radice e` indicata con '/' e viene chiamata 'root' (non confondere con l'utente amministratore)
- Le directory, assieme ad altri file speciali quali link, socket e pipe, vengono visti come file nel filesystem

Filesystem - 3

- I nomi dei file hanno lunghezza massima 256 caratteri
- Tutti i caratteri possono essere utilizzati, anche se è sconsigliato l'utilizzo quelli speciali
- Il nome del file è case sensitive, per cui una lettera minuscola non equivale ad una maiuscola
- Inoltre quasi tutte le periferiche del sistema vengono viste come file. Fanno eccezione la tastiera e la scheda di rete

Filesystem - 4

- Tutti i file sono dotati di Access Control List (detti 'permessi'), che stabiliscono le operazioni permesse da ogni utente per ogni file
- Il filesystem di un sistema GNU/ Linux permette l'integrazione di altri filesystem provenienti da altri dischi
- L'operazione viene chiamata 'mount'
- Sono anche possibili mount di filesystem presenti su altri computer connessi tramite la rete

Struttura filesystem - 1

- '/' è la directory radice, detta root, unica per ogni sistema GNU/ Linux
- Tutte le altre directory o file discendono da esso
- '/bin' contiene i file eseguibili di molti comandi di base
- '/boot' contiene i file dei kernel e delle immagini di avvio, oltre alle informazioni di LiLo e Grub
- È consigliato che tale directory risieda in una partizione apposita all'inizio del disco

Struttura filesystem - 2

- `/dev` contiene solo file speciali, tra i quali quelli relativi alle periferiche
- Sono file virtuali, non sono davvero presenti
- Il file `/dev/null` a cui puo` essere inviato qualsiasi file o stringa da distruggere
- Il file `/dev/zero` che contiene una successione infinita di 0 binario
- Il file `/dev/random` che contiene una successione infinita di valori casuali
- Per esempio il file `/dev/hda` o `/dev/sda` contiene l'immagine dell'intero disco

Struttura filesystem - 4

- '/proc' contiene vari file contenenti informazioni sul sistema, kernel e processi (anch'essi non davvero presenti sul disco)
- In '/usr' vanno tutti gli eseguibili, documenti, librerie, sorgenti della maggior parte dei programmi
- La maggior parte dei file e` in sola lettura per l'utente normale
- '/usr/bin' contiene comandi utenti di base

Struttura filesystem - 5

- '/usr/sbin' contiene comandi aggiuntivi per l'amministratore
- '/usr/local' contiene applicativi del sistema
- '/usr/lib' contiene librerie di sistema
- '/usr/share' contiene documentazione o librerie comuni a tutti, per esempio '/usr/share/man' contiene i testi dei 'man'
- '/var' contiene file di solito scritti dal kernel o servizi, per esempio i log

Struttura filesystem - 6

- '/' etc' contiene i file di configurazione del sistema, in lettura e scrittura principalmente dall'amministratore e dai servizi, per esempio il file contenente le password
- '/home' contiene le directory delle home degli utenti del sistema
- '/mnt' e '/media' e` la directory in cui vengono posizionati i filesystem aggiunti
- '/opt' serve per alcuni applicativi aggiuntivi

Struttura filesystem - 7

- '/tmp' e` una directory temporanea scrivibile da tutti
- '/root' e` la directory utente dell'amministratore
- Queste directory sono normalmente tutte presenti subito dopo l'installazione di un sistema GNU/ Linux

Permessi - 1

- Tutti i file, directory, link, etc sono dotati di permessi
- I permessi (o ACL) sono degli attributi che limitano l'accesso degli utenti ai file
- Si basano sull'identificazione dell'utente e di gruppi di utenti, e sull'impostazione di privilegi di lettura, scrittura ed esecuzione

Permessi - 2

- Ogni utente che accede al sistema è caratterizzato da un numero uid
- Tale numero può non essere univoco
- Inoltre ogni utente appartiene ad uno o più gruppi, indicati con il numero gid
- Di solito root ha uid 0
- Ogni file appartiene ad un utente e ad un gruppo

Permessi - 3

- Il permesso di lettura (r) indica l'accesso in sola lettura del contenuto del file
- Il permesso di scrittura (w) permette la modifica e cancellazione del file
- Il permesso di esecuzione (x) permette l'esecuzione di un file
- Tutti i programmi (anche i comandi base visti) hanno il permesso di esecuzione
- Il permesso di esecuzione per una directory si traduce nella possibilità di accedervi (per esempio con 'cd NOMEDIR')

Permessi - 4

- Gli eseguibili non sono quindi caratterizzati dall'estensione ma dal permesso 'x'
- I permessi r, w, x, l'utente ed il gruppo di un file possono essere visti con 'ls -l'
- Il primo gruppo di 10 caratteri si compone di:
 - Un carattere che indica il tipo di file, che non può essere cambiato (e` deciso alla creazione)
 - '-' indica file normale, 'd' directory, 'l' un link, 's' socket, ...

Permessi - 5

- Tre gruppi di 3 caratteri nella forma 'rwx'
- Il primo gruppo si riferisce ai permessi del proprietario (u – user)
- Il secondo gruppo indica i permessi del gruppo di appartenenza (g - group)
- Il terzo gruppo indica i permessi degli altri utenti e gruppi (o - others)
- Se la lettera del permesso è presente, il privilegio è concesso, altrimenti è presente un '-'

Permessi - 6

- Segue poi l'indicazione del proprietario del file, la dimension e l'ora di creazione
- Per esempio la stringa '-rw-r--r--' indica un file normale che puo` essere letto da tutti e scritto solo dal proprietario
- Una directory con permesso di lettura ma non di esecuzione ('drw-rw-rw-') permette a tutti di elencare e modificare i file contenuti, ma non permette di accedervi (per esempio con 'cd')

Permessi: chown, chgrp

- I permessi e proprietari possono essere modificati con tre comandi base
- 'chown NOMEUTENTE NOMEFILE' permette di cambiare il proprietario di un file
- Solo root puo` impostare arbitrariamente la proprieta` dei file
- Un utente normale non puo` “regalare” un suo file ad un altro utente
- 'chgrp NOMEGRUPPO NOMEFILE' permette di impostare il gruppo del file

Permessi: chmod - 1

- 'chmod' permette di cambiare la sequenza di permessi di un file
- Ha due modalita` di utilizzo
- Tramite indicazione esplicita dei permessi r, w, x per le utenze u, g, o
- E` possibile assegnare, togliere o aggiungere permessi per l'utente o gruppo
- Esempio 'chmod go- w NOMEFILE'
- Nel caso l'utenza sia omessa, si intende tutti e tre (es 'chmod +rw NOMEFILE')

Permessi: chmod - 2

- Per assegnare esattamente i permessi
'chmod u=rw,go=r NOMEFILE'
- La seconda modalita` consiste nel rappresentare la terna rwx su 8 numeri:
- 0 = - - - , 1 = - - x , 2 = - w - , 3 = - wx , 4 = r - - , 5 = r - x , 6 = rw - , 7 = rwx
- Impostare 'chmod 777 NOMEFILE' e` quindi equivalente a permettere qualsiasi operazione a tutti ('rwxrwxrwx')

Permessi: umask

- È poi presente un programma che si occupa di gestire i permessi di default quando si crea un file. 'umask' senza argomenti riporta la maschera attualmente utilizzata
- La maschera è nel formato numerico di chmod, ma rovesciata
- Per esempio 022 indica che una directory verrà creata con permessi 755, ovvero 'rwxr-xr-x'
- I file normali, che di default non sono eseguibili, sono normalmente privati di 'x'

Collegamenti

- I collegamenti sono gestiti con il comando 'ln'
- Essi sono di due tipi, a seconda della loro implementazione nel filesystem: soft e hard
- Gli hard link permettono di accedere ad un file sul disco da due percorsi differenti
- Essi sono raramente utilizzati, e non permettono collegamenti tra due filesystem di due partizioni differenti

Collegamenti: ln

- I soft link sono creati con il comando 'ln -s FILEORIGINE FILEDESTINAZIONE'
- Il file di collegamento creato è un puntatore a livello di filesystem al file di origine
- Occupa molto poco spazio ed è indicato da 'l' all'inizio della stringa dei permessi
- Accedere al soft link equivale ad accedere al file destinazione