BasicLinux

introduzione rapida all'uso di Linux in ambiente desktop e da terminale

2014.09



Indice

Sottotitolo e nozioni di base	.2
Utenti, diritti, file system, mount e dispositivi	.4
Utilizzo base dell'ambiente grafico GNOME	.7
Nautilus: il file manager di GNOME	.9
Il Monitor di sistema	1
Aggiornare un sistema Linux1	12
La shell a caratteri: l'applicazione Terminale1	4
Alcune proprietà della Bash1	4
Gestire directory e file da terminale1	6
Modificare i permessi con comando da terminale1	8
Ricerca di file	8
File e collegamenti simbolici1	9
Utilizzo dell'editor nano	20
Links: navigazione web con browser testuale	21
Aggiornare il sistema con apt-get	22



Sottotitolo e nozioni di base

È necessario, prima di iniziare a trattare gli argomenti oggetto di appunti riguardanti qualsivoglia argomento, introdurre gli argomenti stessi e, principalmente, la logica con cui vengono trattati (si potrebbe anche dire il target del possibile lettore). In questo caso è tutto sintetizzato nel sottotitolo, per cui la prima cosa che si tratterà sarà una estensione dei termini utilizzati. Ciò servirà pure a introdurre alcuni concetti fondamentali:

🔶 Introduzione rapida

Lo scopo di questi appunti è quello di una introduzione, appunto rapida, senza alcuna pretesa di completezza. In questi appunti si possono trovare indicazioni per avere una idea generale di cosa sia e come si possa operare con Linux utilizzando sia interfacce grafiche che interfacce a linea di comando.

🔶 Linux e distribuzioni

Innanzi tutto occorre chiarire che, quando si parla di Linux, si fa riferimento al kernel di un sistema operativo (la parte che gestisce l'hardware), sviluppato da Linus Torvalds (ai tempi studente universitario finlandese e, tuttora, principale responsabile dello sviluppo e manutenzione di Linux coadiuvato da un grande numero di persone sparse in tutto il mondo), e rilasciato sotto licenza GPL. Questa è un tipo di licenza che obbliga il rilascio anche del sorgente e concede diritti di modifica, redistribuzione e sviluppo di nuovo software a partire da quello attuale, a condizione che la parte del software soggetto alla licenza GPL rimanga tale: vengono conservati, cioè, i diritti (per maggiori chiarimenti http://www.fsf.org).

Il kernel è indubbiamente indispensabile per poter usare il computer, ma da solo, sicuramente, non è sufficiente. Se non altro servono anche un minimo di programmi applicativi. È questo il motivo per cui, da una certa epoca in poi, nascono le *distribuzioni*. Le distribuzioni sono pacchetti, pronti all'uso, comprendenti il kernel e tutta una serie di applicativi per gli utilizzi più disparati: applicazioni di tipo office (word processor, tabelloni elettronici, ecc..), giochi, grafica, programmi di utilità. In generale ogni distribuzione ha un'*anima*: per gli applicativi da includere in essa si sceglie, accanto al software di uso comune, di privilegiare un determinato utilizzo. Esistono quindi distribuzioni orientate ad ambienti desktop, ambienti di rete e così via.

Le distribuzioni sono assemblate da appassionati o da aziende che talora offrono servizi anche di assistenza e, generalmente, si occupano di rendere coerente l'assemblaggio delle parti software contenute nella distribuzione.

La maggior parte del software compreso nelle distribuzioni è di tipo GPL, essendo questa la licenza più diffusa in ambiente Linux, anche se in alcuni casi alcune distribuzioni comprendono software proprietario. In genere i curatori della distribuzione, per rendere coerente tutto il pacchetto, aggiungono delle proprie utility: per esempio è abbastanza frequente la personalizzazione del programma di installazione della distribuzione stessa o dell'ambiente grafico. In ogni caso qualunque sia la distribuzione e qualunque sia la lista dei programmi scelti e inclusi nella stessa, è bene tenere presente che non si tratta di un insieme chiuso e intoccabile. Piuttosto la distribuzione è da intendersi come *punto di partenza* che permette di avere un certo insieme, che magari privilegia per la scelta un determinato uso del computer, di applicativi già installati e pronti all'uso. L'aggiunta di ulteriore software che

si ritenesse utile, anche quando non previsto dalla specifica distribuzione, è sempre facilmente possibile in qualsiasi momento.

È opportuno, infine, fare notare che la numerazione associata alle varie distribuzioni è relativa alla distribuzione stessa: la ipotetica *DistroA 10.0* è sicuramente un nuovo rilascio (*release*) aggiornamento della *DistroA 9.9* ma può essere contemporanea, per esempio, della *DistroB 2.0* e quindi, in generale, viene accompagnata dalla numerazione della versione del kernel o degli applicativi compresi nella distribuzione stessa.

Questi appunti fanno riferimento alla distribuzione Ubuntu nella versione GNOME per quanto riguarda la parte che tratta l'interfaccia grafica. L'operatività da terminale è invece generalizzabile a qualsiasi tipo di distribuzione.

Nel caso della famiglia Ubuntu la numerazione è associata all'anno e al mese in cui viene rilasciata. Così Ubuntu xx.yy è la versione rilasciata nel mese yy dell'anno 20xx.

🔶 Uso in ambiente desktop

Inizialmente Linux è stato un sistema operativo utilizzato principalmente in ambienti di rete (la maggior parte dei server internet è operante con questo sistema operativo) o in ambienti di ricerca (Università o appassionati). Successivamente, grazie agli sforzi della comunità di sviluppatori, si è curato l'adattamento del sistema operativo anche a macchine utilizzate in *ambiente desktop*. Con tale termine si intendono quei computer, destinati ad uso quasi esclusivamente personale, in cui girano le applicazioni più comunemente utilizzate nelle attività quotidiane come, per esempio, accedere ad Internet, scrivere una lettera o mantenere i conti di casa o del proprio ufficio: l'insieme, cioè, di quelle che vengono anche talvolta chiamate applicazioni SOHO (Small Office Home Office). Tale destinazione, visto il tipo di utente cui si fa riferimento, presuppone una interfaccia del sistema operativo che metta in condizione anche l'utente non specialista di utilizzare il computer e i suoi strumenti.

➡ Uso da terminale

L'uso di un computer con sistema operativo Linux, specie se si tratta di computer su cui girano programmi server, richiede di potersi orientare anche da terminale. In questo tipo di computer spesso non si trovano interfacce grafiche:

- per motivi di sicurezza. Si tende a far girare sul computer dedicato ai server solo i software strettamente indispensabili agli utenti che richiedono i servizi (e l'interfaccia grafica non lo è), secondo il principio che meno sono i programmi in esecuzione meno sono i bug dei programmi che possono creare malfunzionamenti
- per motivi di prestazioni. È meglio riservare tutte le risorse disponibili per i servizi che si vogliono offrire ai client. La grafica toglie molte risorse senza fornire agli utenti dei servizi alcun vantaggio.

Linux nasce come porting di Unix su macchine con architettura Intel. Lo scopo originario di Linus Torvalds era quello di scrivere una versione di Unix, sistema operativo che governava grossi sistemi di elaborazione, che fosse in grado di girare su un comune personal computer con architettura Intel.

Il sistema Unix usava, come comune, *shell* a linea di comandi. Per mezzo di una shell di questo tipo si interagisce con il sistema utilizzando un ambiente che consente di introdurre comandi che, successivamente, vengono interpretati e mandati in esecuzione ad opera della shell stessa. Anche per Linux furono sviluppate diverse shell, con caratteristiche *Open Source*. rilasciate con licenza

GPL, la più utilizzata delle quali è la shell *bash*, cui si farà riferimento anche in questi appunti. Tuttavia per consentire una più facile interazione con l'utente viene successivamente sviluppata una versione free del server X-Window, funzionante in Unix.

È bene spendere qualche parola in più su tali problematiche, viste le implicazioni che ne conseguono. In poche parole la logica è quella di un programma (il server X) che mette a disposizione degli altri programmi applicativi (i client), servizi che riguardano la gestione grafica dello schermo. I servizi offerti dal server X sono quindi minimi. Al server grafico X si appoggiano i window manager che gestiscono la decorazione delle finestre e i *desktop manager*: programmi che mettono a disposizione tutto quello che può servire per la manutenzione indispensabile di un computer e quindi *file manager* per la gestione dell'organizzazione dei documenti creati dagli utenti, *tools* per la personalizzazione dell'ambiente di lavoro, *pannelli* e *menù* per il lancio rapido di applicazioni e quanto altro; e quindi, in definitiva, un ambiente coerente all'interno del quale girano tutti i programmi (applicativi e utility) che si ha necessità di utilizzare.

L'architettura client-server di X spiega la presenza, per Linux, di diversi ambienti desktop e quindi, in definitiva, diverse possibili modalità di interagire con il S.O. e diversi aspetti grafici. Esiste una vasta gamma di possibilità di scelta di ambienti desktop che va dai più spartani con solo le funzioni del window manager e dedicate a macchine di una certa età o, in genere, con risorse limitate, fino ad ambienti con aspetti grafici molto sofisticati. La famiglia Ubuntu, a partire dalla stessa base di gestione e, anche se in parte, di pacchetti software, rende disponibili diverse versioni, mantenute dal team di sviluppo o da gruppi di sviluppatori autonomi, con le più diffuse interfacce grafiche: Unity (Ubuntu stessa: l'interfaccia di default sviluppata dal team di sviluppo e manutenzione della distribuzione), KDE (Kubuntu), Xfce (Xubuntu), LXDE (Lubuntu), GNOME (ubuntu GNOME), MATE (ubuntu MATE), Cinnamon (Cubuntu). Gli ambienti grafici hanno delle funzionalità in comune derivanti dagli studi sulle interfacce utente sviluppatesi verso gli ani 80, ma anche logiche e operatività diverse. La scelta in genere è fra quegli ambienti che forniscono più funzionalità ma richiedono risorse maggiori e altri con qualche funzionalità in meno ma più leggeri e veloci ma, soprattutto, in dipendenza di preferenze individuali: si può scegliere, in definitiva, il proprio *ambiente su misura*, su come si intende cioè la propria interazione con un sistema di elaborazione.

In questi appunti si tratteranno le caratteristiche generali dell'ambiente desktop GNOME.

Utenti, diritti, file system, mount e dispositivi

Linux come sistema operativo multiutente prevede la coabitazione, all'interno dello stesso computer da parte di più utenti e ciò comporta una serie di conseguenze principalmente riguardanti la sicurezza. Ogni utente deve poter utilizzare la macchina senza interferire con gli altri utenti e quindi ogni utente opera, per esempio, su una zona precisa della memoria di massa distinta da quella utilizzata da un altro utente, utilizza dei propri documenti che, se vuole, può anche decidere di rendere accessibili agli altri, ma che in ogni caso sono di *sua proprietà*. Anche l'ambiente desktop può essere personalizzato da ogni utente in maniera indipendente dagli altri.

- Account e login

Per poter utilizzare la macchina occorre essere riconosciuti, essere aggiunti alla lista degli utenti abilitati ad usare le risorse dell'hardware. Ogni utente sceglie un identificativo (il nome utente) e una password (parola segreta) che serviranno a farsi riconoscere e che lo qualificheranno come colui che, per esempio, può operare in una zona determinata dell'hard disk, può utilizzare determinati programmi ecc. Il sistema, all'accensione della macchina, chiede l'inserimento dei dati dell'utente, se tali dati corrispondono a quelli di un utente riconosciuto, viene permesso l'accesso e si può utilizzare il sistema (effettuare il login).

→ Utente root o superuser

Il nome utente root identifica l'amministratore di sistema cioè colui che ha accesso illimitato alle risorse del sistema. L'utente root, per esempio, crea i nuovi utenti e stabilisce i diritti di cui debbano godere, può installare programmi, può anche, agendo in modo sbagliato, danneggiare il sistema. Qualunque altro utente, non root, può solo danneggiare i file di sua proprietà quelli cioè creati da lui, ma non i file del sistema

Anche in una macchina, funzionante in ambiente desktop e quindi utilizzata da un solo utente, principalmente per motivi di sicurezza, viene generato dal sistema, accanto all'utente *root*, un utente con diritti standard. Si utilizzerà l'account normale per le attività quotidiane di utilizzo del computer, riservando i privilegi dei diritti di *root* solo per le operazioni, temporanee, di manutenzione del sistema. Inizialmente questa era l'impostazione di default usata da tutti i sistemi Linux. La diffusione dei sistemi desktop ha suggerito una altra scelta di gestione. Per evitare, quando serve, di acquisire i diritti di root, in ogni caso pericolosi in relazione al tipo di danni che si possono causare, si è preferito utilizzare un sistema che attraverso la richiesta della password dell'utente permette di eseguire singole operazioni che richiedono i diritti di root senza, però, concedere, in generale, i diritti associati all'account acquisendone, invece, i diritti per un brevissimo tempo: giusto quello che serve all'esecuzione di un singolo comando.

Dal punto di vista dell'utente le informazioni, anche quelle da lui stesso generate (per esempio un documento di testo prodotto con un applicazione di elaborazione testi), sono conservate nelle memorie di massa in unità chiamate file, ognuno con un nome che permette di rintracciarle. Poiché poi nelle memorie di massa possono essere conservati un numero elevato di file, Linux mette a disposizione la possibilità di gestire le *directory* (negli ambienti grafici vengono chiamate *cartelle*). In una prima approssimazione una directory può essere considerata come un contenitore all'interno del quale possono essere presenti sia file che altre directory (sotto-directory) in una struttura ad albero: dalla radice partono diversi rami che, a loro volta, hanno altre diramazioni e così via. In questo modo si possono creare dei raggruppamenti logici di file che permettano di rendere più agevole il reperimento delle informazioni. Così, per esempio, si potrebbe avere una directory contenente le lettere redatte, una directory contenente le registrazioni dei movimenti contabili, e così via. Ogni directory ha un nome, scelto dal sistema o dall'utente a seconda del proprietario della stessa. È, inoltre, opportuno chiarire che la directory non è un contenitore che ha una dimensione fisica definita: è un raggruppamento di file e la capacità dipende soltanto dalla capacità della memoria di massa su cui risiedono i file. L'organizzazione di directory e sotto directory è chiamata filesystem.

Effettuata la procedura di *login* il sistema porta l'utente in un punto particolare del filesystem dove si è abilitati a conservare i propri dati.

/ -- bin |-- boot |-- dev |-- etc |-- home | |-- utente1

```
| |-- utente2
| `-- tux
|-- lib
|-- media
|-- mnt
|-- opt
|-- proc
|-- sbin
|-- sys
|-- usr
|-- var
```

L'albero delle directory inizia dalla principale, la *directory radice*, che viene identificata dal simbolo /; da questo punto si diparte l'albero delle directory. Per identificare una directory si specifica il *cammino* (**path**) per raggiungerla a partire dalla radice, cioè la posizione gerarchica della directory interessata all'interno dell'intero file system. Per esempio /home/tux specifica che a partire dalla directory radice (la /) c'è una directory home all'interno della quale c'è una ulteriore directory chiamata tux. Per ogni utente registrato il sistema genera una directory, con lo stesso nome dell'utente, agganciata alla /home. Nell'esempio di prima, l'utente con identificativo *tux*, una volta effettuato il login, viene portato dal sistema nella /home/tux cioè la *home* (la directory principale) dell'utente *tux* (per inciso *tux* è il nome del pinguino mascotte di Linux). All'interno della propria *home*, l'utente può costruire, a seconda delle esigenze specifiche, il proprio albero di directory.

Directory	Contenuto
/bin	Comandi del sitema
/boot	Configurazioni e file per l'avvio del sistema
/etc	File di configurazione del sistema
/lib	Librerie e moduli del kernel
/media /mnt	Albero dei dispositivi esterni accessibili: cdrom, dvd, chiavette usb
/opt /usr	Pacchetti software installati e albero dei pacchetti addizionali
/proc	Informazioni sui programmi in esecuzione
/sbin	Comandi di sistema generalmente utilizzati dall'utente root
/sys	Informazioni sui dispositivi collegati al sistema
/tmp /var	File e dati variabili

Ad ogni dispositivo di memorizzazione presente nel computer (hard disk, cd-rom, partizioni dell'hard disk) Linux associa un *device*, cioè un file speciale che si trova nella /dev. Per esempio:

Device	Dispositivo
/dev/sda	Hard disk installato come primary master o primo HD
/dev/sdb	Hard disk installato come primary slave o secondo HD
/dev/sd	Altri hard disk
/dev/cdrom /dev/dvd	Ulteriori nomi di dispositivi, eventualmente disponibili, per accedere a lettori ottici o altro

Device	Dispositivo
/dev/sda1	Prima partizione di sda
/dev/sda2	Seconda partizione di sda

Per poter accedere alle informazioni conservate nei dispositivi di memorizzazione occorre associare il device ad una directory del file system. Tale operazione viene effettuata facendo il *mount* del dispositivo. Il vantaggio di questo modo di procedere è quello di operare su un unico file system senza preoccuparsi dell'allocazione fisica dei files.

Gli ambienti grafici mettono a disposizione dei meccanismi per agevolare la facilità di gestione dei dispositivi per cui l'operazione di mounting di essi, necessaria per l'accesso ai dati contenuti, viene effettuata in automatico in seguito all'inserimento del supporto sia questo, per esempio, un CD-ROM o una chiavetta USB.

Utilizzo base dell'ambiente grafico GNOME



Il desktop di GNOME è quasi totalmente dedicato a contenere le finestre con gli applicativi avviati tranne per un pannello nella parte superiore che sulla destra (1) mostra alcune icone di notifica riguardanti lo stato e il tipo di rete, il volume di uscita del suono e una piccola freccia indicante che si può cliccare e fare aprire un menù (2) da cui si possono impostare: il volume di uscita del suono, collegamenti con diversi tipi di rete disponibili ... Nel menù, oltre al pulsante che permette il *logout* e lo spegnimento della macchina, è presente un pulsante (3) per l'apertura del pannello di impostazioni hardware (reti, stampanti ecc...), personalizzazioni varie di GNOME. Le impostazioni, assieme alla modifica del wallpaper, sono accessibili anche dal menù contestuale visualizzato di seguito al clic destro del mouse, sul desktop.

Il pulsante *Attività* (4) sulla sinistra del pannello abilita la schermata delle attività, il cuore del sistema GNOME, e anche il posto dove sono più evidenti le differenze di utilizzo rispetto ad altri sistemi desktop.

La schermata delle attività si abilita anche portando il mouse nell'angolo in alto a sinistra dello schermo o premendo il tasto <SUPER> (quello con la bandierina, a sinistra della barra spaziatrice). Si ritorna al desktop di lavoro premendo nuovamente *Attività*, <SUPER> o <ESC>.



Nella parte centrale della schermata delle *Attività* sono visualizzate le finestre aperte nell'area di lavoro attuale ed è possibile selezionare quella che dovrà essere attiva. Mano a mano che vengono inseriti caratteri da tastiera questi compaiono nella barra di ricerca **5** e la visualizzazione è sostituita dagli applicativi, file, documenti (le attività) che contengono all'interno del nome i caratteri digitati, in ordine di uso (da quelli più recentemente utilizzati a quelli meno) e si può selezionare quella desiderata.

Sulla sinistra (6) è presente una barra (la *dash*) dove si possono posizionare le applicazioni che si intendono utilizzare più frequentemente in modo da averle facilmente a portata di clic. Il menù contestuale sulla singola icona permette di rimuovere una applicazione dai *Preferiti* (dalla dash). Ogni applicazione avviata ha la sua icona nella dash e, in questo caso, dal menù contestuale è possibile aggiungerla fra i *Preferiti*.

Selezionando l'icona 7 si può navigare fra le applicazioni installate. Dal menù contestuale, attivabile con il clic destro, di ogni icona rappresentante una applicazione, si può scegliere di inserirla fra i *Preferiti*.

La barra **8**, normalmente visualizzata come una striscia ed espansa come in figura quando vi si porta il mouse sopra, mostra i desktop virtuali attualmente esistenti. Ogni desktop virtuale è un ambiente diverso che può contenere le proprie applicazioni avviate (e questa è una proprietà importante

perché permette di avere le applicazioni raggruppate e di non affollare in modo ingestibile il desktop). I desktop sono *impilati* uno sopra l'altro e sono gestiti dinamicamente: tutte le volte che si apre una applicazione in un desktop vuoto, il sistema provvede a generare un nuovo desktop *sotto* quello su cui si sta in questo momento. Se viene chiusa l'ultima applicazione presente nel desktop, questo viene eliminato e un eventuale desktop *sotto* con attività aperte viene spostato verso l'alto: non esisterà mai un desktop vuoto fra due che contengono attività aperte ed esisterà sempre almeno un desktop. Nella **8**, una finestra si può trascinare da un desktop ad un altro.

Nel desktop di lavoro sono attive alcune combinazioni di tasti per la gestione dei desktop o delle finestre:

Combinazione	Effetto prodotto
<ctrl><alt><giu>/<ctrl><alt><su></su></alt></ctrl></giu></alt></ctrl>	Ci si sposta, rispettivamente, al desktop sotto o a quello sopra.
<super>A</super>	Mostra direttamente l'elenco delle applicazioni installate da cui si può selezionare quella da avviare.
<super><su>/<super><giu></giu></super></su></super>	Espande la finestra attiva a tutto lo schermo/Ripristina le dimensioni della finestra.
<super><dex>/<super><sin></sin></super></dex></super>	Sono utilizzati per affiancare le finestre. La prima combinazione porta la finestra attiva ad occupare la metà destra del desktop, la seconda la metà sinistra.

La combinazione <ALT><TAB> lancia la Gnome-Shell:



A prescindere dal desktop su cui girano, tutte le applicazioni attive sono accessibili per mezzo di una barra che si sovrappone al desktop, e che rimane attiva finché resta premuto <ALT>. Utilizzando <ALT><TAB> o le frecce si può navigare fra le varie applicazioni. Se esistono più finestre relative ad una unica applicazione, queste sono accessibili via una sotto-barra associata all'icona dell'applicazione. Quando si raggiunge, durante la navigazione, l'applicazione desiderata, si rilascia <ALT> e l'applicazione passa in primo piano.

Nautilus: il file manager di GNOME

L'attività principale che si svolge con un computer è quella di produrre e organizzare i documenti nelle memorie di massa. Ogni interfaccia utente è accompagnata dal proprio *file manager* per la gestione dei file. A parte le funzionalità comuni e facenti riferimento alle caratteristiche standard di applicativi del genere (creazione cartelle, copia o cancellazione, anche multipla, di file, navigazione all'interno delle varie memorie di massa, ...), in questo paragrafo si vogliono esporre alcune caratteristiche tipiche di Nautilus che possono incrementare la produttività quotidiana.



Il gruppo di pulsanti 1 è dedicato alle impostazioni della finestra di Nautilus. Il pulsante a sinistra regola la visualizzazione degli oggetti interni alla finestra permettendo di specificarne diverse modalità o ordinamenti. Per mezzo dei pulsanti 2 si possono scegliere velocemente due modalità di visualizzazione che risultano fra quelli più comodi (*griglia di icone*, quella abilitata di default, o *elenco*). Il pulsante a destra di 1 riguarda invece le opzioni sulla finestra nel suo insieme come contenitore.

Il pulsante **3** è utile quando si cerca qualcosa in una directory affollata. Si abilita una barra all'interno della quale si possono inserire le lettere che compongono il nome del file cercato. La barra è visualizzata anche senza premere il pulsante: basta, quando è abilitata la finestra di Nautilus, inserire i criteri di ricerca direttamente da tastiera. Mano a mano che si digitano caratteri la finestra si aggiorna selezionando i file che soddisfano i criteri di ricerca.

All'interno della stessa finestra del file manager si possono visualizzare (4) più schede, ognuna delle quali indipendente rispetto alle altre, in modo da rendere più efficace l'operatività fra le stesse. Una nuova scheda si abilita con la combinazione di tasti <CTRL>T o selezionando l'opzione dal menù associato al pulsante a destra nel gruppo 1. Si può anche lanciare un duplicato della finestra con <CTRL>N.

In **5** è mostrato, come sequenza di pulsanti, il cammino che porta alla directory il cui contenuto è visualizzato nella scheda attuale. Ogni pulsante permette velocemente di andare alla directory specificata o si può trascinare nella sezione *Segnalibri* (**6**) in modo da avere a portata di clic le posizioni da raggiungere rapidamente.

Un altro modo per raggiungere rapidamente una determinata posizione all'interno dell'albero delle directory, è quella di aprire il menù associato all'icona del file manager visualizzata nel pannello di GNOME e selezionare *Inserisci posizione*.



Il file manager visualizza, al posto dei pulsanti 5, la barra 7 dove inserire il cammino che individua la posizione da raggiungere.

Nel riquadro laterale (6), nella sezione *Dispositivi*, compaiono anche le chiavette USB, o altre periferiche, quando inserite. Cliccando sopra si può accedere al contenuto. La freccetta che compare a fianco del dispositivo serve per disconnettere il dispositivo in modo da non correre il rischio di perdere dati.

Il Monitor di sistema

Una applicazione particolarmente interessante è il *Monitor di sistema* (basta cominciare a scriverne il nome nel box della schermata delle *Attività*).

			Monitor	di	sistema				
Processi Risorse	FI	le system							
Carico medio per gli ultimi 1, 5	5, 15 π	inuti: 0,32	, 0,24, 0,1	15				Aggioma	Visualizza 🗸
Nome del processo	~	Utente	% CPU		ID	Memo	Priorità		
lat-spi-bus-launcher		nunzio		0	1697	4,4 MB	Normale		
💷 bash 📐		nunzio		0	2041	1.6 MB	Normale		
at hinhish		nunzio		0	2234	68.0 Ki	Normale		
@ dbus-daemon		nunzio		0	1701	424.01	Normale		
dbus-daemon		nunzio		0	1622	1,5 MiB	Normale		
dbus-daemon		nunzio		0	1600	272,01	Normale		
dbus-launch		nunzio		0	1595	244,01	Normale		
W dconf-service		nunzio		0	1886	496,01	Normale		
deja-dup-monitor		nunzio		0	2662	4,7 MB	Normale		
				0	2236	O A MR	Mormale		

Nella scheda *Processi* dell'applicazione sono mostrati i programmi in esecuzione in un determinato istante (dal menù *Visualizza* si può scegliere quali programmi visualizzare), con informazioni sulla percentuale di CPU impegnata e la memoria utilizzata da ciascun programma. In questo modo si possono monitorare i programmi che consumano più CPU o memoria.

Selezionando un programma e premendo il pulsante *Termina processo* se ne può forzare la terminazione quando, per esempio, il programma non risponde.

Nella scheda *Risorse* viene monitorata istante per istante l'attività della CPU, l'occupazione di memoria centrale, l'uso della memoria virtuale e l'attività di rete.

La scheda File system invece mostra l'occupazione e lo spazio vuoto per ogni partizione disponibile.

Aggiornare un sistema Linux

Ogni distribuzione installa nel computer una serie di applicazioni, ma qualunque sia la distribuzione scelta e i programmi che essa installa, l'utente può avere l'esigenza di utilizzare ulteriori software.

Il software per Linux è disponibile in formato sorgente (per quello con licenza GPL) che può essere compilato sulla propria macchina. È comune, comunque, trovare gli applicativi in formato di **pacchetti** pronti per essere installati. Ogni pacchetto contiene una applicazione o una parte di una applicazione. I pacchetti sono, in massima parte anche se non solo, disponibili in due formati: il formato DEB tipico delle distribuzioni derivate dalla Debian e il formato RPM tipico delle distribuzioni derivate dalla Red Hat. Debian e Red Hat sono due delle prime distribuzioni assemblate per Linux. La famiglia Ubuntu, derivata da Debian, utilizza il formato DEB.

I pacchetti sono conservati in archivi presenti in rete e da cui possono essere scaricati per poter essere installati. Un archivio di questo tipo viene chiamato **repository**. L'installazione di una applicazione da un repository è facilitata da servizi presenti, generalmente, nella distribuzione installata.

I manutentori delle distribuzioni della famiglia Ubuntu mettono a disposizione, quasi quotidianamente, aggiornamenti del software installato. Quando si è on-line, se non è visualizzata una finestra con l'elenco degli aggiornamenti disponibili, si può richiamare il servizio dalla pagina delle *Attività* (1):



Per mezzo di questo servizio è garantito anche il passaggio a un nuovo rilascio della distribuzione, quando disponibile.

Un sistema semplice per installare, invece, nuovi programmi è quello di utilizzare *Ubuntu Software Center* (2):



Si tratta di un software semplice da utilizzare e che permette di scegliere l'applicazione fra quelle

presenti nei repository e installarla con un semplice clic del mouse, basta essere connessi.

Per gli utenti che hanno necessità di installare singole librerie o di effettuare scelte più dettagliate si può utilizzare direttamente Apt-get, il sistema presente nelle distribuzioni derivate da Debian fra cui Ubuntu, e che sta alla base anche di Ubuntu Software Center.

Negli ambienti grafici, di qualsiasi distribuzione Linux, sono disponibili diverse interfacce grafiche (**front-end**) che permettono all'utente di interagire con il sistema apt-get (che in questo caso sarà il **back-end**) utilizzando il mouse. Una di queste interfacce, tipica di Ubuntu, è Ubuntu Software Center, un'altra, comune a tutte le distribuzioni, è Synaptic che si può avviare selezionando *Gestore pacchetti*:

ile Modifica Pacchetto Im	postazioni Ajuto			
Ricarica Marca tutti gli agg	iomamenti Applica Prop	orietă Cerca		
Tutti	S Pacchetto	Versione installata	Ultima versione	4
Ambiente desktop GNOME	acovea-results		20040329-1	
Ambiente desktop GNOME (🔲 🔮 acpi	0.09-3ubuntu1	0.09-3ubuntu1	1
Ambiente desktop KDE /mid	🔳 🔮 acpid	1.0.4-5ubuntu9.3	1.0.4-5ubuntu9.3	
Ambiente desktop KDE (mul	acpidump		20061130-1	
Amministrazione di sistema	🔳 🔮 acpi-support	0.109-0hardy2	0.109-0hardy2	
Amministrazione di sistema	acpital		0.1-1	_
Parterna agrorie di siscerna				٠
Sezioni Stato	Mostra informazioni Tenta di replicare la funzion ACPI, incluse le informazion sospensione ACPI, visualizz	altà del "vecchio" coman ni su batteria e ventole. N ta solo informazioni sui de	ido apm su sistemi Ion supporta la evice ACPI.	
Origine				
Altri personalizzati				
Carron ten i sim dtati				

Il pannello di sinistra di Synaptic mostra le categorie nelle quali sono raggruppate le applicazioni. Nella parte destra della finestra è elencato, nella zona superiore, in ordine alfabetico, il software contenuto nella categoria selezionata e, nella zona inferiore, una breve descrizione del contenuto del pacchetto selezionato.

Per le applicazioni installate è visualizzata la versione installata ed, eventualmente, l'ultima versione disponibile presente nei repository.

Normalmente la prima operazione che dovrebbe essere svolta sarebbe quella di configurare i repository da dove prelevare il software. In generale nella distribuzione installata sono configurati quelli ufficiali, ma se ne possono aggiungere altri in conseguenza della necessità di installazione di particolari software selezionando *Impostazioni, Repository* e spuntando le voci non selezionate o aggiungendone dalla scheda *Altro software*. È bene però fare attenzione a che il software contenuto nei repository aggiunti, se non sono quelli *ufficiali* della distribuzione, non sia in contrasto con il parco software installato.

Prima di selezionare il software da installare o da aggiornare ad una versione più nuova, è opportuno, dalla finestra principale di Synaptic, premere il pulsante *Ricarica* o, in alternativa, scegliere la voce *Aggiorna informazioni sui pacchetti* dal menù *Modifica*. Questa opzione fa in modo che Synaptic si colleghi ai repository e scarichi l'elenco aggiornato delle applicazioni permettendo, in questo modo, la sincronizzazione dell'elenco dei pacchetti visualizzati nella finestra con gli elenchi contenuti attualmente nei repository.

Per installare un pacchetto basta fare doppio clic con il tasto sinistro del mouse sopra il nome. Synaptic si occuperà di **risolvere le dipendenze**: selezionare, cioè, anche altri pacchetti necessari al corretto funzionamento del pacchetto scelto.

L'elenco dei pacchetti che verranno installati viene visualizzato in una nuova finestra assieme ai pacchetti, qualora ci fossero, che necessitano di aggiornamento per far funzionare l'applicazione da installare.

L'installazione dei pacchetti viene avviata in seguito alla selezione del pulsante *Applica* o alla scelta dell'opzione *Applica le modifiche selezionate* dal menù *Modifica*.

Synaptic si collegherà ai repository interessati, scaricherà i pacchetti, li installerà nel sistema e avviserà l'utente della fine del processo.

La shell a caratteri: l'applicazione Terminale

Selezionando *Terminale* nel box di ricerca delle *Attività* può venire avviata una sessione di lavoro della shell a caratteri.

La shell a caratteri è l'interfaccia tradizionale dei sistemi Unix. Un interprete si occupa di mandare in esecuzione i comandi digitati dall'utente. L'insieme dei comandi che si possono utilizzare dipende dall'interprete scelto. Nel caso di Linux è avviata per default la Bash (Bourne Again SHell). Si tratta di una estensione prodotta ad opera della Free Software Foundation della shell Bourne esistente nei sistemi Unix.



Nella Bash la directory di lavoro dell'utente è indicata dal simbolo ~. Con tale simbolo viene, in ogni caso, identificata la directory di lavoro dell'utente connesso al sistema.

Il simbolo \$ indica il *prompt* (segnale di attesa di un comando da eseguire) di un utente con normali permessi, il prompt dell'utente *root* è il simbolo #.

Il prompt indica che la shell è pronta per ricevere i comandi. L'utente può digitare un comando e premere il tasto <INVIO> per permettere all'interprete di avviare l'esecuzione del comando.

Negli esempi di uso del Terminale che seguiranno, i caratteri digitati da tastiera dopo il prompt saranno evidenziati in grassetto mentre le risposte del sistema saranno in caratteri normali.

Alcune proprietà della Bash

Man pages. Si tratta di un manuale in linea per i comandi della shell e non solo. Per esempio digitando, al prompt, man ls viene visualizzata la pagina, del manuale, relativa al comando ls con una spiegazione dei diversi modi ammissibili di uso del comando. Con la pagina man visualizzata, si può navigare nel testo usando i tasti <PagGIU> per andare avanti di una schermata e <PagSU> per tornare indietro di una schermata. Quando si vuole uscire dalla pagina del manuale, si digita il tasto q (quit). Il comando man man mostra la pagina di utilizzo dell'utility man con tutte le possibilità offerte: per esempio la formattazione delle pagine per la stampa.

Del sistema di aiuto fa parte anche l'utility apropos che permette di elencare tutte le pagine man in cui si parla dell'argomento: apropos directory, per esempio, propone tutte le pagine che trattano le directory.

Tab completion. La shell Bash è dotata di un sistema di *completamento automatico*: basta scrivere le prime lettere del comando e, successivamente premendo il tasto <TAB>, la riga viene completata. Per esempio, se esiste la directory prova e si vuole cambiare la directory corrente con questa, basta digitare, al prompt, cd pr premere il tasto <TAB> e l'interprete completerà la riga con cd prova. Se ci fossero più nomi che cominciano con pr, la shell non può completare la riga e si limita a dare un segnale sonoro, tuttavia una successiva pressione del tasto <TAB> mostra un elenco di tutti i possibili completamenti cioè, nel caso specifico, di tutti i file il cui nome inizia con la coppia di lettere digitate. Il completamento automatico è utile anche nel caso in cui non ci si ricordi esattamente il nome di un comando, ma, per esempio, ci si ricordi che comincia con la lettera c. In questo caso basta, dopo aver digitato la lettera c, premendo <TAB> per 2 volte viene visualizzato l'elenco dei comandi che cominciano con tale lettera.

Command history. La shell tiene nota degli ultimi comandi digitati qualora sia necessario ripeterne qualcuno, cosa che capita spesso: ci sono dei comandi che vengono utilizzati frequentemente. Digitando al prompt history viene mostrato l'elenco degli ultimi comandi digitati. Ogni comando è preceduto da un numero. Si possono visualizzare anche, per esempio, gli ultimi 10 comandi con il comando history 10. Per eseguire nuovamente, per esempio, il comando associato al numero 15, basta digitare, al prompt !15. Se al prompt si preme tasto <SU>, vengono visualizzati, pronti per essere eseguiti, i comandi presenti nella history.

Un uso interessante dello storico dei comandi digitati è quello che permette di effettuare una ricerca indietro al fine di evitare di riscrivere lunghe righe di comando. Se al prompt si preme la combinazione di tasti <CTRL>R si può scrivere la sequenza di caratteri che si ricerca:

(reverse-i-search)`ba': cp basiclinux.odt basiclinux1.odt

mano a mano che si digitano i caratteri, viene riproposta la riga presente nello storico che contiene i caratteri, con una ricerca all'indietro. Trovata la riga interessata basta premere un tasto cursore e la riga viene proposta al prompt pronta per essere eseguita o modificata.

Sono definite parecchie combinazioni di tasti che permettono un uso efficiente e veloce della Bash. Nella tabella successiva vengono riportate quelle che possono avere utilizzi più frequenti.

Per l'elenco completo delle caratteristiche della Bash consultare la pagina di manuale man bash.

Combinazione tasti	Effetto prodotto
<ctrl>R</ctrl>	Effettua una ricerca all'indietro nello storico dei comandi digitati
<ctrl>U</ctrl>	Cancella dall'inizio della riga alla posizione del cursore
<ctrl>K</ctrl>	Cancella dalla posizione del cursore alla fine della linea
<ctrl>L</ctrl>	Cancella la pagina e porta il prompt alla prima riga. Lo stesso effetto si ha digitando il comando clear

Una cosa che è bene tenere presente è il fatto che la Bash non è *chiacchierona*: l'esecuzione di un comando, se la sintassi utilizzata è corretta, avviene immediatamente e non c'è presentazione di

alcun messaggio di ritorno o preventivo rispetto all'esecuzione. Questo comporta un supplemento di attenzioni da parte dell'utente, in generale, e di root in particolare, quando si lanciano comandi che comportano modifiche o cancellazioni, al fine di non danneggiare dati in maniera involontaria.

Gestire directory e file da terminale

Linux distingue fra caratteri minuscoli e maiuscoli. Tutti i comandi inviati alla Bash devono essere scritti utilizzando <u>soltanto</u> caratteri minuscoli.

I comandi per la gestione di file richiedono che li si specifichi in maniera dettagliata. Come regola generale per accedere ad un file è necessario specificare il nome e il *path* (il cammino che deve essere percorso a partire dalla radice per raggiungere la posizione dell'hard disk dove si trova il file).

Il cammino può essere specificato anche partendo dalla posizione, all'interno dell'albero delle directory, in cui ci si trova (cammino relativo). In questi casi la directory corrente (quella in cui ci si trova) è identificata dal simbolo . e la directory precedente può essere individuata dal simbolo ... così come l'utilizzo del simbolo ~ può sostituire il path della propria home: per esempio per l'utente *tux* il simbolo ~ può essere usato al posto di /home/tux.

Comando	Significato ed esempi d'uso				
pwd	Print Working Directory. Mostra la directory in cui ci si trova (directory corrente)				
mkdir	MaKe DIRectory. Crea una sotto directory nella directory dove ci si trova:				
	 mkdir /home/tux/prova (crea la directory prova nella /home/tux da qualsiasi punto del file system ci si trovi) mkdir prova (crea la directory prova nella directory in cui ci si trova) 				
cd	Change Directory. Permette di spostarsi nella directory specificata:				
	cd prova (ci si sposta in prova. Deve esistere nella directory corrente, altrimenti bisogna specificare il cammino)				
rmdir	ReMove DIRectory. Cancella la directory specificata se vuota. Se la directory non è vuota occorre, prima, cancellare tutti i file in essa contenuti:				
	rmdir prova (elimina la directory prova)				
less cat	Ambedue comandi utilizzati per visualizzare il contenuto di un file di testo. Utilizzando il primo comando si entra in un ambiente per la visualizzazione del testo. Se non può essere contenuto in una sola schermata, si può navigare con i tasti <pagsu>, <paggiu>. Si chiude la visualizzazione digitando il tasto Q. Utilizzando il secondo comando, invece, il testo viene mostrato interamente a video e si ritorna al prompt:</paggiu></pagsu>				
	 less prova.txt (mostra su video il contenuto del file permettendo la navigazione al suo interno) cat prova.txt (mostra il file a video e ritorna al prompt). 				
ср	CoPy. Copia uno o più file da una posizione ad un altra:				
	 cp ~/prova1/linux.png /home/tux/prova2/ (copia il file linux.png, che si trova in /home/tux/prova1 nella directory /home/tux/prova2). Se l'utente ha identificativo tux, il carattere ~ sostituisce /home/tux. cp prova1/*.png prova2/ (ammesso che ci si trovi in /home/tux, copia tutti i file che terminano con .png e che si trovano in prova1, in prova2. Il carattere * sostituisce qualunque sequenza di caratteri) 				

Comando	Significato ed esempi d'uso
rm	ReMove. Cancella uno o più file:
	 rm linux.png (cancella il file specificato) rm *.png (cancella <u>tutti</u> i file che contengono .png come parte finale)
mv	MoVe. Sposta un file da una posizione ad un altra. Comando utilizzato anche per cambiare il nome ad un file:
	 mv prova1/linux.png prova2/ (sposta il file specificato dalla directory prova1 alla directory prova2)
	2. mv linux.png tux.png (cambia il nome del file da linux.png a tux.png)

Un comando importante per la gestione dei file è ls (LiSt).

- Is Il comando usato da solo mostra l'elenco dei file contenuti nella directory specificata. Se non viene specificata la directory, si intende quella all'interno della quale ci si trova.
- Is -a In questo esempio viene usata l'opzione -a (All). Le opzioni che modificano l'effetto dei comandi sono specificati facendoli precedere dal simbolo -. Nello stesso comando possono essere raggruppate più opzioni, basta specificarle, una dopo l'altra, dopo il simbolo. L'opzione -a permette di includere, nell'elenco dei file visualizzati, anche quelli cosiddetti nascosti (sono i file il cui nome comincia con il punto). Si tratta di file di sistema utilizzati dai programmi per conservare le proprie impostazioni e, in generale, di file di testo visualizzabili (vedi l'esempio proposto nel comando 1ess).
- → 1s -1 In questo modo si ottiene una lista *lunga* dei file contenuti nella directory:

```
$ 1s -1
totale 683800
                         4096 2010-03-29 23:04 backup
drwxr-xr-x 3 tux tux
drwxr-xr-x 13 tux tux
                         4096 2010-01-30 21:58 Biblioteca
                        4096 2010-07-10 13:42 Musica
drwxr-xr-x 37 tux tux
                          48 2010-09-14 09:32 documento.1
-rw----- 1 tux tux
                         4096 2010-07-12 12:52 ogg
drwxr-xr-x 4 tux tux
                         4096 2010-07-26 17:02 PDF
drwxr--r-- 2 tux tux
-rw-r---- 1 tux tux
                         48 2010-09-14 09:32 prova.txt
drwxr-xr-x 7 tux tux
                         4096 2010-09-14 09:32 public_html
```

nella lista visualizzata è possibile leggere, nell'ordine, i permessi sul file, nome del proprietario e del gruppo cui questi appartiene, dimensioni in byte, data ed orario ultima modifica, nome.

Per quanto riguarda i diritti sul file, sono previsti 10 caratteri che possono essere:

-	Indica assenza del diritto
d	Specificato solo come primo carattere: indica se il file è una directory
r	Read: diritto di lettura sul file
W	Write: diritto di scrittura, modifica del file
х	eXecute: diritto di esecuzione

A parte il primo carattere che indica se si tratta di un file semplice o di una directory, il resto va letto a gruppi di 3 nella forma rwx: il primo gruppo da sinistra indica i diritti del proprietario, il secondo quelli del gruppo, il terzo quelli degli altri. Per esempio la stringa (divisa, per comodità di lettura, nelle sue componenti) – rwx rw- r--, indica che si tratta di un file normale, che il proprietario ha tutti i diritti, che il gruppo ha diritti di lettura e scrittura e che gli altri hanno il solo diritto di lettura.

Modificare i permessi con comando da terminale

Il proprietario di un file può modificarne i permessi usando chmod. Il comando accetta il tipo di permesso specificando rwx, eventualmente raggruppati, preceduto dal simbolo + o - a seconda se si vuole attribuire o negare il permesso. Il destinatario del permesso è espresso nella forma:

u	Indica il proprietario (user)
g	Indica il gruppo cui fa parte il proprietario (group)
0	Indica tutti gli altri non facenti parte del gruppo (others)
a	All: il permesso si applica a tutti

I destinatari del permesso possono essere raggruppati:

```
$ ls -l prova.txt
-rw-r---- 1 tux tux 48 2010-09-14 09:32 prova.txt
$ chmod g+w,o+r prova.txt
$ ls -l prova.txt
-rw-rw-r-- 1 tux tux 48 2010-09-14 09:32 prova.txt
```

il file prova.txt, in un primo momento, può essere letto e modificato dal proprietario, soltanto letto dal gruppo mentre gli altri non hanno alcun diritto sul file.

Il comando chmod attribuisce il diritto di scrittura per il gruppo e il diritto di lettura per gli altri così come evidenziato dal successivo 1s.

Nell'esempio con un unico comando si attribuiscono diritti diversi al gruppo e agli altri separandoli con la virgola.

```
$ chmod g-w prova.txt
$ ls -l prova.txt
-rw-r--r-- 1 tux tux 48 2010-09-14 09:32 prova.txt
```

Nell'esempio precedente, invece si toglie al gruppo il diritto di modifica sul file.

Ricerca di file

Un file che contiene un programma che si vuole avviare si può lanciare direttamente specificandone il nome solo se il file è registrato in una delle directory contenute nella variabile di ambiente PATH del sistema:

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/usr/games
```

Il comando echo visualizza sullo schermo il contenuto di una variabile ambiente (PATH nell'esempio). Nel caso di esempio specificare un file solo con il nome è sufficiente se il file del comando, per ipotesi, si dovesse trovare in /bin o qualsiasi altra directory compresa nell'elenco. In tutti gli altri casi è necessario specificare l'intero identificativo comprendente la path e il nome.

\$ which chmod
/bin/chmod

\$ which prova.txt
\$

Il comando which visualizza il nome, completo di path, del file, ma, come evidenziato dall'esempio, funziona solo con i comandi registrati nelle directory contenute nell'elenco della PATH.

Uno dei modi per poter cercare la posizione, all'interno di una memoria di massa, di un file qualsiasi è quello di utilizzare la coppia di comandi updatedb e locate. Il primo effettua una scansione della memoria di massa e crea, o aggiorna se esiste già, un database con l'elenco del contenuto del supporto. È necessario effettuare una nuova scansione solo nel caso in cui il contenuto della memoria di massa sia stato modificato in modo da sincronizzarlo con le informazioni contenute nel database. Il comando locate cerca nel database il file richiesto e ne mostra il nome completo.

```
$ sudo updatedb
$ locate prova.txt
/home/utente1/prova.txt
/home/tux/varie/prova.txt
```

Il primo comando sincronizza il contenuto del database con quello della memoria di massa. Il comando è preceduto da sudo che permette di far eseguire il comando con i permessi del super utente. Ciò è necessario per poter accedere a directory non accessibili agli utenti con normali diritti e poter avere un elenco completo del contenuto della memoria di massa.

Il secondo comando effettua la ricerca all'interno del database e restituisce, nell'esempio, informazioni sull'esistenza di due file con quel nome.

File e collegamenti simbolici

I *collegamenti simbolici* (o symbolic link chiamati così per distinguerli da un altro tipo di collegamenti, utilizzati da Linux, non trattati in questi appunti) sono file speciali che puntano ad altri file in modo che, agendo sul collegamento, si interviene sul file collegato. In questo modo si può creare in una directory un collegamento ad un file, conservato in un altra directory, e operare sul collegamento, elaborando, di fatto, il file collegato.

Rispetto alla copia di un file, il collegamento offre dei vantaggi:

- La copia duplica il file occupando spazio su disco, tanto quanto ne richiede la dimensione del file che si sta copiando
- Il collegamento, prescindendo dalle dimensioni del file collegato, occupa solo lo spazio, che è sempre lo stesso, utile per conservare le informazioni per il rintracciamento del collegato.

Utilizzando collegamenti può essere gestita, per esempio, la condivisione di dati fra più utenti. I collegamenti registrano il cammino che porta al file collegato e, in questo modo, il sistema può agire sul collegamento allo stesso modo del file collegato. Se si cancella il collegamento, naturalmente, il file collegato non viene interessato dall'operazione.

Si genera un collegamento per mezzo del comando ln:

```
$ ls -l prova*
-rw-r--r- 1 tux tux 19617 giu 18 1997 prova.cpp
$ mkdir seconda
$ cd seconda
```

\$ ln -s ../prova.cpp nuovo.cpp

Nella directory corrente si trova il file prova.cpp. Il secondo comando crea una nuova directory nella corrente e, con il terzo comando, ci si sposta dentro la directory creata. Qui si crea un link simbolico chiamato nuovo.cpp al file prova.cpp che si trova nella directory padre.

La visualizzazione delle proprietà di nuovo.cpp, mostra alcune caratteristiche:

```
$ 1s -1
lrwxrwxrwx 1 tux tux 12 mar 26 09:28 nuovo.cpp -> ../prova.cpp
```

- 🔶 La stringa dei diritti comincia con 1 ad indicare che si tratta, appunto, di un link
- 🔶 La stringa dei diritti di nuovo.cpp, in realtà, non esiste: valgono i diritti del file puntato
- È evidenziato, per mezzo dell'uso dell'operatore ->, che nuovo.cpp fa riferimento a prova.cpp.

Utilizzo dell'editor nano

Gli editor di testo sono software appartenenti alla categoria delle utility presenti *a corredo* di qualsiasi sistema operativo. Si pone spesso, infatti, la necessità di modificare file di testo (file di configurazioni). In ambiente Linux esistono moltissimi editor con caratteristiche e prestazioni diverse da utilizzare in relazioni alle diverse esigenze: scrivere programmi, editare semplici file di configurazione. In questi appunti si tratterà l'uso generale dell'editor **nano**.



Si tratta di un editor molto leggero e generalmente già installato nelle distribuzioni, ma, a dispetto delle dimensioni minime, dotato di caratteristiche avanzate. Si avvia digitando nano dal prompt. Lo schermo comprende una prima riga con il nome del programma, del file che si sta editando e indicazioni se tale file risulta modificato, l'area di lavoro che si espande su quasi tutto lo schermo e che è la zona sulla quale verrà editato il file, una riga di stato (terza riga dal basso) che visualizza messaggi dal programma, due righe con le abbreviazioni dei comandi più utilizzati.

Nano accetta comandi utilizzando le combinazioni di tasti: tasto <CTRL> (indicato ^) o tasto <ESC> (indicato con M) e un tasto alfanumerico che, in genere, ricorda l'operazione che si vuole effettuare. Per esempio la combinazione ^x, che provoca l'uscita dal programma (eXit), si ottiene

premendo il tasto <CTRL> e il tasto X.

Il testo si inserisce utilizzando i normali tasti da tastiera e sono abilitati i tasti freccia o <PagSU>/<PagGIU> per gli spostamenti all'interno del testo.

Comando	Effetto					
^X	Esce dal programma. Se il file editato è stato modificato viene chiesto di salvarlo					
^G	Mostra una pagina di aiuto con l'elenco dei comandi accettati da nano					
M-6	Copia la riga su è posizionato il cursore in una zona di memoria chiamata cutbuffer , in modo da permettere operazioni di Copia/Incolla					
^K	Elimina la riga su cui è posizionato il cursore e la conserva nel cutbuffer					
^U	Copia a partire dalla posizione del cursore il testo presente nel cutbuffer					
^C	Mostra nella riga di stato la posizione corrente del cursore (numero di riga e colonna)					
M-G	Chiede, con un messaggio nella riga di stato, una posizione (riga e colonna) su cui spostare il cursore					
^W	Chiede, con un messaggio nella riga di stato, un testo da cercare e posiziona il cursore sulla prima ricorrenza del testo cercato a partire dalla posizione attuale del cursore					
M-W	Ripete la ricerca precedentemente impostata e sposta il cursore sulla prossima ricorrenza del testo					

La combinazione ^R permette di inserire un nuovo file nella posizione attuale del cursore:

Fi	le da inseri	re [da	./]	:				
^G	Guida			^	Sfoglia	M - F	Nuovo	Buffer
^C	Annulla			^	Esegui comando			

Il comando inserisci è molto potente:

- → scrivendone il nome, permette di inserire il file nella posizione del cursore
- → può caricare il file in un nuovo buffer per permettere di editare più file (M-F)
- non è necessario conoscere il nome del file da inserire, si può navigare (^T) fra le directory e scegliere nell'elenco dei file mostrati
- ➡ si può eseguire un comando di shell (^x) e avere nel buffer l'output del comando

Quando sono caricati in memoria, in più buffer, più di un file, nell'area di lavoro ne può essere visualizzato soltanto uno ma si può commutare fra i file presenti utilizzando M-< e M->.

Links: navigazione web con browser testuale

Anche dal terminale, da un computer su cui non è installato un server grafico, è possibile utilizzare browser per la navigazione in Internet. D'altra parte, in origine, i primi browser per la visualizzazione e la navigazione nel web erano testuali: i sistemi grafici erano molto costosi e utilizzati da pochissime persone. Anche oggi nei sistemi senza grafica (principalmente i computer destinati a far girare programmi server) è necessario utilizzare browser non grafici.

In questi appunti si tratterà del programma links/links2. Non si tratta di due programmi distinti ma il secondo (links2) è links compilato con le opzioni per la grafica e quindi si può utilizzare alla stessa stregua di altri browser più conosciuti.



Il programma può essere avviato digitandone il nome da solo o facendolo seguire dall'URL della pagina che si vuole consultare:

```
$ links http://ennebi.solira.org
```

La pressione del tasto $\langle ESC \rangle$ abilita la visualizzazione, nella parte superiore, della riga dei menù che permettono le operazioni comuni presenti negli altri browser. Per esempio scegliendo *Apri URL* dal menù *File* o, direttamente, mentre è visualizzata la pagina premendo il tasto g, si può inserire l'indirizzo della pagina che si vuole visualizzata.

La pagina visualizzata può essere fatta scorrere utilizzando i tasti <PagGIU> e <PagSU>. I tasti <SU> e <GIU> fanno passare al successivo/precedente link. I tasti <DEX>, <SIN> o <INVIO> selezionano il link e caricano la pagina relativa per la visualizzazione.

Il programma può essere utilizzato in modalità grafica, a condizione che il sistema sia in grado di visualizzare schermate grafiche, aggiungendo una opportuna opzione.

```
$ links2 -g http://ennebi.solira.org
```

Aggiornare il sistema con apt-get

Come già accennato apt-get è un sistema che permette di mantenere aggiornato il software presente nella macchina accedendo ai repository on-line.

Il sistema, di cui si sono trattati in paragrafi precedenti due front-end, si basa fondamentalmente su alcuni file di testo e su un programma che si occupa di effettuare il collegamento ai repository, scaricare i pacchetti necessari e installarli.

Il file /etc/apt/sources.list è un file di testo, il cui contenuto può quindi essere visualizzato per esempio con less, con l'elenco dei repository da cui prelevare il software. Nel file inizialmente sono già presenti i repository ufficiali della distribuzione.

La directory /var/lib/apt/list/ contiene per ogni repository un file di testo con le informazioni sui pacchetti presenti nel repository. Si tratta in pratica delle informazioni che sono visualizzate nelle varie aree della finestra di Synaptic: nome del pacchetto, descrizione ...

Il comando apt-get si occupa praticamente della gestione delle installazioni: basta specificare l'operazione da effettuare (install, update, upgrade o remove) e il nome del pacchetto. Es:

\$ sudo apt-get install libreoffice

lancia l'installazione del pacchetto LibreOffice. Naturalmente l'installazione di nuovi programmi richiede i privilegi di *root* ed è per tale motivo che il comando è completato da sudo.

Altri parametri che possono essere utili:

- update: ricarica le liste dei programmi dai repository in modo da sincronizzare la quantità e le versioni dei programmi visualizzati nel computer e quelle presenti nei repository
- + upgrade: aggiorna un pacchetto alla nuova versione disponibile
- → remove: disinstalla un'applicazione.

