

Giornale di Sistema

APPUNTI ED IMMAGINI DAI MIEI VIAGGI NEL MONDO DEI BIT ED I SUOI DINTORNI

(<http://giornaledisistema.blogspot.com>)

Appunti su Linux

Raccolti a cura di Michele Benvegnù
(<http://www.benvegnu.it>)

Aggiornato al 18 novembre 2007

sabato 10 novembre 2007

Linux: che dite, approfondiamo?

Della serie: riflessioni sotto la doccia... avrete notato che in questi giorni non ho messo online nuovi post. Mi sono preso un attimo di pausa per staccare un pò da tutto, blog compreso, e lasciare lo spazio perché qualcosa di nuovo emergesse.

In fondo questo blog, a dispetto dei suoi stessi post che sono in gran parte dedicati alla virtualizzazione ed a VirtualBox in particolare, non ha un argomento prefissato, se non quanto compreso nel sottotitolo del blog stesso: "*Appunti ed immagini dai miei viaggi nel mondo dei bit ed i suoi dintorni*". E appellandomi a quel "*ed i suoi dintorni*", ogni tanto la faccio fuori dal vaso postando anche qualche scritto un pò più... più... come dire, "intimista".

Come diario invece del mio viaggio nel mondo dei bit, il blog non può che riflettere quanto mi accade lungo il cammino, quanto scopro, quanto imparo. E se all'inizio scrivevo per me stesso, per dare una forma ad una passione che mi prendeva dentro ma non trovava poi nessuna manifestazione, nessuna espressione, le cose nel corso di pochi mesi sono cambiate.

Lungo il cammino ho trovato altri che come me viaggiano, esplorano, scoprono, e curiosi si pongono domande, e curiosi cercano risposte: ho incontrato voi. :-D

A volte vi incontro nella tranquillità dei commenti ai post, altre volte nelle più fredde statistiche di consultazione dei post, altre volte nelle mail che arrivano all'indirizzo di posta dedicato al blog (lo trovate in fondo alla barra laterale sinistra). E quale che sia la forma, sono felice di sapervi in cammino insieme a me, anche solo per un brevissimo tragitto, di sapere che altri viaggiatori viaggiano, che passano di qua, che spesso trovano quello che cercano, che altre volte invece ripartono, con una direzione, uno spunto che prima non avevano.

Ho cercato perciò di scrivere sempre in modo semplice ed esaustivo (ci sarò riuscito?), non dando mai nulla per scontato, perché tutti potessero trovarvi informazioni comprensibili, alla mano. Niente cose da super guru, il primo dei neofiti sono io. E rileggendo gli ultimi post, mi sono reso conto però che stanno diventando sempre più un atto di fede.

Per far funzionare bene Linux con VirtualBox infatti (ad esempio per la condivisione cartelle oppure la compressione del disco rigido virtuale), molto spesso ci siamo addentrati insieme nei meandri di Linux. OK, la cartina era buona, il viaggio spero sia stato comodo e senza troppi intoppi. Ma ogni post assomiglia sempre di più ad una ricetta: fai così, fai così e ottieni questo risultato. Con una differenza sostanziale: se nella ricetta vi dico "ora buttate gli spaghetti nell'acqua bollente", posso tranquillamente omettere di dirvi che vanno prima tolti dalla busta, con Linux no.

In ogni ricetta si danno per scontate un sacco di cose che fanno semplicemente già parte del bagaglio comune di conoscenze. Con Linux questo bagaglio di conoscenze pregresse manca, siamo pur sempre dei neofiti che arrivano da Windows, e mi chiedo come sia per chi viene da MacOS... se mai esista! ;-)

Diceva Michele Benvegnù: "*Ci sono troppe differenze tra le varie distribuzioni linux, ci si perde la testa*".

Rispondeva: "*Si, in effetti Linux è davvero trasformista per chi come noi è abituato a Windows. Questo perché (usando una metafora), mentre Windows è stato costruito in un unico blocco di marmo, Linux è stato costruito come l'unione di tanti mattoncini Lego.*

Risultato: ciò che in Windows ha un solo ed unico aspetto, in Linux può assumere tanti aspetti diversi. Abituati noi alle cose con unico aspetto, ci lasciamo abbagliare e confondere dalle apparenze. L'unico modo per superare la difficoltà è dare un'occhiata da vicino a quei mattoncini, fuor di metafora, agli elementi base di Linux che si ripetono seppur con varie sfumature in ogni distribuzione.

Compresi i mattoncini ed il modo di metterli insieme, le sfumature non saranno più una rottura di balle, anzi, diverranno il punto di forza di Linux. E ognuno andrà a cercarsi la distribuzione con la sfumatura che più gli si addice."

Ecco perché, per tutto quanto vi ho detto finora, penso sia arrivato il momento di iniziare un nuovo viaggio, di addentrarci, questa volta senza ricette precotte, in un nuovo campo: Linux. Vorrei accompagnarvi nei suoi meandri, farvelo vedere da vicino, farvi vedere il suo cuore, farvelo toccare mentre pulsa pompando vita in un modo nuovo di intendere il mondo, di intendere la vita.

Che dite, cominciamo? :-D

@:\>

Publicato alle 6.50 Argomenti: amministrazione di sistema, Linux, sotto la doccia

lunedì 12 novembre 2007

Linux: iniziamo

Nello scorso post, "[Linux: che dite, approfondiamo?](#)", vi avevo parlato delle motivazioni che mi hanno portato alla decisione di darci dentro con Linux. Niente più atti di fede, niente più ricette precotte, è arrivato il momento di guardare Linux in modo più organico, più organizzato, cominciando a dare un senso a cose che sembrano spuntare dal nulla, a comandi che sembrano parole dall'effetto magico.

In realtà avevo già scritto un paio di post in proposito. Non so se li abbiate letti, ma sono la base da cui partiremo per costruire un percorso di apprendimento. Ho deciso perciò di spostarli ad oggi, così che possiate leggerli, o semplicemente farne un breve ripasso. La curva di apprendimento di Linux non è proprio tra le più agevoli, anzi, a dirla tutta, è abbastanza ripida. Ci metterò tutto l'impegno a semplificarvi il più possibile la cosa, aiutatemi mettendoci tutta la pazienza necessaria.

E statene certi, ne varrà la pena. :-)

@:\>

Pubblicato alle [9.26](#) Argomenti: [amministrazione di sistema](#), [guida](#), [Linux](#), [tutorial](#)

lunedì 12 novembre 2007

Amministrazione di sistemi Linux: premessa

Una doverosa premessa. Scordatevi sin d'ora la vostra interfaccia grafica preferita per dare i comandi: non cliccheremo sul file per cancellarlo, è una cosa intuitiva che chiunque venga da Windows riuscirà a fare benissimo anche su Linux, daremo invece il relativo comando *rm* (remove) da shell (l'interfaccia testuale di Linux, tipo finestra del DOS su Windows ma enormemente più potente). Per quanto possa sembrare astruso usare un'interfaccia testuale invece di una comodissima Gnome o KDE, tutto ciò ha perfettamente senso. Vediamone i motivi principali:

- **non sempre è disponibile un'interfaccia grafica:** soprattutto se il sistema è in panne, ma anche perché state lavorando su una macchina vecchia che non c'è la fa a far girare un'interfaccia grafica, oppure perché siete su un server e preferite risparmiare le pesanti risorse che un'interfaccia grafica consuma, potreste trovarvi ad avere a disposizione solo la riga di comando, sempre presente in ogni sistema.
- **le interfacce grafiche hanno bisogno continuo di interazione:** occorre stare lì continuamente a cliccare su qualcosa per scegliere opzioni e dare comandi. Immaginate di dover rinominare un migliaio di file: dovete cliccare sul file, scegliere in qualche modo l'opzione di rinomina del file, scrivere il nome del file, renderlo effettivo; ricominciare d'accapo tutto ciò e ripeterlo per tutti i mille file. Beh, potete spendere meglio il vostro tempo: lanciate uno *script* (una sequenza di comandi registrati su un file, come i file *.bat* di Windows per capirci) e nel frattempo vi bevete un buon caffè (se la vostra macchina è lenta nello svolgere l'operazione).
- **le interfacce grafiche non rendono sempre disponibili all'utente tutti i comandi realmente presenti:** molto spesso i comandi e le opzioni che l'interfaccia grafica rende disponibili sono solo una parte di tutti quelli disponibili, per poter usare gli altri bisogna scriverli dalla riga di comando.
- **l'uso della riga di comando è più elastico di quello dell'interfaccia grafica:** tramite una sequenza di opportuni comandi (i già citati script), è possibile creare nuove operazioni completamente automatizzate. In realtà ciò sarebbe possibile anche dall'interfaccia grafica usando uno degli appositi programmi che registrano la nostra attività sullo schermo (pulsanti cliccati, opzioni scelte), per poi ripeterla come un unico comando. Dato che gli script sono alla fine una vera e propria forma di programmazione, possiamo però prevedere, a seconda della situazione, l'esecuzione di alcuni comandi o di altri, rendendo il tutto molto più adattabile alle situazioni che si presentano "in corso d'opera" cioè durante l'esecuzione stessa dello script.
- **l'interfaccia grafica nasconde molte delle cose che succedono nel dietro le quinte:** nella metafora del teatro, se voi siete lo spettatore, cioè l'utente normale che vuole: avviare Linux, scrivere il suo documento con OpenOffice.org, spedirlo come allegato via e-mail, allora l'interfaccia grafica va benissimo. Ma se siete l'attore, cioè l'amministratore del sistema, allora avete bisogno di sapere anche cosa accade davvero nel dietro le quinte e sapervi muovere in esso. Poi, quando avrete imparato, saprete muovervi sia sul palco che dietro il palco, e saprete quando è il momento di essere davanti e quando dietro il palco stesso.

Fuor di metafora: quando saprete bene cosa accade sul vostro sistema, saprete anche come non complicarsi la vita usando di volta in volta, l'interfaccia grafica o quella a riga di comando a seconda delle esigenze, ma sapendo bene, nell'uno e nell'altro caso, cosa avviene davvero e come mantenere il controllo totale del sistema a voi affidato.

Il materiale sarà valido per Linux in tutte le sue incarnazione; pertanto, se avete voglia di capire davvero cosa c'è sotto il cofano della vostra distribuzione preferita, non vi resta che seguire [i prossimi post sull'argomento](#).

@:\>

Publicato alle 9.41 Argomenti: [amministrazione di sistema](#), [guida](#), [Linux](#), [tutorial](#)

lunedì 12 novembre 2007

Linux: architettura di base

Che lo abbiate installato su una macchina virtuale di VirtualBox oppure direttamente sul vostro PC, da un liveCD oppure da Windows usando Wubi, è arrivato il momento di guardare da vicino come funziona un sistema operativo Linux.

Linux è un sistema operativo *multitasking e multiutente*, ossia può gestire più processi e più utenti contemporaneamente (un *processo* è l'unità minima di elaborazione di un programma).

Il cuore di un sistema operativo è il *kernel*. Quello di Linux si occupa solamente di una cosa: eseguire processi sfruttando le risorse hardware della macchina su cui gira. Tutto il resto, dai servizi di sistema all'interfaccia grafica, è ottenuto facendo girare un programma. Il kernel di per se non offre nessuno di questi servizi: non offre ad esempio funzionalità per copiare i file, non offre un'interfaccia utente, né di testo né di tipo grafico.

Quando usate un'interfaccia grafica come Gnome o KDE, queste non sono funzionalità offerte dal kernel, bensì normalissimi programmi esterni al kernel, al pari di un OpenOffice.org o Firefox. Ecco perché in Linux potete scegliere più interfacce grafiche, è semplicemente un programma come tutti gli altri. In Windows invece l'interfaccia grafica è integrata nel sistema operativo, ed in alcune versioni, anche il browser per la navigazione Internet.

Quando diciamo Linux poi, non stiamo riferendoci all'intero sistema operativo, ma solo al kernel. Sfruttando i servizi che questi mette a disposizione, una moltitudine di programmi si occupa di fornire tutte le funzionalità che ci si aspetta da un moderno sistema operativo. Questi programmi arrivano direttamente, o derivano in qualche modo, dal progetto GNU della *Free Software Foundation*: ecco perché in generale si parla di sistema operativo *GNU/Linux*.

Infine, quella che comunemente viene chiamata *distribuzione*, Debian, Ubuntu, openSUSE, RedHat solo per fare qualche nome fra le più conosciute, è l'insieme del sistema operativo GNU/Linux più una raccolta coerente di programmi usati dagli utenti per le loro normali attività: programmi di grafica, produttività personale, navigazione Internet, player audio/video e così via.

Da qui in poi però, per semplificarci la vita, un sistema operativo GNU/Linux continueremo a chiamarlo con il solo termine Linux.

Ok, accordatici sulla terminologia da usare, torniamo ora ad occuparci del kernel vedendo quali sono le poche ma fondamentali mansioni che svolge per darci l'unico servizio che offre: *eseguire processi*, l'unità minima di elaborazione di un programma. Queste mansioni sono svolte dalle tre sottosezioni del KERNEL visibili in **fig. 1**:

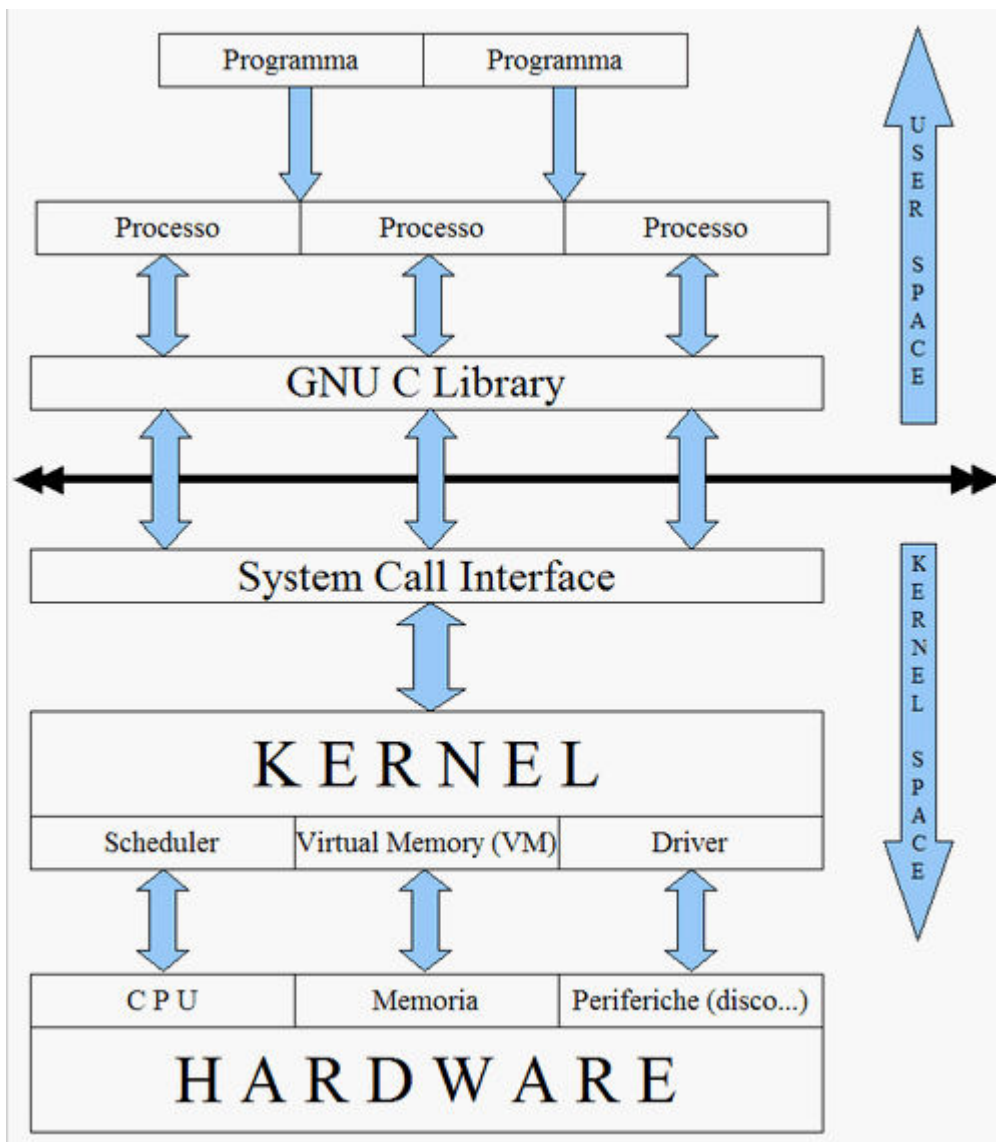


fig. 1

- gestione dei processi tramite lo *Scheduler*.

Questa parte del kernel si occupa di tutto ciò che riguarda la gestione di un processo, ad esempio di stabilire l'ordine di esecuzione dei processi, l'esecuzione stessa, la sospensione dell'elaborazione, la ripresa, lo stop e così via.

- gestione della memoria tramite il *Virtual Memory (VM)*.

La memoria fisica non è liberamente accessibile ai processi. È il VM che, anche con l'aiuto del Memory Management Unit (MMU), apposita unità presente direttamente nell'hardware dei microprocessori, rimappa opportunamente quella fisica in uno spazio virtuale che rende disponibile al processo. In questo modo, ogni processo non accede mai alla memoria reale del sistema, ma solo a quella resagli disponibile dal VM.

Sigillato all'interno del suo spazio di memoria, il processo non può creare problemi accedendo per sbaglio allo spazio di un altro processo (chi ha usato le prime versioni di Windows, sa quanto fossero frequenti i blocchi dell'intero sistema causati proprio da problemi di questo tipo). Quando la memoria RAM poi per qualche motivo diventa insufficiente, è sempre il VM a gestire lo swap, lo spostamento cioè di parti della memoria sul disco fisso per liberare spazio.

- gestione delle periferiche tramite i *Driver*.

Questa indicazione generica è relativa a quella parte del kernel che si occupa di accedere alle periferiche rendendole disponibili ai programmi. La particolarità di questa sezione è che fornisce un modalità comune (*un'interfaccia*) di accesso a periferiche anche molto diverse fra loro, porte seriali, dischi rigidi, scheda audio tanto per fare degli esempi, trattandole tutte come se fossero dei file; da qui uno dei motti di Linux: *tutto è un file*. Vedremo successivamente come ciò sia possibile e i vantaggi di questo approccio.

Ecco, questa è in estrema sintesi l'architettura di un sistema Linux, giusto per iniziare a capire; in realtà le cose sono ben più complesse, provate a dare un'occhiata a questa [mappa interattiva del kernel di Linux](#) e vi renderete conto.

@:|>

PUBBLICATO ALLE 9.49 ARGOMENTI: AMMINISTRAZIONE DI SISTEMA, GUIDA, LINUX,
TUTORIAL

martedì 13 novembre 2007

Linux: avvio e funzionamento del kernel

All'accensione del PC, il primo programma ad essere eseguito è contenuto direttamente nel firmware della macchina: il *BIOS* (Basic Input/Output System). Questo programma effettua tutta una serie di controlli sull'hardware che se conclusi correttamente portano alla fase successiva: l'avvio, dal dispositivo indicato nelle impostazioni del bios, del *bootloader*. L'unico compito del *bootloader* è cercare un'immagine del kernel, caricarla in memoria e avviarla.

Partito il kernel, questi prende il controllo della macchina, effettua una serie di operazioni di inizializzazione come la scansione delle periferiche disponibili, e infine lancia il primo processo di un sistema Linux, *init*. In realtà questa è solo una convenzione sul nome del primo processo da lanciare; in sistemi embedded ad esempio, cioè quei sistemi adibiti ad un singolo compito quali un comune telefonino, in genere viene lanciato altro.

init, sfruttando la possibilità di un processo di lanciare a sua volta un altro processo (il primo è detto padre, il secondo figlio, *init* è quindi il padre di tutti i processi Linux), lancia altri programmi che a loro volta lancia ancora altri programmi, tutto questo fino a che sono partiti tutti i programmi che servono per il corretto funzionamento dell'intero sistema operativo.

Ma come funziona l'intero sistema operativo basandosi su un kernel che può fare solo le poche cose già viste nel post "[Linux: architettura di base](#)"? Guardate **fig. 1** mentre vi spiego.

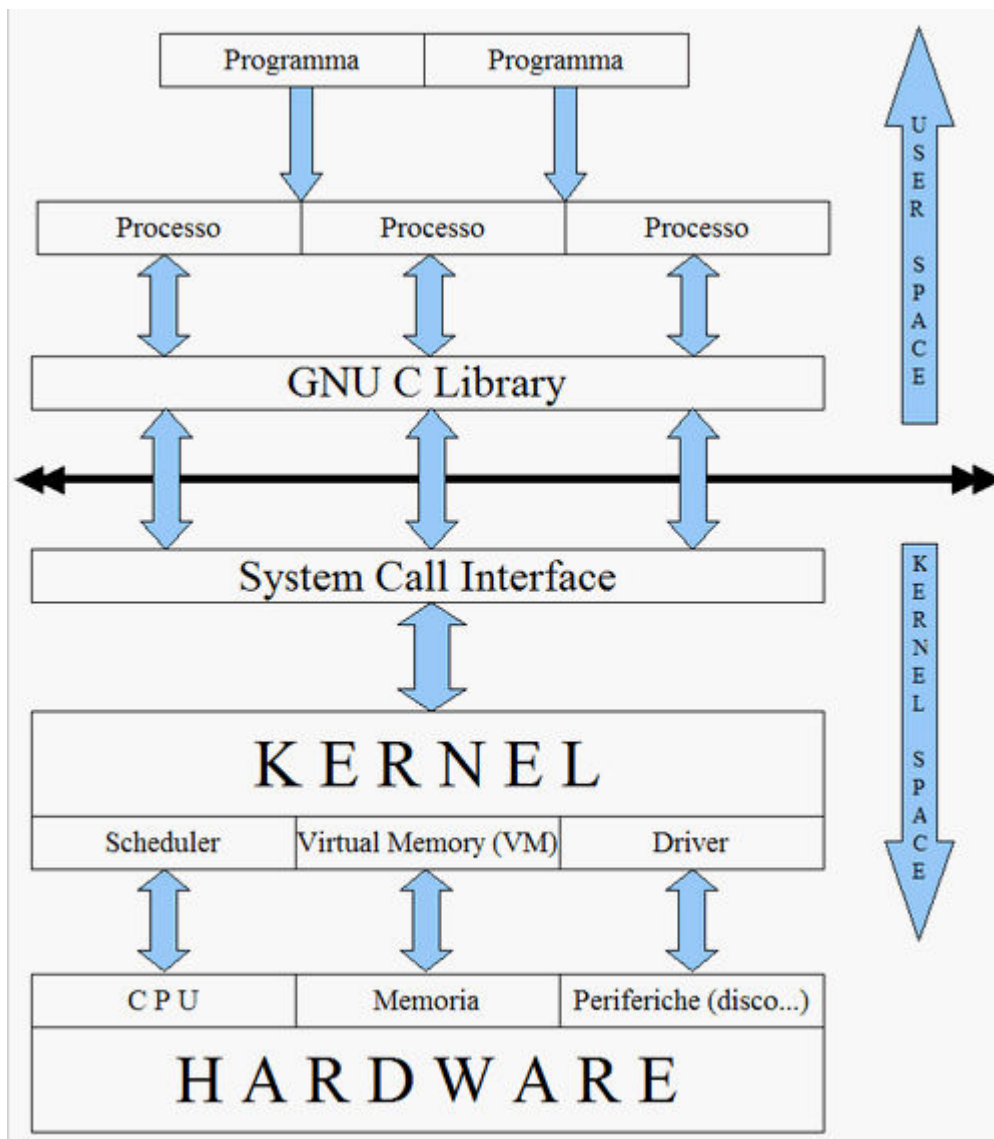


fig. 1

Abbiamo già accennato al fatto che il kernel è l'unico che può accedere direttamente all'hardware della macchina: grazie alle sue tre sezioni, scheduler, virtual memory, driver sfrutta la potenza e le risorse della macchina e le rende disponibili ai processi.

Quando un programma (e quindi uno dei processi da cui è costituito) ha bisogno per esempio di altra memoria, la chiede ad un apposito programma contenuto nel sottostante livello chiamato *GNU C Library* (guardate **fig. 1**), una serie di programmi scritti in linguaggio C. Questo a sua volta, invoca un'apposita funzione del sottostante livello chiamato *System Call Interface* (Interfaccia alle Chiamate di Sistema). Queste sono funzioni rese disponibili dal kernel per poterlo informare che si ha bisogno di qualcosa da lui. Ora il kernel sa che un processo ha bisogno di più memoria, e tramite l'apposita sezione del Virtual Memory provvede a fornirgliela.

Si può notare una netta separazione fra lo spazio in cui si trovano a girare i programmi, detto *User Space* (Spazio Utente), da cui non vi è accesso diretto all'hardware, e lo spazio in cui gira il kernel, detto *Kernel Space*, da cui invece si accede direttamente all'hardware; ecco perché solo il kernel può accedere direttamente all'hardware, mentre i processi girando in user space ne sono impossibilitati.

Il compito del kernel è unicamente quello di rendere disponibili le risorse hardware ai processi facendo da interfaccia fra questi due, di suo non effettua nessun'altra operazione. E' questo il motivo per cui in Linux tutto è realizzato tramite un programma: dalla semplice copia di un file all'intero filesystem passando per l'interfaccia grafica, si tratta sempre di programmi eseguiti in User Space, il kernel in se non ha disponibile nessuna di queste funzionalità.

Lungi dell'essere un limite, questo è invece un grande vantaggio di Linux: è per questo motivo che ad esempio, potete usare più interfacce grafiche (due su tutte, Gnome e KDE) invece dell'unica che ad esempio si può utilizzare in Windows. Ed è sempre per questo motivo che, se in Windows potete usare il solo filesystem NTFS (oppure il FAT, ormai obsoleto ma ancora utile in certe situazioni), in Linux ne potete usare una buona decina tra cui anche NTFS.

Chi ha avuto modo di installare Ubuntu 7.10 direttamente sul PC, senza usare VirtualBox o altri virtualizzatori, si sarà accorto come la partizione di Windows gli compare direttamente sul desktop come fosse un normale disco rigido di Linux: potete navigarlo a vostro piacimento e fare qualunque operazione vogliate sui file che contiene nonostante si tratti di un filesystem Windows. Ecco, questa è la potenza dell'approccio "tutto è un programma" di Linux.

@:\>

PUBBLICATO ALLE 9.31 ARGOMENTI: AMMINISTRAZIONE DI SISTEMA, GUIDA, LINUX, TUTORIAL

mercoledì 14 novembre 2007

Linux: la shell, la vostra centrale di comando

Con il post "[Linux: avvio e funzionamento del kernel](#)" abbiamo concluso una prima serie di post intesi a comprendere le basi dell'architettura di un sistema Linux. I prossimi sono post di passaggio prima di addentrarci nella giungla dei comandi Linux: una vera foresta in cui soprattutto chi proviene da Windows o più in generale gli amanti dell'interfaccia grafica, potrebbero facilmente perdersi. E in questo non siamo per niente aiutati dal front-end, dall'interfaccia usata per dare i comandi: la shell.

La shell non è altro che un'interfaccia di tipo testo fra noi e Linux. Come tutto il resto anch'essa è un programma, ed infatti esiste più di una shell. Alcune sono caratterizzate da grande leggerezza e un fabbisogno limitato di risorse, altre da comandi molto potenti, altre ancora dall'aderenza agli standard Posix. Quella di default sulla maggioranza delle distribuzioni, tra cui anche le nostre due di riferimento, Mandriva e Ubuntu, è la *bash* (Bourne Again SHell).

Il nome ed il modo di avviare la bash cambia a seconda che stiate usando l'interfaccia grafica Gnome (quella di Ubuntu per esempio) o KDE (di default in Mandriva): nella prima si chiama *Terminale* e potete avviarla come in **fig. 1**,

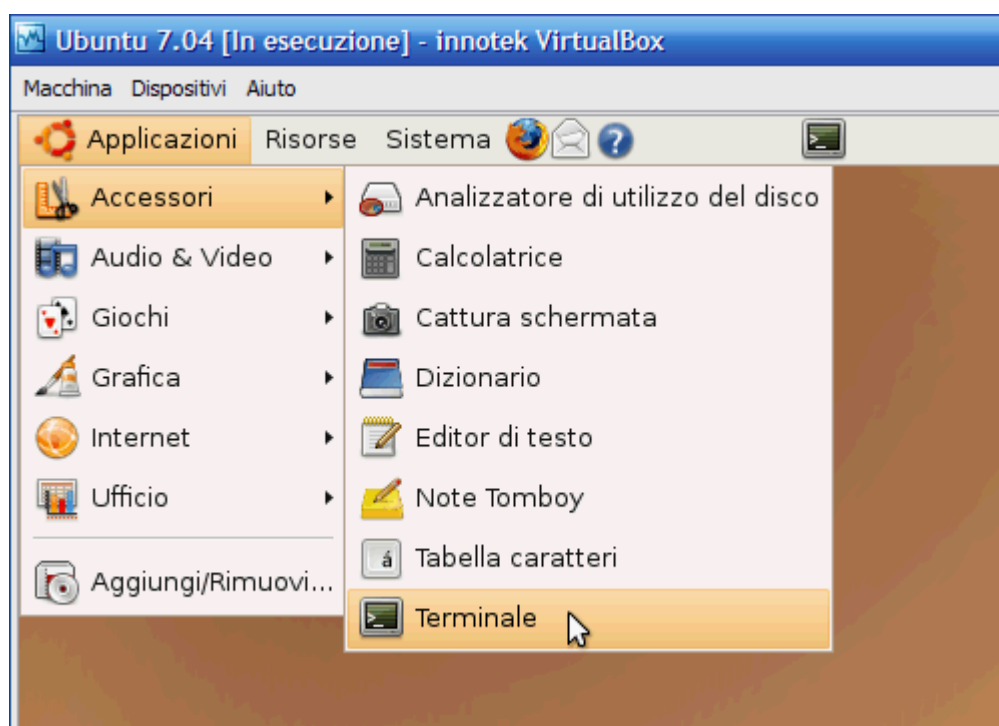


fig. 1

nella seconda si chiama *Konsole* (è consuetudine aggiungere una k iniziale nelle applicazioni KDE) e la potete avviare come in **fig. 2**.

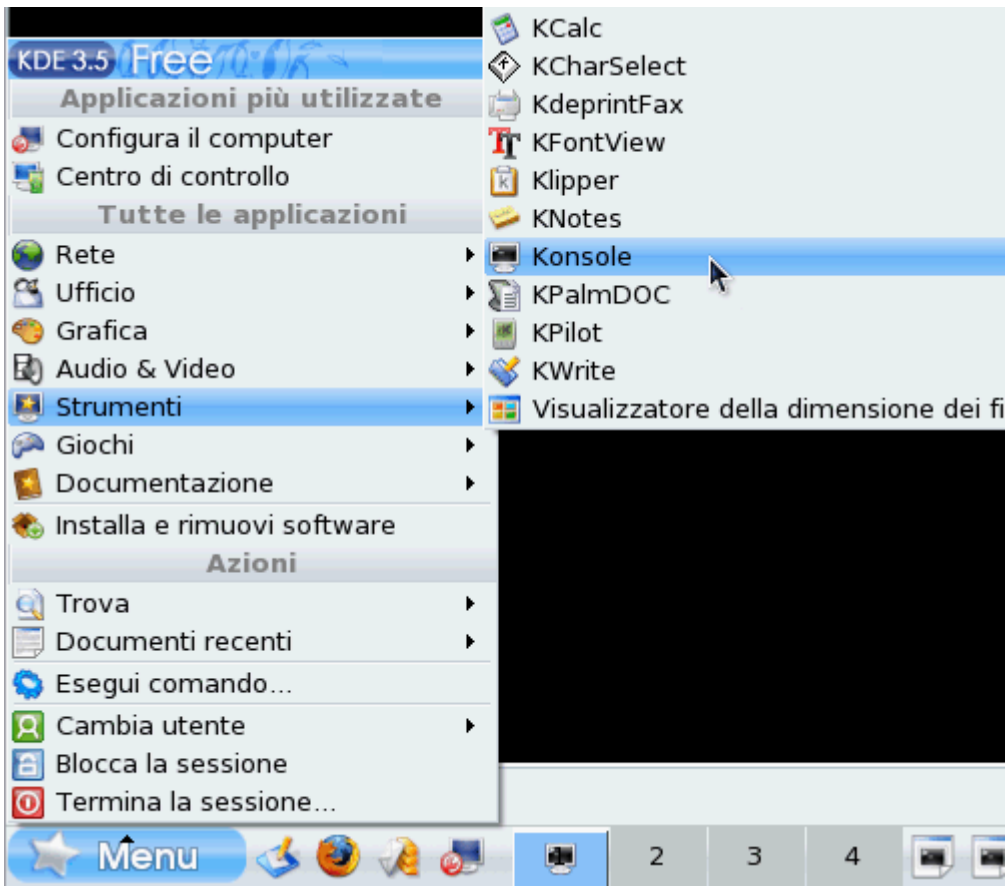


fig. 2

Una volta avviata, la shell mostrerà un prompt, un testo all'inizio di ogni riga, che contiene una serie di informazioni utili; vediamo quelle della bash in Ubuntu (fig. 3):

- la parte prima della "@" è il vostro nome utente (o userid);
- la parte successiva alla "@" che termina dove ci sono i "." è il nome con cui viene vista la vostra macchina all'interno di una rete;
- il simbolo "~" (tilde) per convenzione in Linux indica la vostra directory home. Dato che la home di un utente ha come nome quello dell'utente stesso, in fig. 3 allora ci troviamo nella directory /home/occhipervinca;
- il simbolo "\$" (dollaro) finale sta ad indicare che in questo momento siete un utente normale, se fosse il simbolo "#" invece siete l'amministratore; ricordatelo quando siete in dubbio se siete in modalità amministrativa o utente normale, nel primo caso dovete essere molto più attenti per la portata maggiore dei danni che potreste fare.

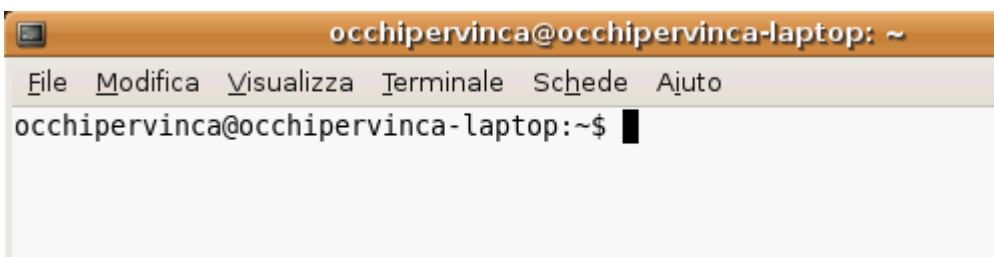


fig. 3

Due sono le funzionalità della bash che voglio mostrarvi, sono semplici ma di grande utilità; riguardano entrambi lo storico dei comandi precedentemente dati:

1) premendo il tasto "freccia su" della vostra tastiera potrete navigare all'indietro ed uno alla volta, tutto lo storico dei comandi già dati nella shell. Fate una prova, date i comandi:

- \$ ls che vi presenta la lista dei file della directory in cui vi trovate,
- \$ cd / per portarvi nella radice del vostro filesystem (è pressappoco come essere in C: in Windows),
- \$ ls -l, che mostra molte più informazioni del solo ls.

Vi ricordo che il simbolo dollaro "\$" del prompt sta ad indicare che siete in modalità utente normale. Ora premete "freccia su" e uno alla volta li vedrete ricomparire tutti in ordine inverso, dal comando più recente a quello più datato.

2) dando il comando hystory vedrete riportata tutta la lista dei comandi dati preceduti da un numero. Se volete eseguire un certo comando presente nella lista, è sufficiente dare !numero per vederlo eseguito.

Ecco, con solo questi due piccoli aiuti da parte della shell, il vostro lavoro diventerà molto più veloce e leggero, niente più riscrittura degli stessi noiosi comandi, soprattutto quando sono composti dal concatenamento di tanti singoli comandi che prossimamente vedremo.

@:\>

PUBBLICATO ALLE 8.08 ARGOMENTI: AMMINISTRAZIONE DI SISTEMA, GUIDA, LINUX, TUTORIAL

giovedì 15 novembre 2007

Linux: man, informazioni sui comandi

Abbiamo già visto nel [post precedente](#), come la shell non sia altro che un'interfaccia di tipo testo con cui dare comandi a Linux. Non clicchiamo su qualcosa, niente tasto destro o sinistro: solo testo, puro testo, nient'altro che testo. Ed una vera foresta di comandi mnemonici da mandar giù, con ancora più opzioni da poter usare per ognuno di essi, in cui soprattutto chi proviene da Windows o più in generale gli amanti dell'interfaccia grafica potrebbe/protebbero facilmente perdersi.

Prima di andare perciò dispersi in questa selva oscura, voglio condividere con voi altre informazioni di base per non perderci mai, o quanto meno ritrovarci nel caso "*che la diritta via era smarrita*". Linux infatti è molto ben documentato, e per ogni comando c'è la possibilità di averne immediatamente una descrizione e la lista della sue possibili opzione tramite un apposito comando: *man*.

Man è un'interfaccia ai manuali disponibili sul vostro sistema consultabili da shell. Prende in input i file contenenti questi manuali scritti in un linguaggio di formattazione chiamato Troff (come concetto analogo all'HTML nel senso che all'interno dello testo stesso vi sono anche le istruzioni di formattazione), e produce qualcosa di simile a quanto riportato più sotto.

Facciamo un esempio concreto: tramite il comando `man ls` visualizziamo le pagine del manuale del comando `ls` usato per avere la lista dei file contenuti in una directory. Aprite la shell (come fare lo trovate nel [post precedente](#)), e date il comando

```
$ man ls
```

(il `$` rappresenta il prompt, anch'esso spiegato nel [post precedente](#)), otterrete quanto segue:

```
LS(1) User Commands LS(1)
```

```
NAME
```

```
ls - list directory contents
```

```
SYNOPSIS
```

```
ls [OPTION]... [FILE]...
```

```
DESCRIPTION
```

```
List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort.
```

Mandatory arguments to long options are mandatory for short options too.

```
-a, --all
```

```
do not ignore entries starting with .
```

```
-A, --almost-all
```

```
do not list implied . and ..
```

```
--author
```

```
with -l, print the author of each file
```

```
-b, --escape
```

```
print octal escapes for nongraphic characters
```

```
--block-size=SIZE
```

```
use SIZE-byte blocks
```

```
-B, --ignore-backups
```

```
do not list implied entries ending with ~
```

```
-c with -lt: sort by, and show, ctime (time of last modification of file status information) with -l: show ctime and sort by name otherwise: sort by ctime
```

```
-C list entries by columns
```

--color[=WHEN]

control whether color is used to distinguish file types. WHEN may be 'never', 'always', or 'auto'

-d, --directory

list directory entries instead of contents, and do not dereference symbolic links

Manual page ls(1) line 1/208 18%

Come potete vedere man formatta a video una serie di informazioni sul comando ls:

- per comodità infatti il manuale è stato diviso in sezione, una per ogni tipologia di comandi, ls è nella prima (1), la sezione dei comandi utente (user commands);
- NAME contiene il nome del comando così come va scritto nella shell ed una sua breve descrizione;
- SINOPSIS indica come va usato il comando in tutti i possibili casi, comprese le eventuali opzioni e gli argomenti;
- DESCRIPTION fornisce una descrizione più dettagliata del comando;
- segue la lista dei possibili argomenti del comando con la loro descrizione.

Per muovervi all'interno di documenti molto lunghi premete i tasti "page up" e "page down" ("Pag con freccia all'insù" e "Pag con freccia all'ingiù" sulla vostra tastiera) e potrete andare su e giù nel testo visualizzandone la parte che v'interessa.

Cercate un termine all'interno del testo? Premete il tasto "/" (sopra il 7 nella tastiera italiana) e scrivete la stringa cercata, poi date invio. Man vi evidenzierà tutto le occorrenze trovate.

Ma non finisce qua: man fa altre mille cosette simpatiche. Quali? Beh, chi meglio di man stesso può dirvelo. Date il comando

\$ man man

Si esatto, come avrete già immaginato, abbiamo chiesto a man di visualizzare il suo stesso manuale. Consultatelo usando una Ubuntu, trovate in italiano tutta una serie di utili descrizioni sul modo in cui sono organizzati i manuali, e capirete meglio l'output di man quando avete chiesto informazioni sul comando ls.

Ultima cosa: quando avete terminato la consultazione e volete uscire dalla pagina di manuale, premete il tasto "q", sta per quit.

@:\>

Pubblicato alle [7.46](#) Argomenti: [amministrazione di sistema](#), [guida](#), [Linux](#), [tutorial](#)

sabato 17 novembre 2007

Linux: comandi info, apropos, whatis per ulteriori informazione sui comandi

Sempre per non perdersi nella foresta dei comandi Linux, dopo il post scorso sull'[uso di man](#), l'interfaccia da shell alla documentazione presente sul vostro sistema Linux, in questo post conclusivo sull'argomento vi mostro altri 3 comandi per accedere ad ulteriore documentazione: info, whatis e apropos.

Info in qualche modo è analogo a man: per usarlo basterà dare il comando info nomecomando e anch'esso permette di accedere alla documentazione dei comandi, ma le due documentazioni non sono equivalenti: alcune cose infatti che trovate nei file di man non le trovate in quelli di info e viceversa. Anche il sistema di consultazione è diverso: quello di info è un pò più ostico trattandosi di pagine ipertestuali.

Potrete saltare da una pagina all'altra premendo *invio* quando il cursore è su un termine che contiene un rimando ad un'altra pagina. Premendo *u* si torna al livello precedente, con *n* si passa alla pagina seguente, con *p* alla precedente, *i* torna all'indice, */* permette di fare ricerche, *?* mostra i comandi disponibili.

Passiamo ora al comando whatis. Nel post precedente vi avevo spiegato come i manuali sono raccolti in sezioni: whatis nomecomando restituisce il numero della sezione del manuale in cui ha trovato la stringa "nomecomando" ed una sua descrizione, quella che potete vedere nella sezione NAME quando ad esempio avete dato il comando man ls, proprio all'inizio della schermata. Esempio:

```
$ whatis ls
ls (1) - list directory contents
```

Scopriamo così che ls si trova nella sezione 1 dei manuali e dalla descrizione del comando capiamo cosa fa, la stessa descrizione, ripeto, che il comando man ls presenta nella prima sezione NAME. In realtà whatis così non è poi di grandissima utilità, a meno di trovarci in questa situazione:

```
$ whatis passwd
passwd (1) - change user password
passwd (1ssl) - compute password hashes
passwd (5) - the password file
```

Whatis ci restituisce tutte le occorrenze di passwd con tanto di indicazione della sezione in cui si trova e ovviamente la breve descrizione del comando. Per come è congegnato man, quando diamo il comando man passwd ci viene mostrata solo la prima occorrenza, per visualizzare le altre 2 occorre indicarle esplicitamente, per esempio:

```
man 5 passwd
```

Tenetelo presente: quando chiedete a man un certo comando ma vi compare altro, controllate con whatis la presenza di altre voci che siano maggiormente corrispondenti a quanto cercate.

Ultimo comando di cui ci occupiamo è apropos "nome comando": cerca la stringa corrispondente a "nome comando" nella sezione DESCRIPTION delle pagine visualizzate da man, notare l'uso degli apici per parole composte.

Bene, e con questo abbiamo terminato il veloce sguardo agli strumenti grazie alla quale avere rapidamente informazioni sui comandi Linux. Naturalmente rimane il web come ulteriore fonte d'informazione, ma lì non credo di dovervi dire qualcosa: voglio solo segnalarvi due ottime fonti di informazioni che trovate nella barra di navigazione sinistra, alla sezione "Risorse sul Web": si tratta di "Appunti di informatica libera" e di "Truelite"; entrambi in italiano, chiare, complete, e molto molto utili per approfondire quanto andremo a vedere nei prossimi post sull'argomento. A presto. :)

@:\>

Pubblicato alle [6.56](#) Argomenti: [amministrazione di sistema](#), [guida](#), [Linux](#), [tutorial](#)