

Giornale di Sistema

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

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

Appunti su Innotek VirtualBox

(<http://www.virtualbox.org/>)

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

Aggiornato al 13 novembre 2007

venerdì 9 febbraio 2007

Virtualizzazione: cos'è e quando usarla

Una cosa che mi ha sempre affascinato è l'idea che su un PC potessero girare più sistemi operativi contemporaneamente, ognuno nella sua finestra, e fare più cose insieme senza che si dessero il minimo fastidio. Quando ho scoperto e provato i programmi di virtualizzazione un piccolo sogno si è realizzato. Ma cos'è la virtualizzazione? A cosa può servirci?

Andiamo con ordine.

Cos'è la virtualizzazione

Consideriamo un attimo com'è fatto il nostro PC.

Un processore, una buona dotazione di memoria RAM, la scheda grafica, scheda audio, un disco rigido, il lettore DVD, un pò di porte USB ed un modem sono la dotazione più comune.

Prendiamo ora un programma che simula tutto questo. Ecco, in estrema sintesi (non me ne vogliano gli esperti, voglio solo far capire senza l'intenzione di essere rigoroso) questa è la virtualizzazione: ricreare all'interno di un programma un intero PC.

Quando usare la virtualizzazione

Ma in concreto poi, quando ci conviene usare la virtualizzazione?

Vediamo qualche situazione da usare come spunto per le nostre personali esigenze: Quasi sempre all'acquisto del nostro PC c'è Windows pre-installato.

- Immaginate ora di voler installare Linux su un PC con Windows già installato.

Dovreste "sezionare" il vostro disco rigido in più parti (in gergo si chiamano *partizioni*), una che resta al Windows già installato (stando bene attenti a non danneggiarlo), le altre (ne servono almeno altre 2) da destinare all'installazione di Linux. Ma perché impiegare tanto tempo e rischiare che alla fine non funzioni più nulla, ne Windows e ne Linux? Installiamo invece il nostro programma di virtualizzazione e senza mettere in pericolo i nostri dati ne Windows già funzionante, ci ritroviamo con un PC virtuale su cui installare e usare Linux.

- Provare programmi su un PC "cavia" virtuale.

Non è necessario "sporcare" la macchina che magari usiamo per lavoro installando software del cui funzionamento non siamo sicuri. Creiamo un salvataggio del PC virtuale (in gergo *Snapshots*, istantanee) e poi installiamo il programma da provare sulla macchina virtuale dove abbiamo installato Windows (si, avete capito bene, installiamo Windows all'interno di Windows stesso). Qualunque cosa accada, in pochi minuti siamo in grado di ritornare ad avere il nostro PC virtuale funzionante semplicemente ricaricando il salvataggio fatto in precedenza.

- Provare nuove versioni di un sistema operativo.

Volete provare Windows Vista oppure una nuova distribuzione di Linux senza modificare nulla nel vostro PC? Niente paura: lo installate sulla macchina virtuale e fate tutte le prove che volete.

Usare vecchie versioni di sistema operativo.

Eh si, a volte si pone il problema di tornare indietro ad una vecchia versione di un sistema operativo. Ad esempio il programma di contabilità aziendale gira solo sotto Windows 2000 e nessuno vuole spendere soldi per la nuova versione tanto quella vecchia va benissimo. Oppure un vecchio corso d'inglese gira solo sotto Windows 3.1/95 (si, è proprio vecchio) e non posso certo reinstallare Windows 95 solo per far girare il corso d'inglese. Tra l'altro con i PC più recenti questo sistema operativo non gira neanche, non riconoscendo buona parte dell'hardware attualmente in circolazione.

- Far girare un programma scritto per un sistema operativo diverso da quello installato.

E' il caso tipico di chi usa per lo più Linux e ha un programma, magari quello fornito con la macchina digitale, che funziona solo in Windows. Invece di installare Windows sulla macchina reale solo per scaricare le foto, lo installa in totale sicurezza su quella virtuale a cui accede solo quando serve.

- Sviluppare e testare programmi per più sistemi operativi.

Uno sviluppatore sta scrivendo e testando un'applicazione in versione Windows, Linux, OS X, dovrebbe avere installati tutti e tre i sistemi operativi sulla sua macchina e riavviare ogni volta che vuole cambiare sistema operativo: inutile. Virtualizzerà gli altri due sistemi operativi e li avvierà in due finestre diverse del programma di virtualizzazione potendo così lavorare in parallelo in tutti e tre i sistemi operativi.

- Testare un sito Web con diversi browser e sistemi operativi.

Immaginate di voler sviluppare un sito web. Come sarà visualizzato a seconda della combinazione browser-sistema

operativo? Niente di più semplice: in una finestra fate girare un PC virtuale con Linux configurato con Firefox, Opera o quello che serve, in un'altra Windows con Explorer versione 5.5, in un'altra ancora Windows con Explorer 6 e così via.

- Navigare in maniera sicura.

Immaginiamo di aver configurato un PC virtuale per poter navigare in internet. Inavvertitamente rimaniamo infettati da un virus: è infettato solo il PC virtuale e lì resta confinata l'infezione. Anche il tentativo di un Cracker (come si chiamano esattamente quelli che gli organi d'informazione definiscono Hacker) di entrare abusivamente nel nostro PC rimarrà confinato a quello virtuale con cui stiamo navigando.

Bene, anche se questi sono solo alcuni esempi in cui torna utile usare la virtualizzazione, spero vi abbiano sollecitato quel tanto che basta la curiosità per volerla provare sul campo.

Nel post "Virtualizzazione: installiamo Virtualbox" vedremo come farlo in concreto.
@:\>

Pubblicato alle 16.10 Argomenti: guida, tutorial, virtualizzazione

mercoledì 14 febbraio 2007

Virtualizzazione: installiamo Virtualbox

Aggiornamento del 5 settembre 2007

Il post resta valido anche per le versioni successive di VirtualBox (qui è utilizzata la 1.3.4).

@:>

Come programma di virtualizzazione VirtualBox è una valida scelta per almeno 3 motivi:

1. è open source e gratuito per uso personale o a scopo di valutazione;
2. è disponibile sia per Windows che per Linux;
3. è veramente semplice da usare e allo stesso tempo altrettanto potente di programmi molto più blasonati.

Bene, rimbocchiamoci le maniche e cominciamo.

Scaricare e installare VirtualBox

Fate il download di VirtualBox nella versione Windows. Appena terminato il download un doppio click sul file scaricato ed eccoci nella finestra di benvenuto (**fig. 1**); clicchiamo su *Next*.



fig. 1

Dopo aver letto la licenza scegliamo *I accept...* e clicchiamo su *Next* (**fig. 2**).

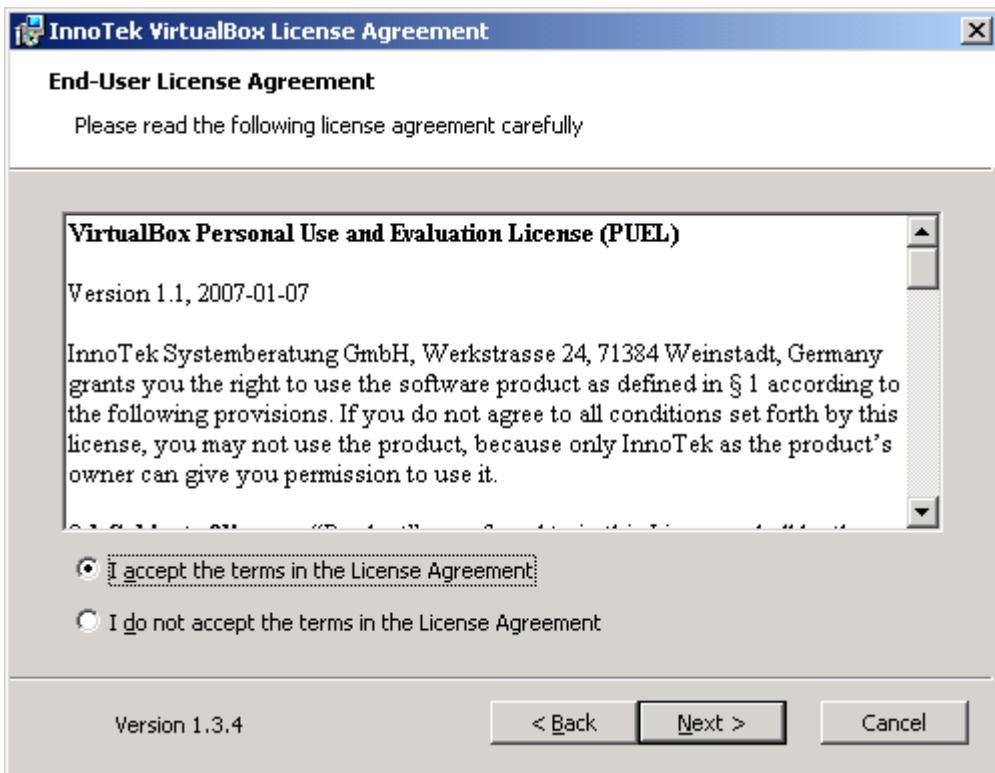


fig. 2

Ci verrà chiesto dove installare il programma, lasciate pure le impostazioni predefinite (normalmente la cartella Programmi) e cliccate su *Next* (fig. 3).

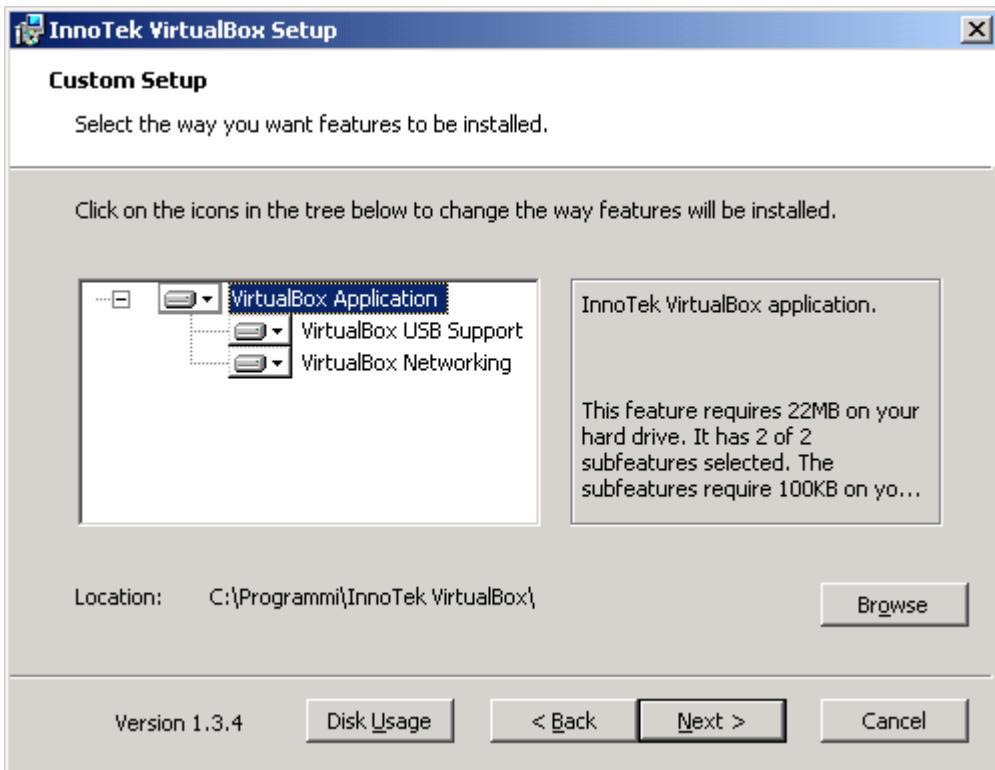


fig. 3

A questo punto non ci resta che cliccare su *Install* per iniziare l'installazione di VirtualBox (fig. 4).

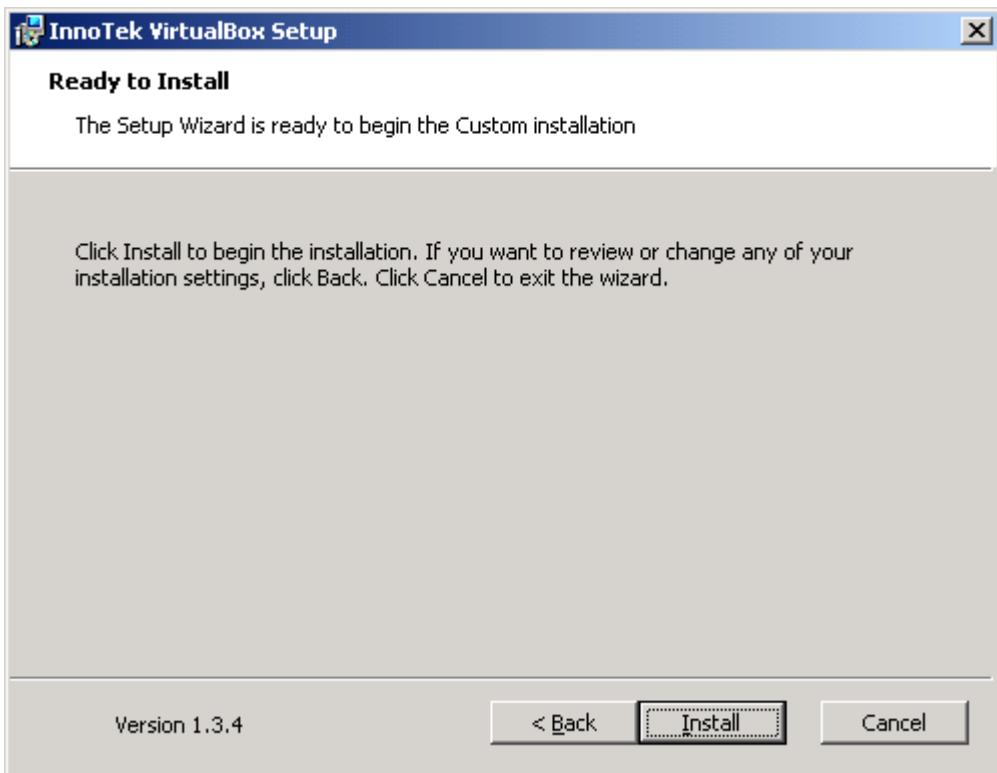


fig. 4

Portata a termine l'installazione clicchiamo su *Finish* (fig. 5).



fig. 5

Ci verrà chiesto di riavviare Windows, chiudiamo prima tutte le applicazioni aperte e infine clicchiamo su Yes. (fig. 6).

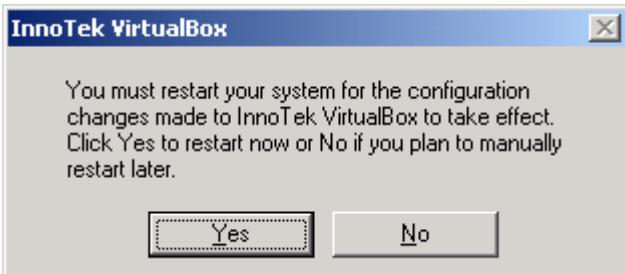


fig. 6

Terminato il riavvio di Window potremo finalmente lanciare VirtuaBox dal pulsante Start e cominciare la configurazione della nostra prima macchina virtuale (fig. 7).

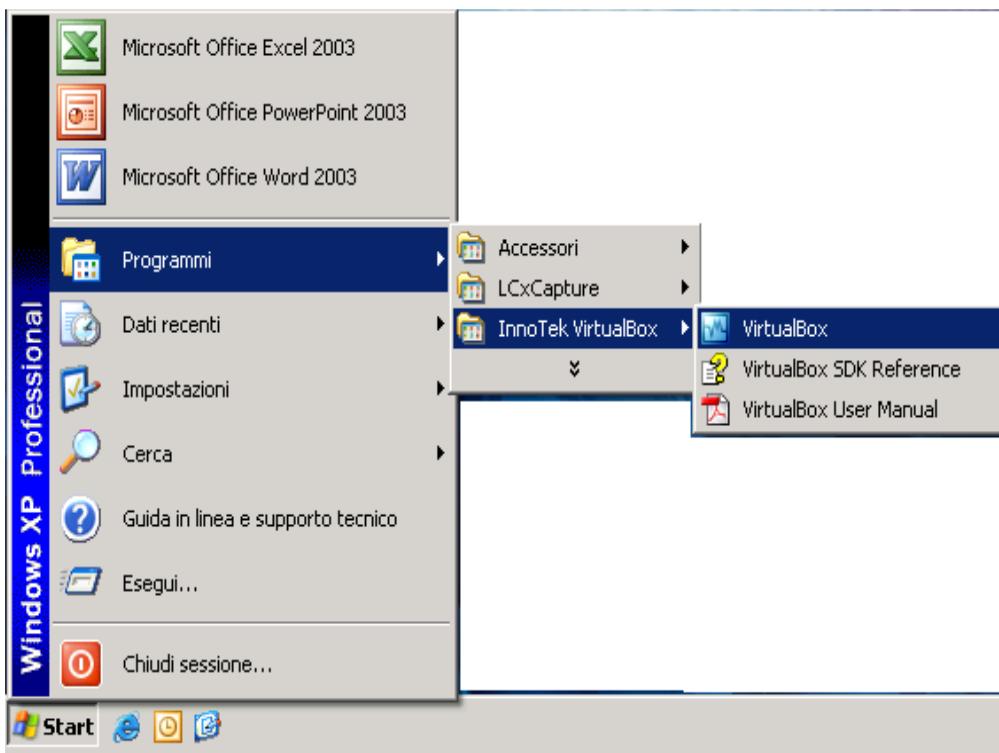


fig. 7

Ma di questo ci occuperemo nel prossimo post, "Virtualizzazione: creiamo un PC virtuale con VirtualBox (1a parte)".

@:>

Pubblicato alle 12.09 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

martedì 20 febbraio 2007

Virtualizzazione: creiamo un PC virtuale con VirtualBox (1a parte)

Aggiornamento di: martedì 12 giugno 2007

Dopo l'uscita di VirtualBox versione 1.4, questo post è stato sostituito da "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 1)"; buona lettura.

@:>

Nel post "Virtualizzazione: installiamo Virtualbox" abbiamo visto come installare VirtualBox, un potente programma open source di virtualizzazione. Grazie ad esso diventa realtà il sogno di far girare più sistemi operativi simultaneamente su un solo PC, ognuno nella sua finestra, e far in modo che condividano dati fra loro.

Con VirtualBox potete creare dei PC virtuali completi di tutte le loro periferiche: disco rigido, scheda grafica, audio, periferiche USB e così via. Su ognuno di essi è possibile installare un sistema operativo e farlo funzionare simultaneamente a tutti gli altri, compreso quello che sta girando sul PC reale. Potete anche fare in modo che i PC, quello reale detto *host*, e quelli virtuali detti *guest*, comunichino tra loro condividendo file e cartelle.

Vediamo allora come creare un PC virtuale. Siamo in Windows, una scelta carina mi sembra quella di creare una macchina virtuale su cui installare una distribuzione Linux. La creazione della macchina virtuale è uguale per tutte le distribuzioni di Linux, per questo esempio useremo la Suse Linux 10.2, usatela come esempio passo-passo per la vostra distribuzione.

Cominciamo.

Dal pulsante *start* ---> *Programmi* ---> *InnoTek VirtualBox* ---> *Virtual Box* lanciamo con un clic il nostro virtualizzatore (fig. 1):

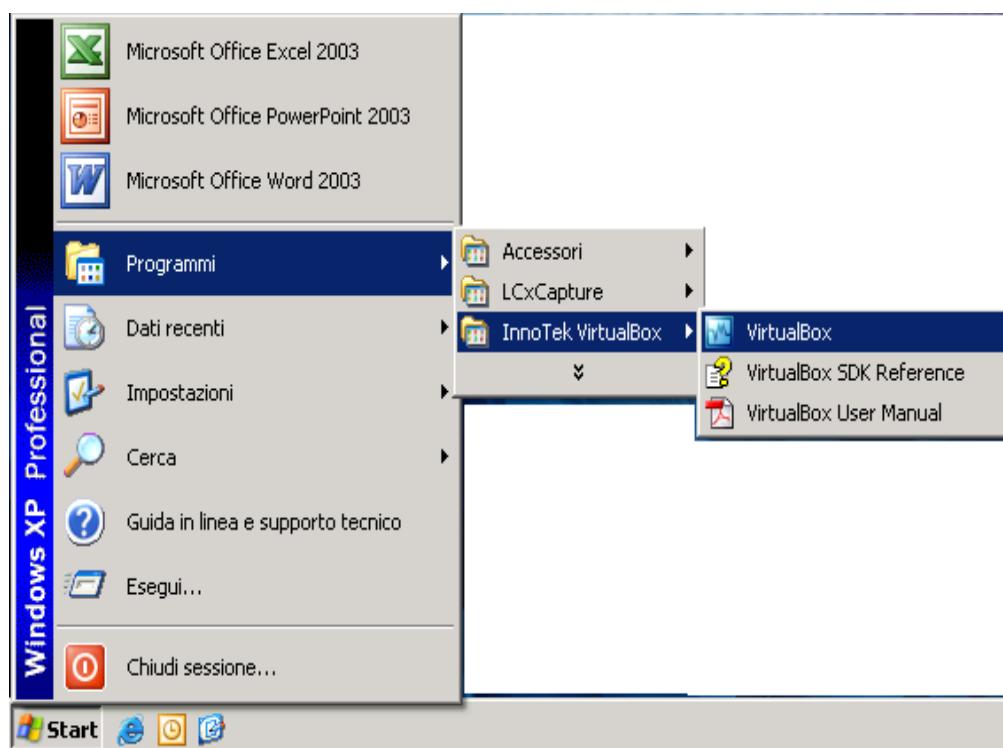


fig. 1

Eccoci in VirtualBox, clicchiamo su *New* per creare la nostra prima macchina virtuale (fig. 2):

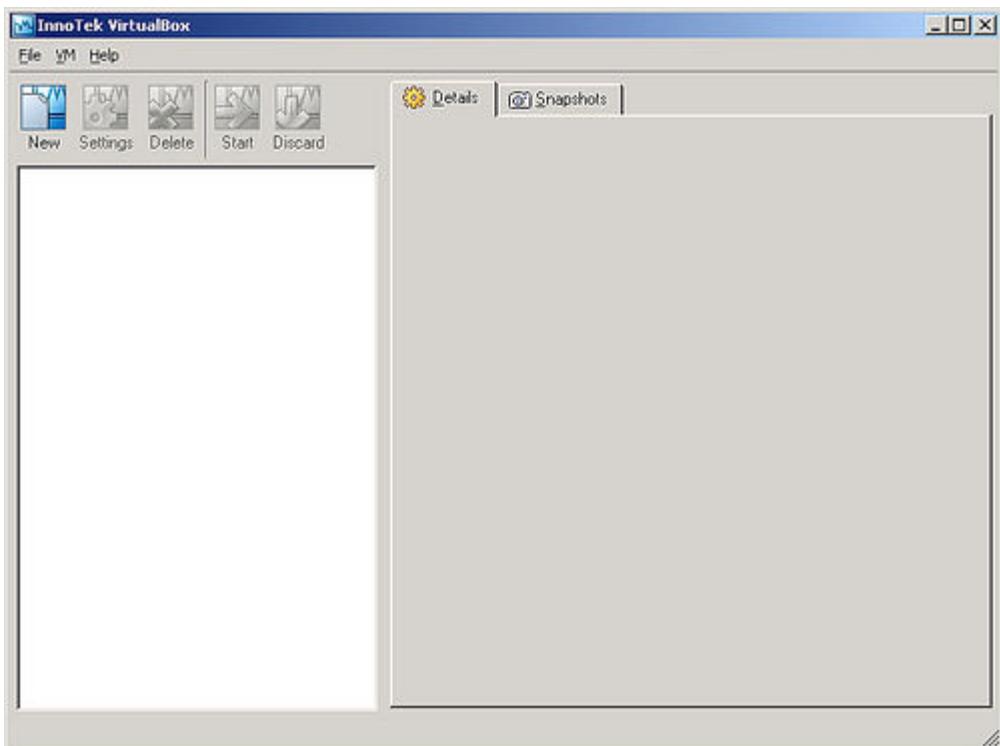


fig. 2

Il wizard che si apre ci guiderà nella creazione della macchina virtuale; clicchiamo su **Next** (fig. 3):

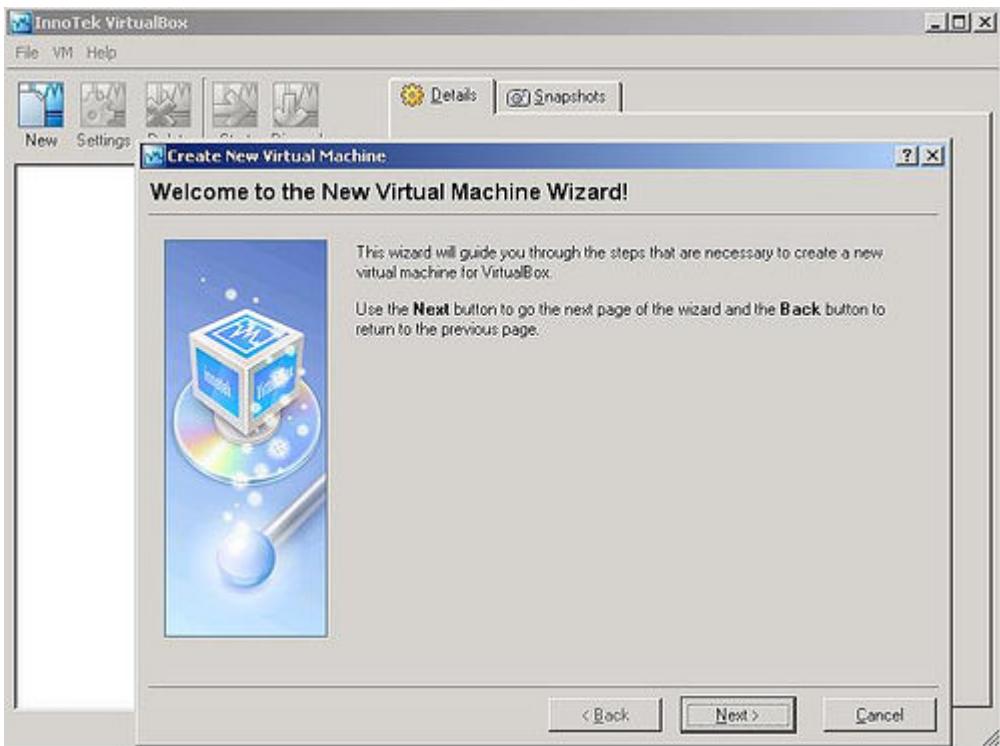


fig. 3

Nel campo **Name** scriviamo il nome della nostra macchina virtuale, ad esempio "open SuSE 10.2" oppure il nome della distribuzione che volete usare; nel campo **OS Type** selezioniamo Linux 2.6, questa è la versione del kernel (il kernel è il nucleo, il cuore di un OS) di Linux in dotazione alla distribuzione SuSE che installeremo, e andiamo avanti cliccando su **Next** (fig. 4):

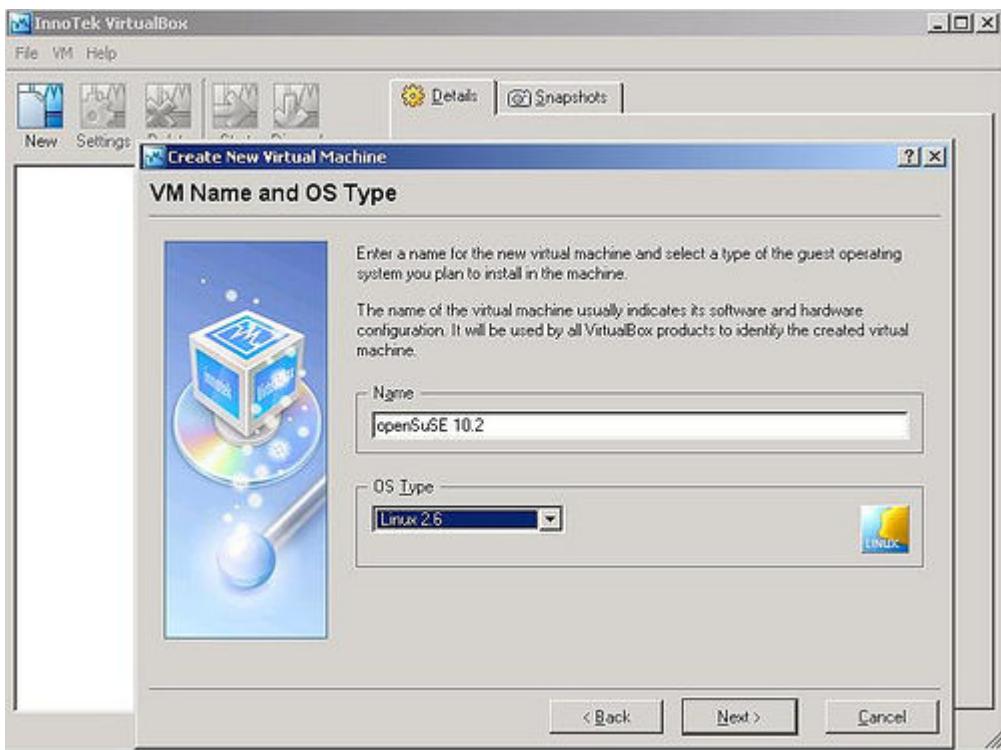


fig. 4

Scegliamo ora la quantità di memoria che la nostra macchina virtuale avrà a disposizione per funzionare: 256 MB sono da considerarsi il minimo perché tutti funziona ad una velocità accettabile, se invece avete un PC con almeno 1 GB di memoria, 512 MB è la quantità ottimale; fatta la vostra scelta cliccate su *Next* (fig. 5).

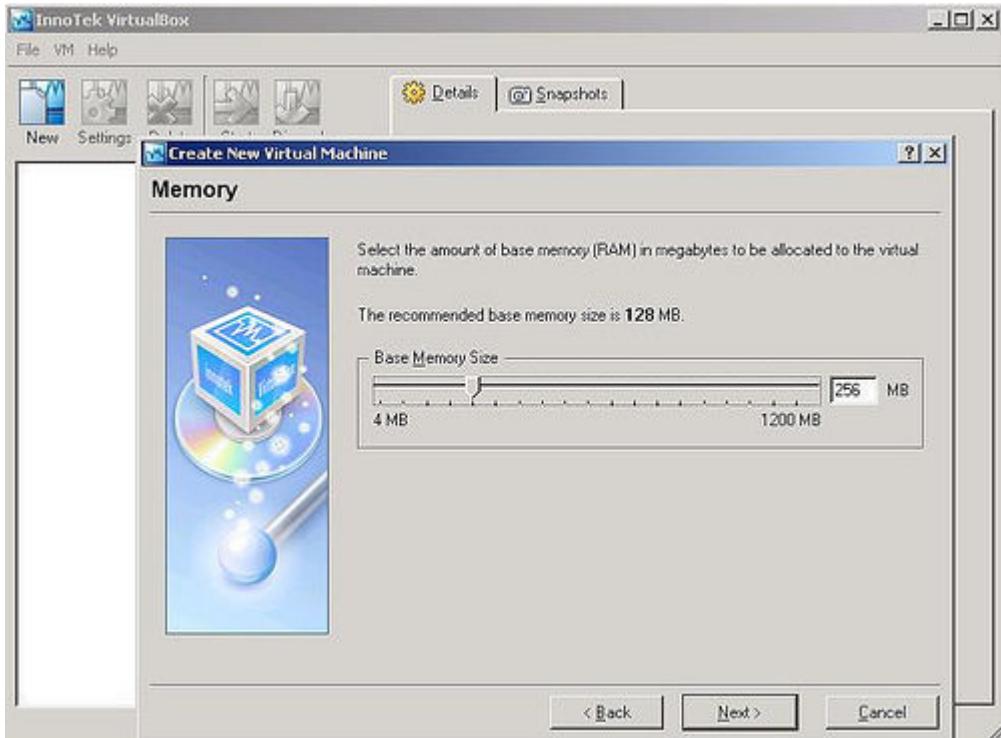


fig. 5

Il nostro PC virtuale ora ha bisogno di un disco rigido virtuale; clicchiamo su *New* (fig. 6):

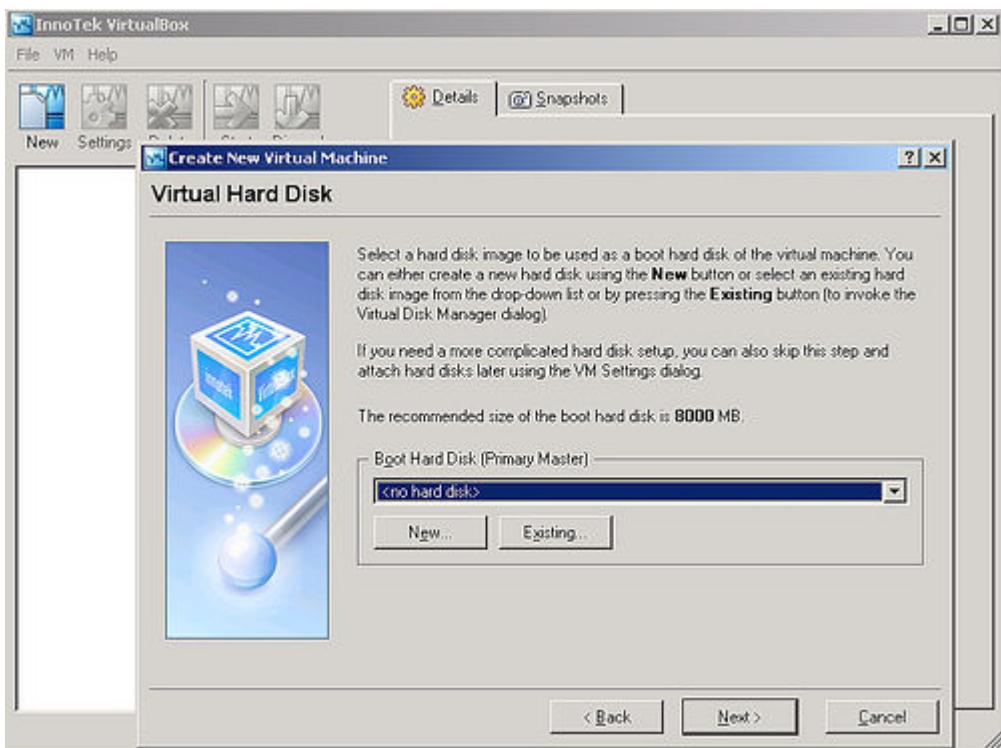


fig. 6

Il wizard che si è aperto ci guiderà nella creazione del disco rigido del nostro computer virtuale. Teniamo a mente una cosa: lo spazio che destineremo al disco rigido virtuale viene sottratto al disco rigido reale del nostro computer, pertanto assicuratevi di avere spazio sufficiente su quest'ultimo. Bene, proseguiamo cliccando su *Next* (fig. 7):



fig. 7

Dobbiamo scegliere il tipo di disco virtuale da allocare:

- **Dynamically expanding image;** lo spazio richiesto non verrà allocato subito tutto ma solo al momento dell'effettivo bisogno. Scegliendo per esempio gli 8 GB consigliati come grandezza del nostro disco rigido

virtuale, inizialmente questi occuperà solo pochi Kilobyte sul disco rigido reale. Man mano servirà più spazio questo verrà allocato sul momento, fino a quando avrà raggiunto il massimo prestabilito di 8 GB oltre il quale non va.

- **Fixex-size image:** la quantità di spazio prescelta viene allocata subito tutta.

Quale scegliere?

Valutate voi; la prima opzione permette di occupare meno spazio sul disco reale, la seconda però evita i rallentamenti che si verificano nel momento dell'allocazione "al volo" dello spazio; fatta la vostra scelta (personalmente ho scelto la prima per motivi di spazio), cliccate su *Next* (fig. 8).

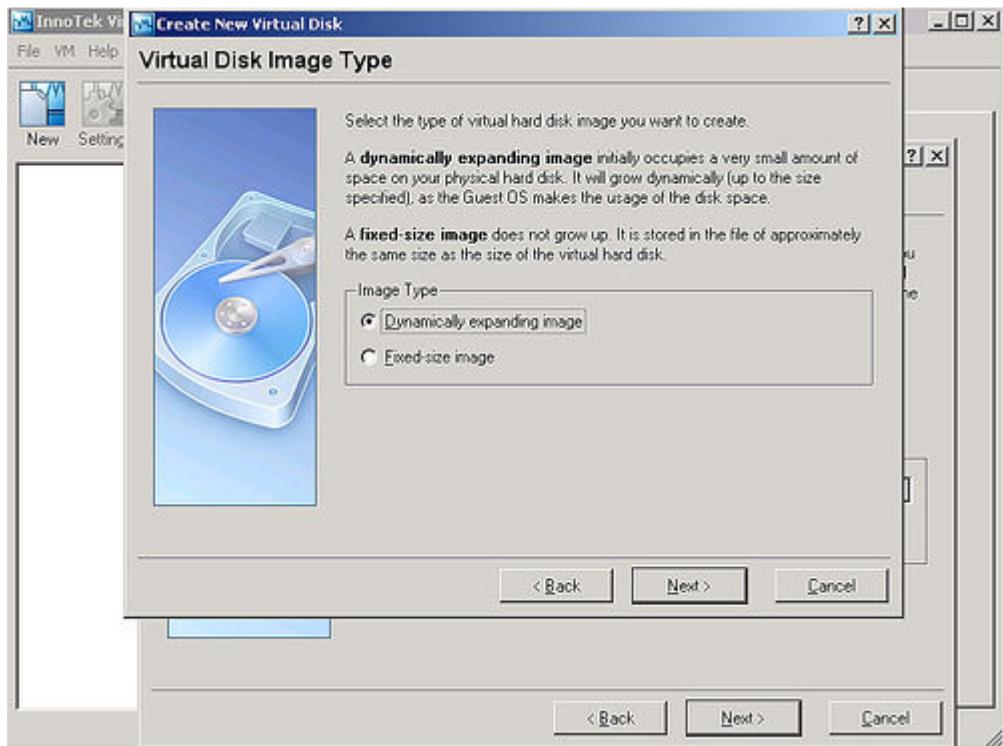


fig. 8

E sceglieremo infine la capienza del disco rigido virtuale; per provare comodamente Linux 10 GB saranno più che sufficienti. Ancora una volta, per proseguire clicchiamo su *Next* (fig. 9).

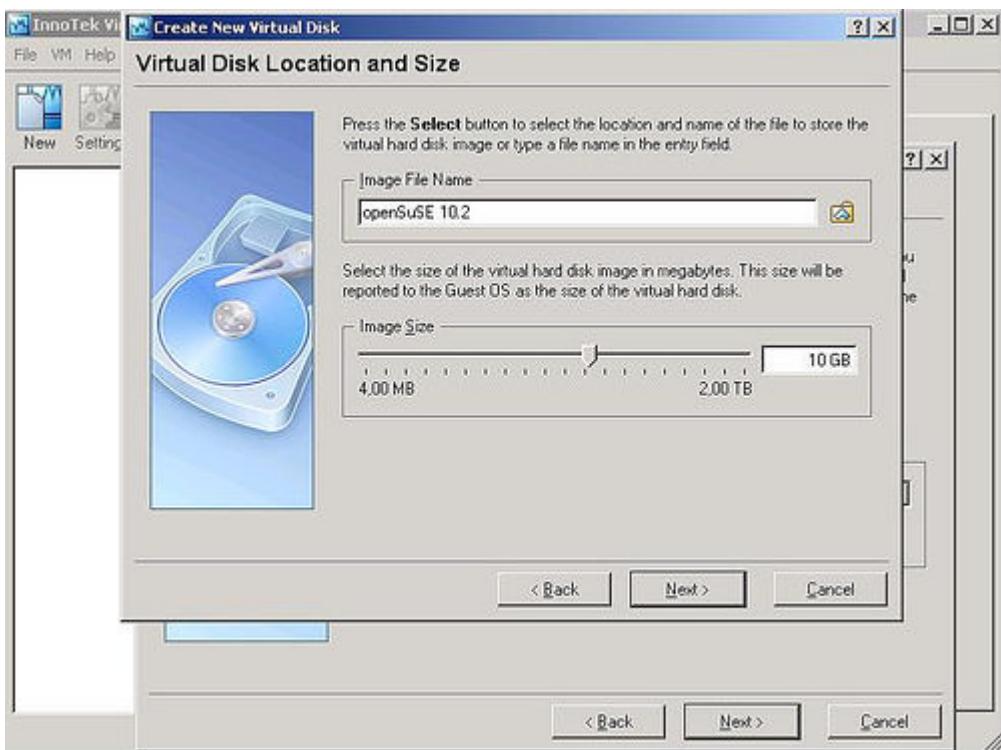


fig. 9

Se i dati forniti dalla schermata riepilogativa (tipo di disco rigido virtuale, cartella del disco rigido reale dove verrà collocato, spazio occupato) ci vanno bene, clicchiamo su *Finish* (fig. 10).

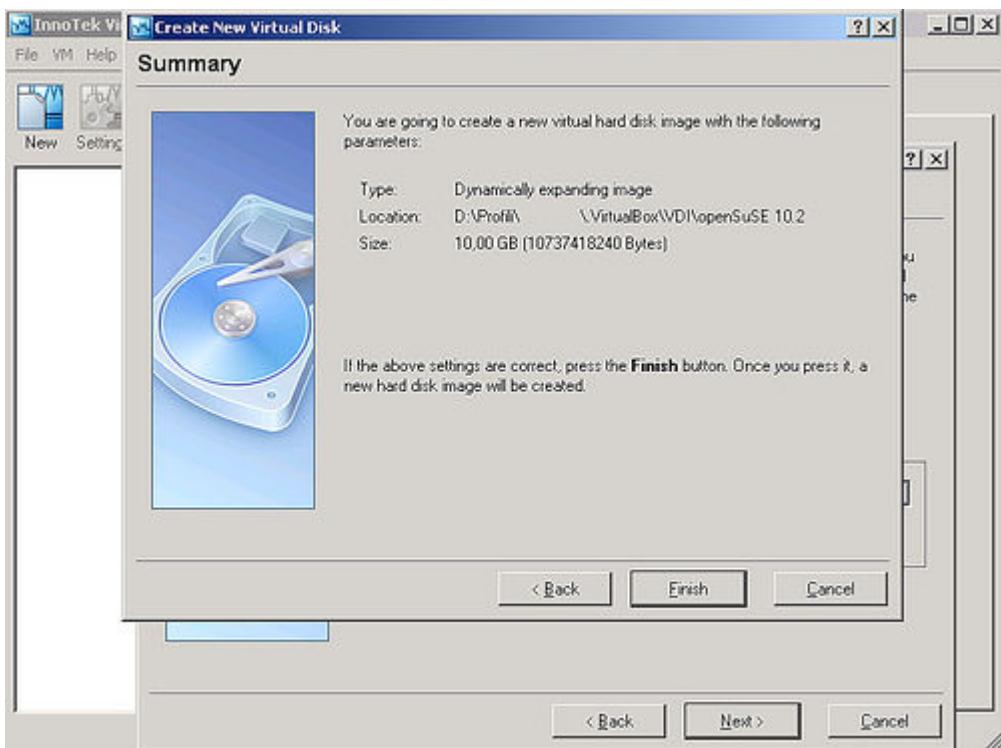


fig. 10

Bene, il disco rigido virtuale è pronto, clicchiamo su *Next* (fig. 11).

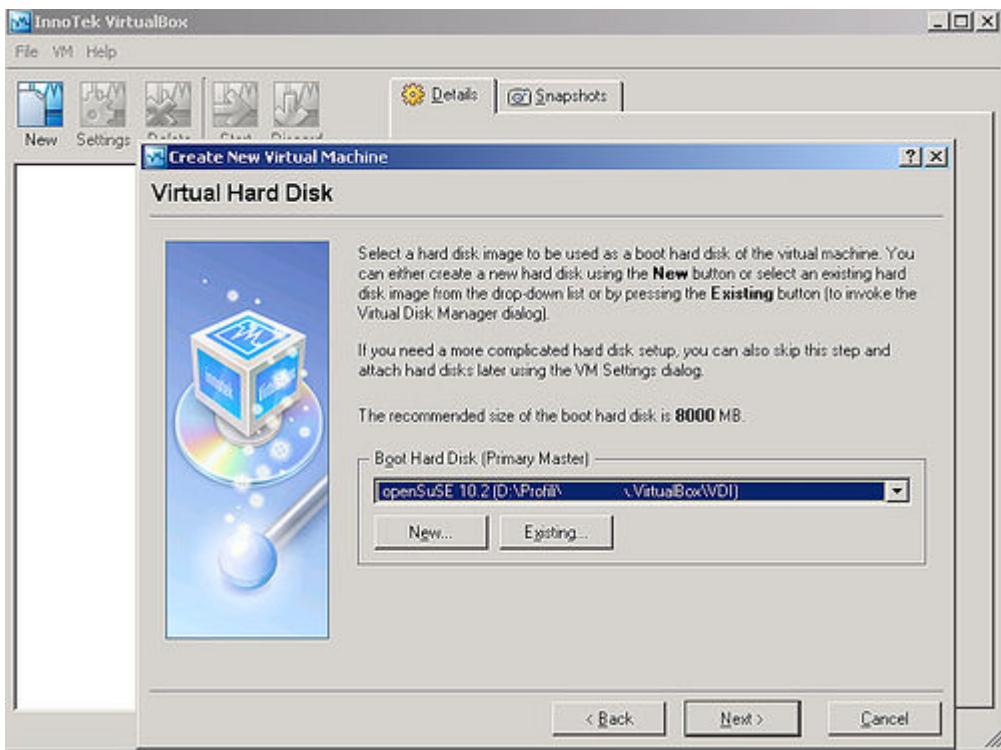


fig. 11

Anche il computer virtuale su cui installare Linux è pronto, terminiamo la procedura di setup cliccando su *Finish* (fig. 12).

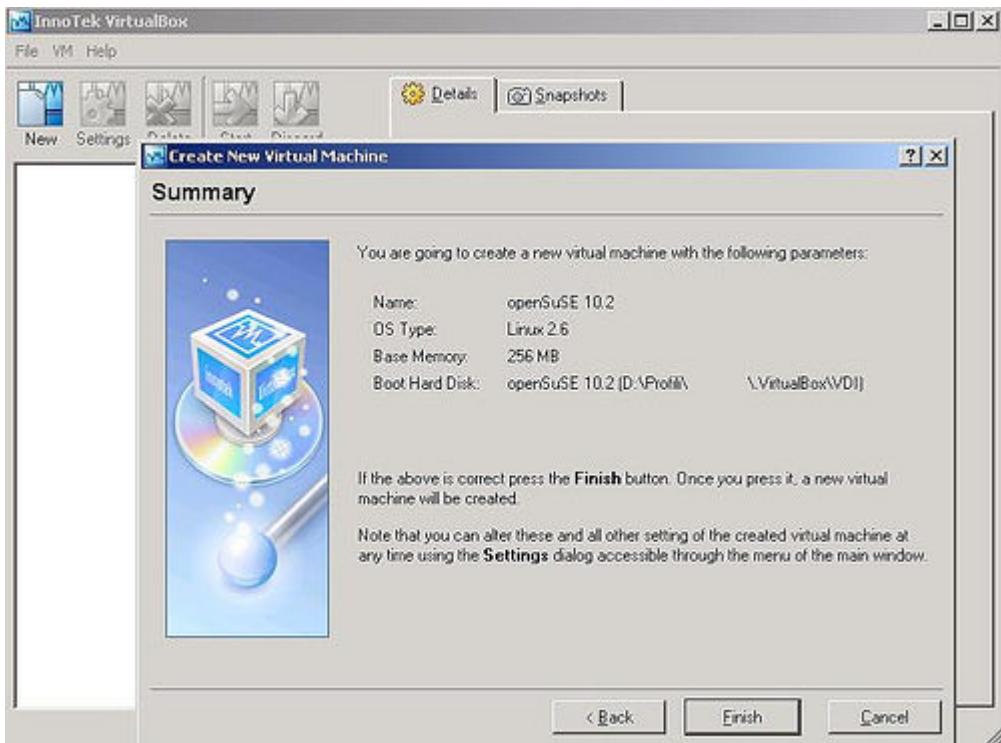


fig. 12

Ed eccolo finalmente, il nostro computer virtuale (fig. 13).

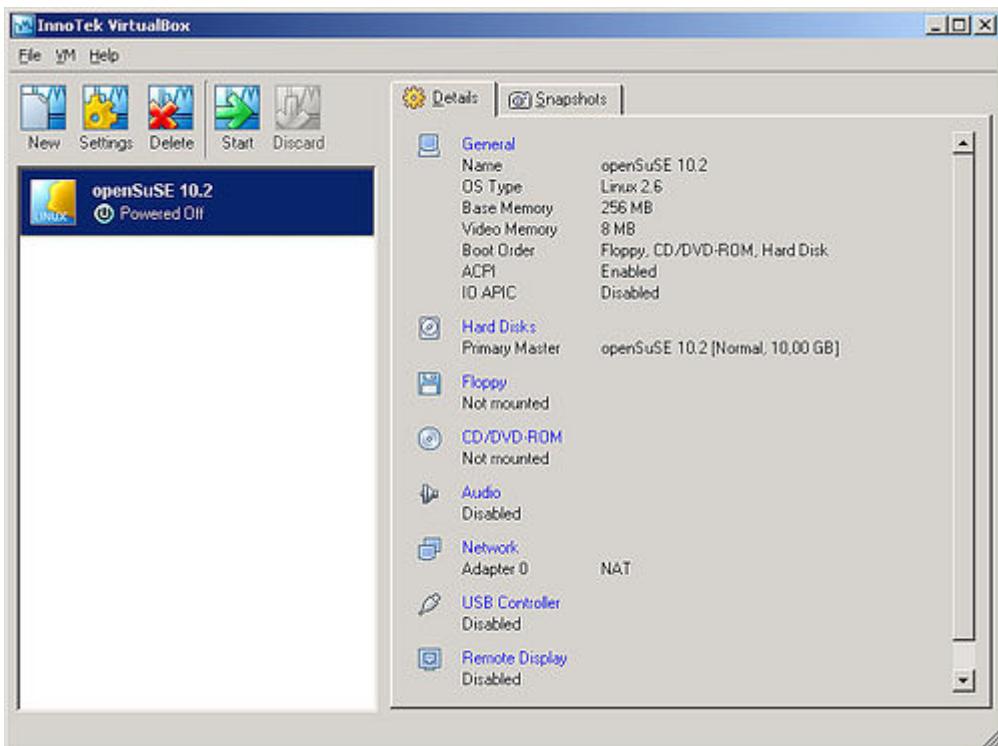


fig. 13

Come si può notare in questo momento è tutto spento (*Powered Off*): in realtà non è ancora configurato del tutto. Audio e controller USB non sono attivi, il CD/DVD-ROM ed il Floppy idem, la scheda grafica ha appena 8 MB di memoria... insomma, bisogna ancora lavorare un pò per affinare la configurazione.

Ed è quello che faremo nel post: "Virtualizzazione: creiamo un PC virtuale con VirtualBox (2a parte)".

@:>

Pubblicato alle 11.15 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

giovedì 22 febbraio 2007

Rilasciato Virtualbox 1.3.6

Ad appena 8 giorni di distanza dal precedente rilascio, è uscito la nuova versione di VirtualBox: la 1.3.6.

Sono stati corretti alcuni bug e apportati vari miglioramenti. Per esempio ora la memoria della scheda video del PC virtuale può essere portata fino a 128 MB.

Ecco una lista completa dei cambiamenti in questa nuova versione, mentre per scaricarla andate su VirtualBox.org:

- Windows installer: perform installation for all users instead of just the current user (old behavior still available)
- Linux installer: fixed license display to not block installation
- Linux installer: added RPM for openSUSE 10.2
- GUI: fixed problems with several keyboard layouts on Linux hosts
- GUI: added online help on Linux hosts (using kchmviewer)
- GUI: fixed handle leak on Windows hosts
- Graphics: increased VRAM limit to 128 MB
- BIOS: fixed CD/DVD-ROM detection in Windows Vista guests
- VMM: fixed incompatibility with OpenBSD 4.0
- VDI: fixed issues with snapshot merging
- Network: fixed incompatibility between Vista UAC and Host Interface Networking
- Network: fixed issues with Windows NT 4.0 guests
- Audio: fixed problem with ALSA on Linux before 2.6.18 causing system reboots
- RDP: added support for MS RDP 6.0 clients
- RDP: fixed issue with PAM authentication on certain distributions
- RDP: fixed sporadic disconnects with MS RDP clients
- RDP: fixed memory leak
- iSCSI: improved behavior when pausing a VM with iSCSI connections
- iSCSI: improved read timeout handling.

@:\>

Pubblicato alle 10.19 Argomenti: news, software, Virtualbox, virtualizzazione

mercoledì 28 febbraio 2007

Virtualizzazione: creiamo un PC virtuale con VirtualBox (2a parte)

Aggiornamento di: giovedì 14 giugno 2007

Dopo l'uscita di VirtualBox versione 1.4, questo post è stato sostituito da "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 2)"; buona lettura.

@:>

Nel post "Virtualizzazione: creiamo un PC virtuale con VirtualBox (1a parte)" avevamo configurato solo parzialmente una macchina virtuale per installare una distribuzione Linux; terminiamo adesso la configurazione.

Vi ricordo che la distribuzione Suse Linux 10.2 è stata presa solo come esempio, la procedura in realtà è uguale per tutti i sistemi operativi guest (quelli installati sul PC virtuale) supportati da Virtualbox.

Riavviamo VirtualBox (**fig. 1**), ci ritroveremo davanti alla macchina virtuale spenta (*Powered Off*). Rimangono da configurare il supporto per l'USB, il CD/DVD-ROM, la scheda audio, quella video. Selezionate la macchina virtuale ancora spenta e cliccate sul pulsante *Settings* (da tastiera *Ctrl+s*) nella barra degli strumenti:

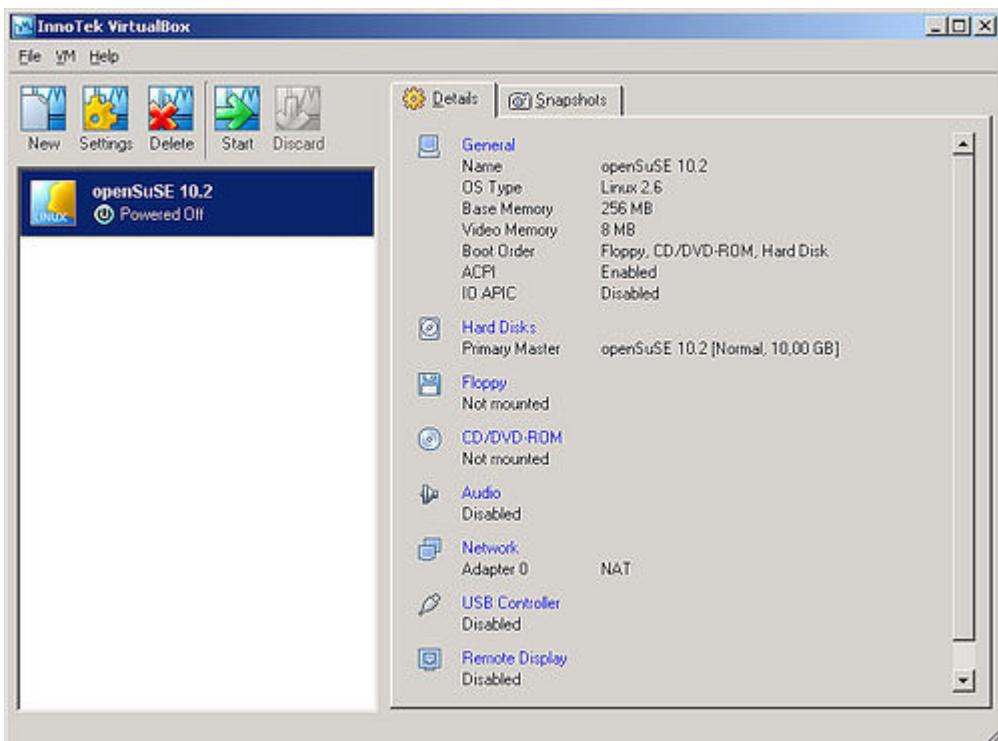


fig. 1

La voce *General* (**fig. 2**) nella barra di navigazione a sinistra è costituita da 2 schede visualizzate a destra:

- *Basic*: contiene una serie di voci quali il nome della macchina virtuale ed il sistema operativo da installarvi all'interno che abbiamo già configurato tramite i wizard visti la scorsa volta. A noi interessa la voce *Video Memory Size*; clicchiamo sul cursore e facciamolo scorrere fino a portarlo alla quantità desiderata di Megabyte da destinare alla scheda video. Tenete presente che in linea di massima, più sarà alta questa quantità e più sarà alta la risoluzione video ed il numero di colori a disposizione nel PC virtuale. Vi è però un limite: finché non installate le VirtualBox Guest Additions, non andrete comunque oltre la risoluzione massima di 1024 x 768, una volta installate invece potrete arrivare fino a 3200 x 2400.

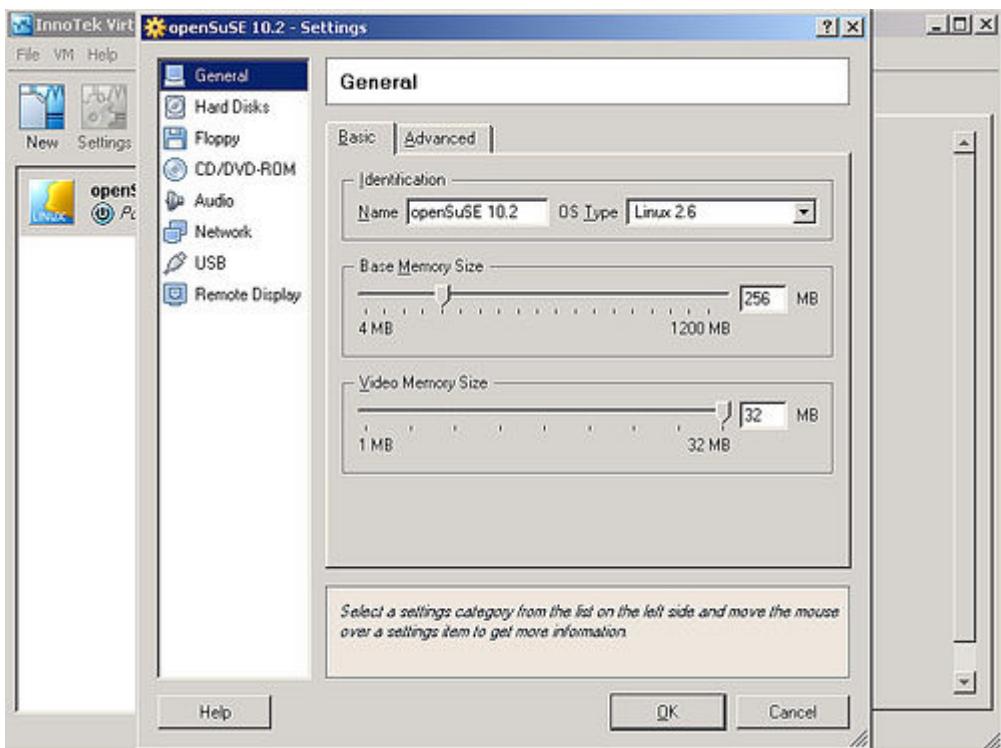


fig. 2

- **Advanced** (fig. 3); contiene la voce *Boot Order* (dell'Extended Features e Snapshot Folder non ci occupiamo lasciamo semplicemente le impostazioni predefinite). Questa è la lista dei dispositivi da cui la macchina virtuale cercherà di far partire il Sistema Operativo nell'ordine che leggiamo: cercherà prima sul floppy, poi CD/DVD-ROM, infine dal disco rigido (non compare ma si può scegliere anche da rete locale tramite la voce Network).

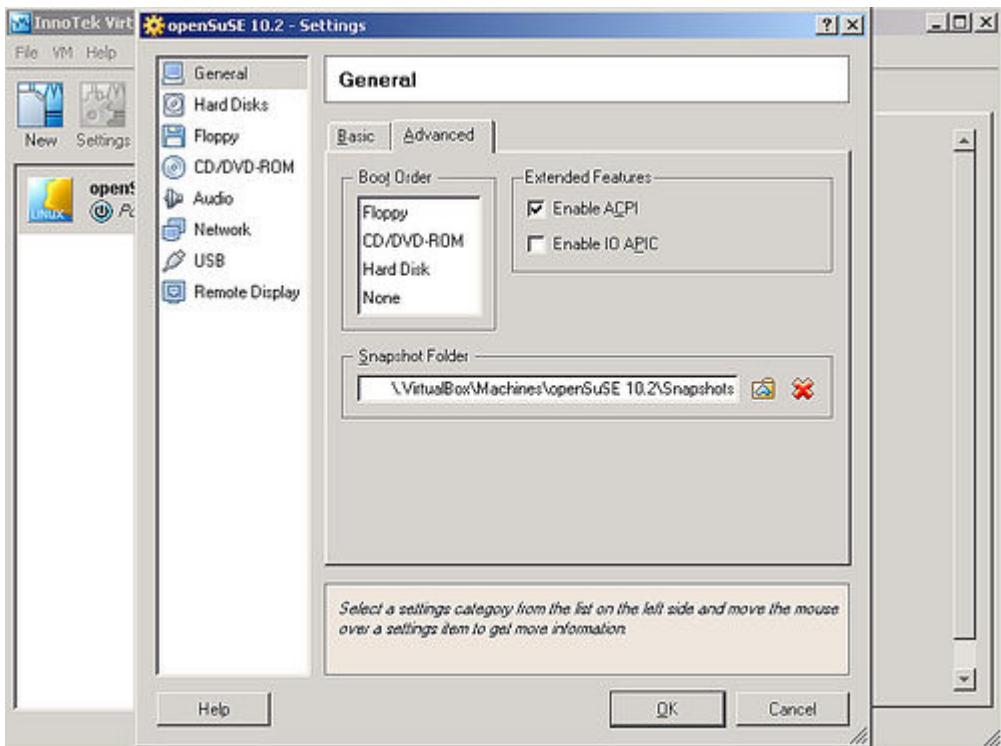


fig. 3

Lasciamo tutto com'è e andiamo avanti nella configurazione cliccando sulla voce *Hard Disk* (fig. 4):

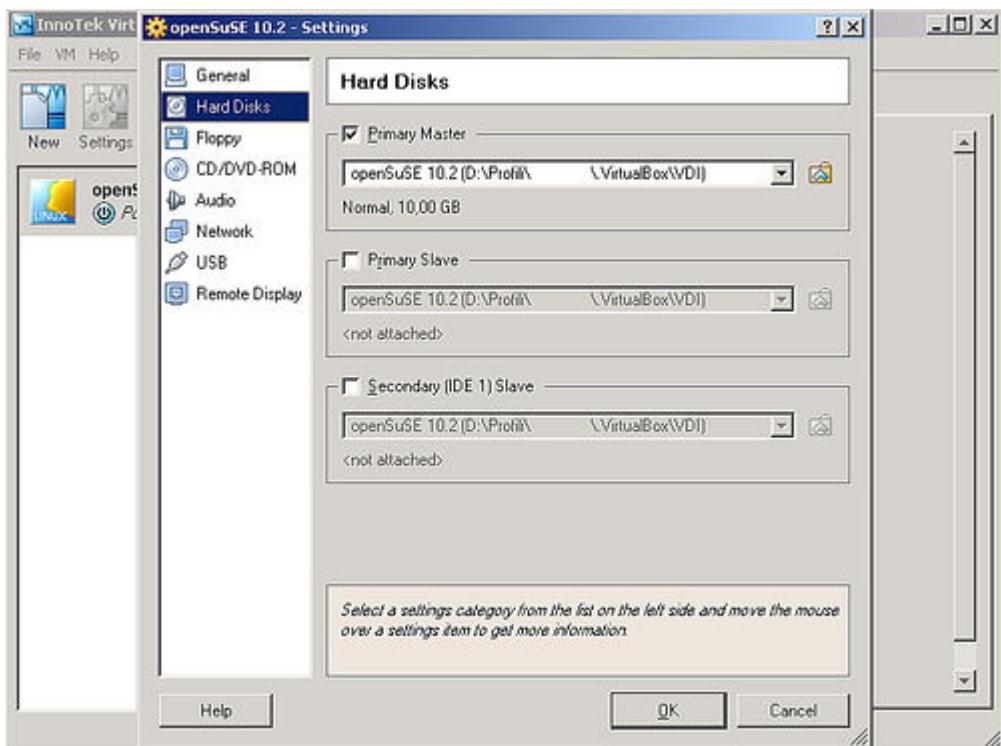


fig. 4

Quello che vedete (fig. 4) è il disco rigido del PC virtuale configurato come *Primary Master*, la sua grandezza (10 GB), e il file sul computer reale in cui è memorizzato il disco rigido virtuale. Come vedete possiamo aggiungere altri 2 dischi rigidi virtuali ma lasciate tutto così com'è e andiamo avanti cliccando sulla voce *Floppy* (fig. 5).

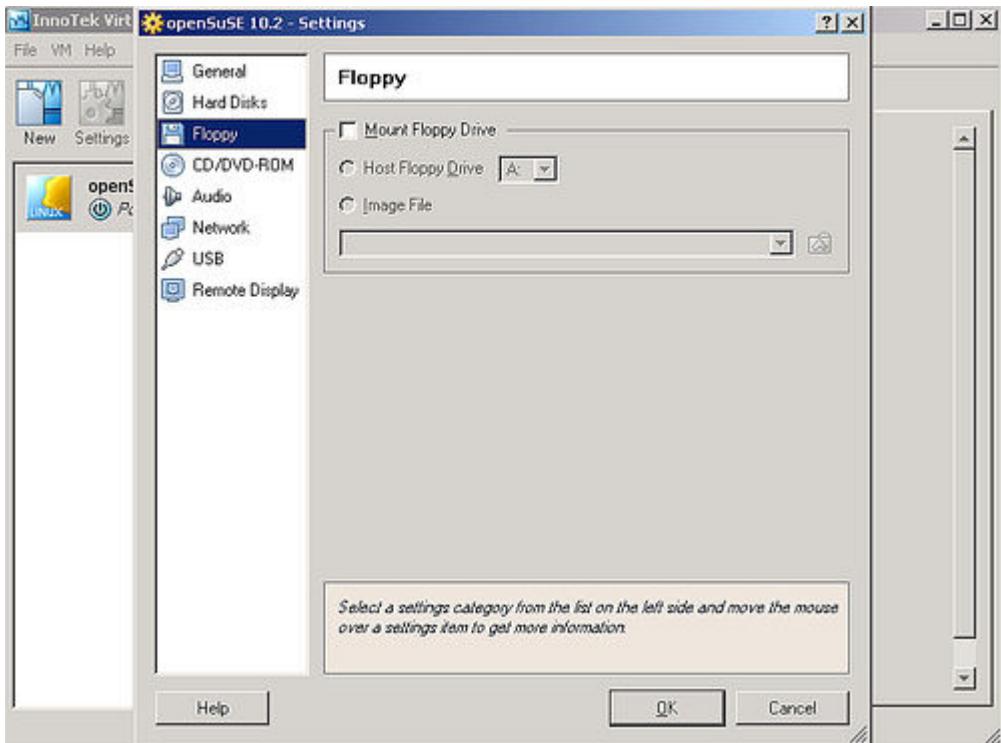


fig. 5

Del floppy ne faremo a meno (sulla mia macchina, un notebook, non è neanche presente) pertanto lasciate deselezionata la casella *Mount Floppy Drive* e cliccate sulla voce successiva, *CD/DVD-ROM* (fig. 6).

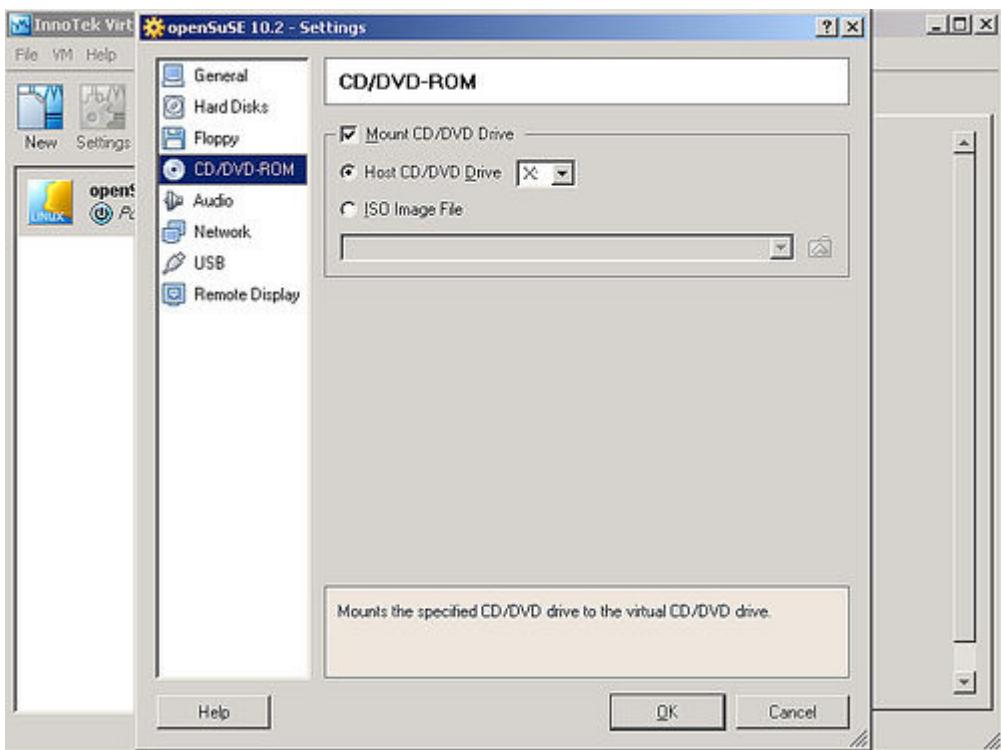


fig. 6

Spuntiamo intanto la casella *Mount CD/DVD Drive*, avremo così disponibile nel nostro PC virtuale anche il CD/DVD. Se il Linux da installare è su un CD/DVD allora scegliete la prima opzione, *Host CD/DVD Drive*. Se invece come spesso accade, avete a disposizione il file .iso, allora VirtualBox ci mette a disposizione una comodissima opzione per installare Linux direttamente da quel file senza doverlo più masterizzare prima. E' sufficiente cliccare sulla seconda opzione, *Iso image file* e poi nel punto indicato dalla freccia (fig. 7):

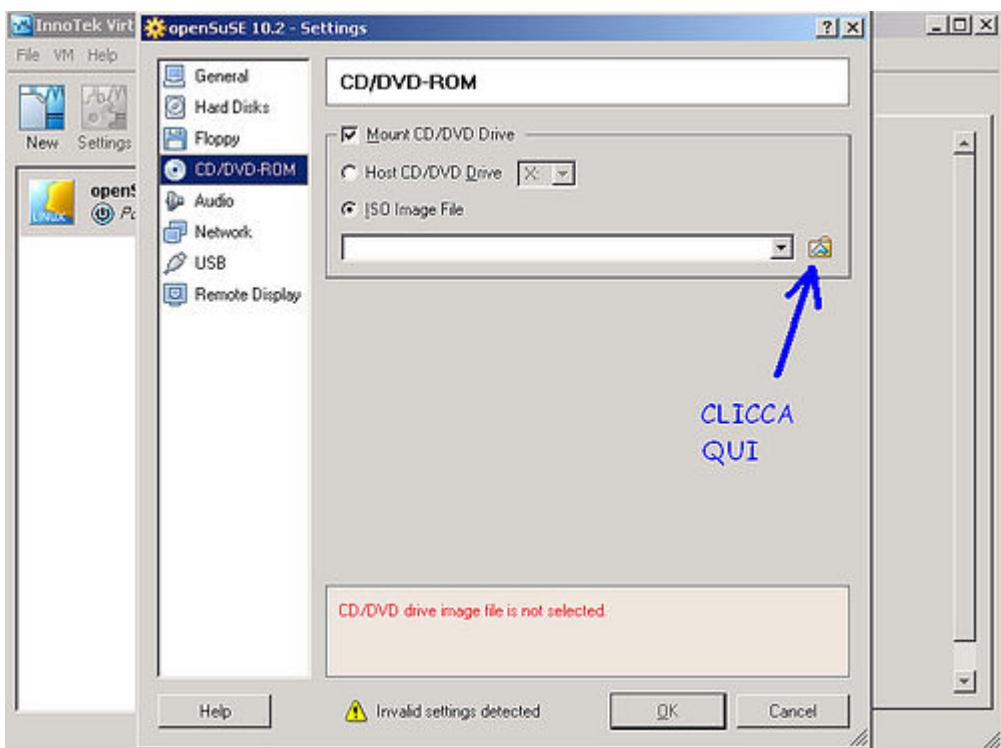


fig. 7

Parte un nuovo Wizard (fig. 8), quello del *Virtual Disk Manager*, il gestore di tutte le periferiche a disco del PC virtuale (infatti se guardate bene la schermata, oltre al tab *CD/DVD Image*, vi sono anche quello del disco rigido e del floppy, rispettivamente *Hard Disks* e *Floppy Images*).

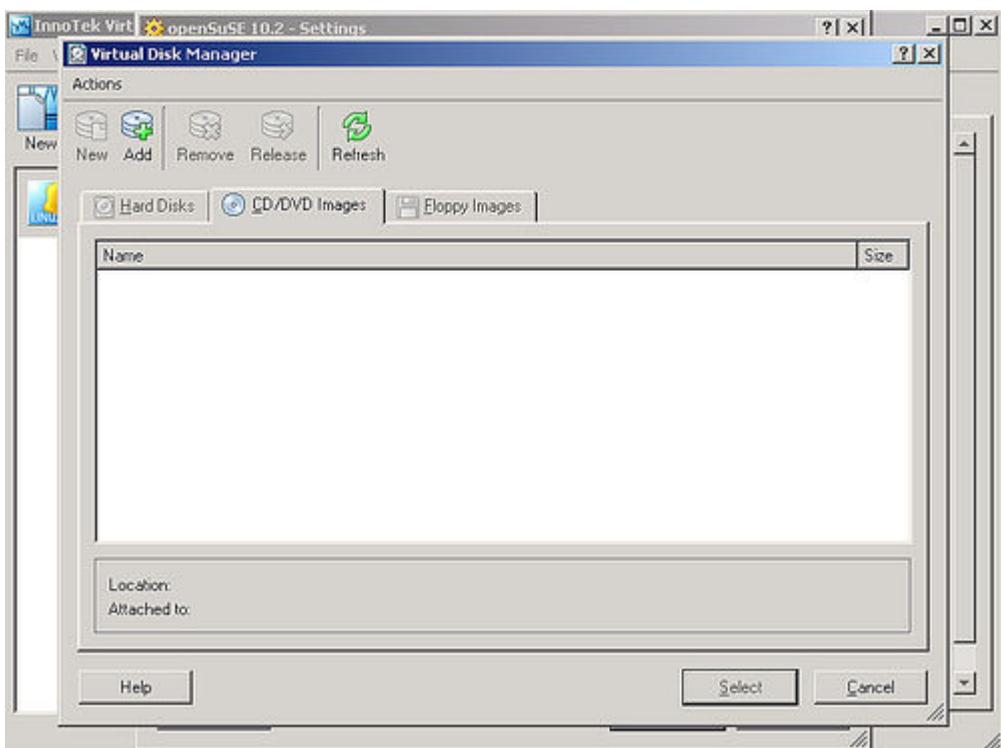


fig. 8

Clicchiamo sull''icona *Add* in alto a sinistra nella barra degli strumenti, si aprirà questa finestra (fig. 9):

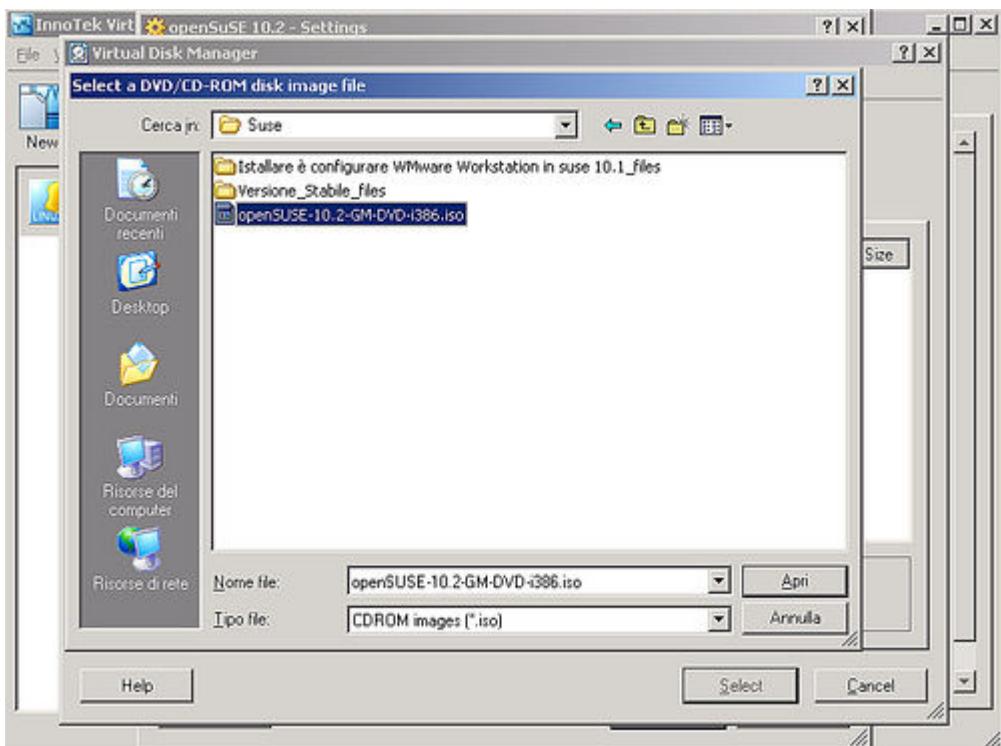


fig. 9

Cercate fra le vostre cartelle il file .iso e selezionatelo, poi cliccate su *Apri*.

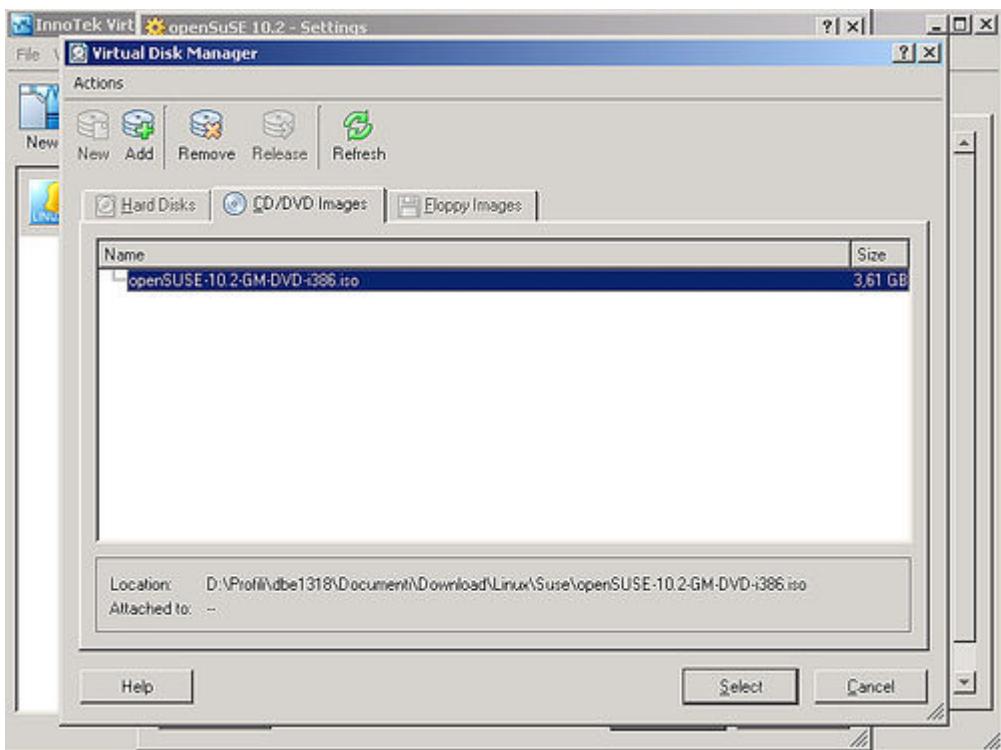


fig. 10

Il file viene reso disponibile per il caricamento sul vostro DVD virtuale, cliccateci sopra (**fig. 10**) e poi sul pulsante *Select* (se come è capitato a me, anche voi non riuscite a cliccarci sopra per selezionarlo, cliccate prima sul pulsante *Refresh* in alto nella barra degli strumenti).

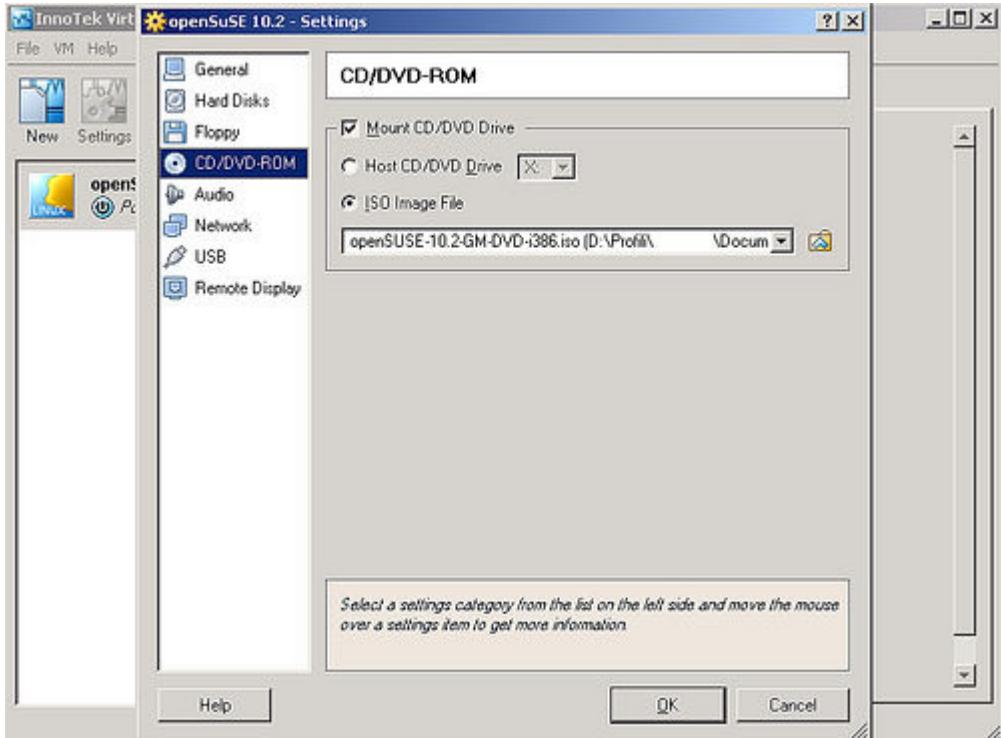


fig. 11

La finestra del *Virtual Disk Manager* (VDM) si chiuderà e torneremo alla finestra dei settaggi (**fig. 11**) dove il CD/DVD-ROM ora ha caricato la iso di Linux.

Clicchiamo ora sulla voce *Audio* (**fig. 12**) e configuroate la sezione audio del vostro PC virtuale come indicato in figura:

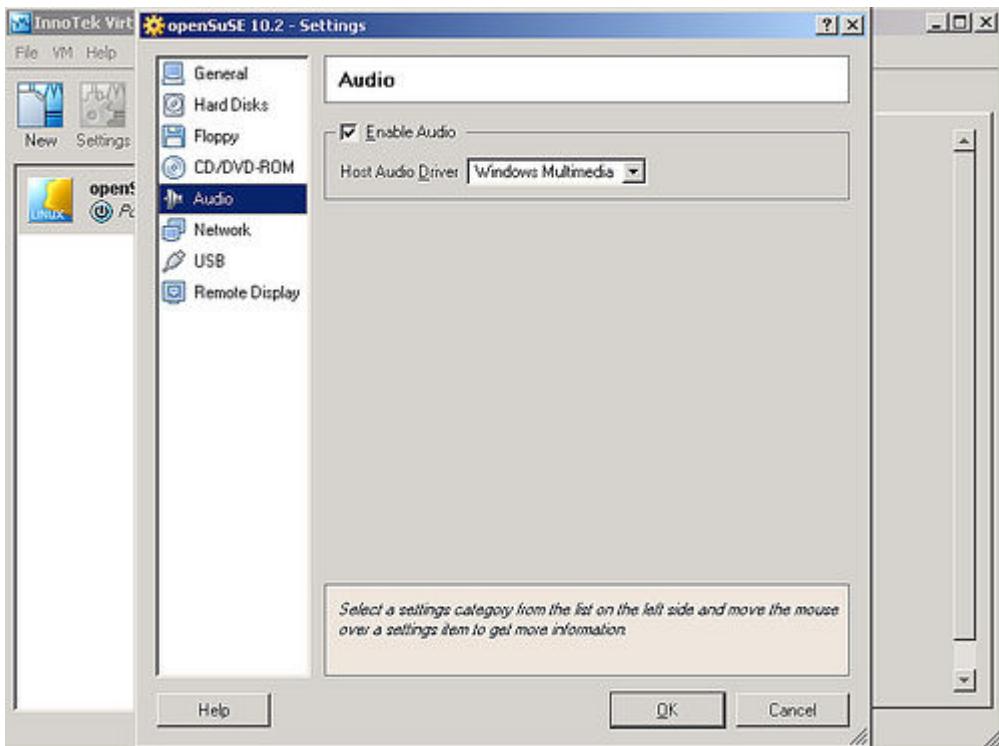


fig. 12

Saltiamo invece a pié pari la configurazione della rete (la voce *Network*), le pre-impostazioni dovranno andar bene così come sono, e passiamo a **configurare** l'ultima cosa di cui ci occupiamo, **le porte USB**.

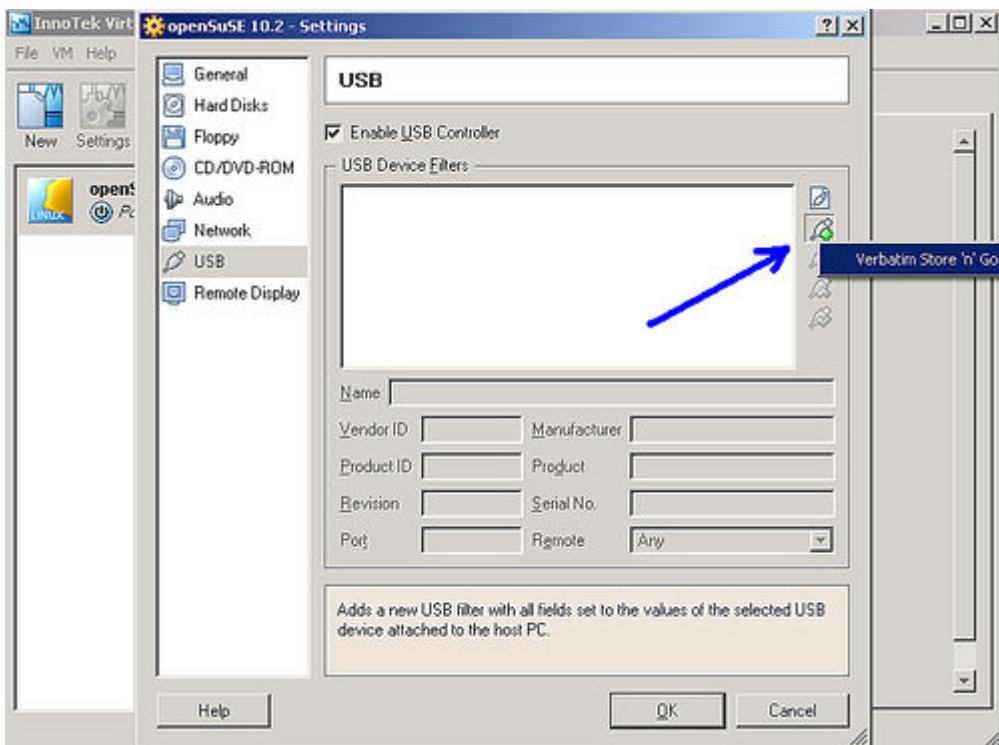


fig. 13

Clicchiamo sulla voce *USB* e spuntiamo *Enable USB Controller*, poi clicchiamo su dove indicato dalla freccia (fig. 13), comparirà l'elenco delle periferiche USB collegate in quel momento sul computer reale che potremo aggiungere alla nostra macchina virtuale. Quella che vedete a titolo d'esempio è la mia penna USB, cliccandoci sopra questa è stata aggiunta al mio sistema virtuale (fig. 14).

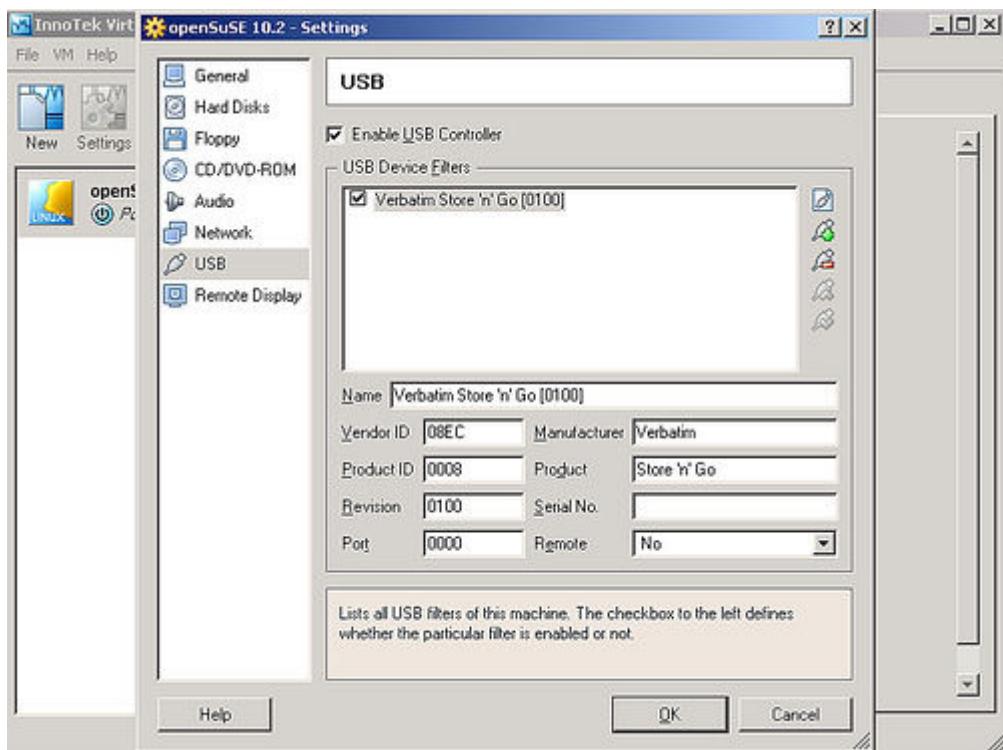


fig. 14

Voglio sottolineare una cosa: la lista delle periferiche USB che potete selezionare, sono quelle effettivamente collegate in quel momento alla vostra macchina. Se volete che la vostra penna USB compaia nella lista, dovete collegarla al vostro PC reale prima di avviare VirtualBox. Maggiori informazioni le trovate nel post "VirtualBox: come configurare le porte USB".

Clicchiamo sul pulsante OK.

Bene, abbiamo finalmente concluso il setup della macchina virtuale (fig 15).

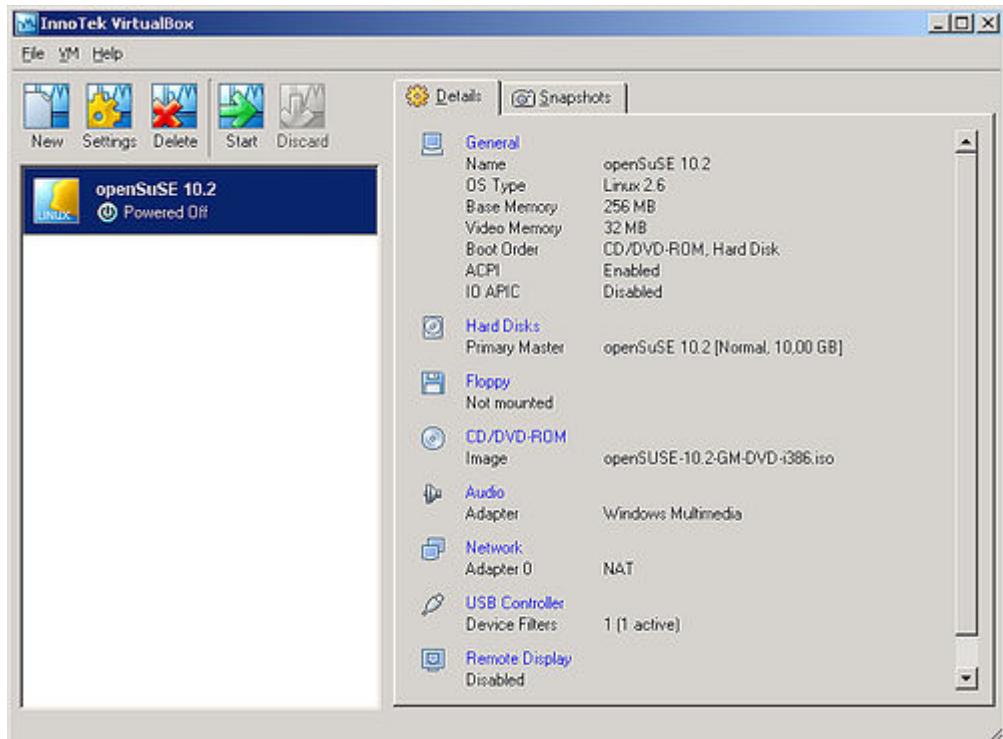


fig. 15

Non ci resta che provare ad avviare la macchina virtuale. Se avete scelto l'installazione da DVD mettete prima il DVD contenente Linux nel lettore fisico, selezioniamo ora la macchina virtuale e clicchiamo sull'icona *Start* nel menu strumenti:

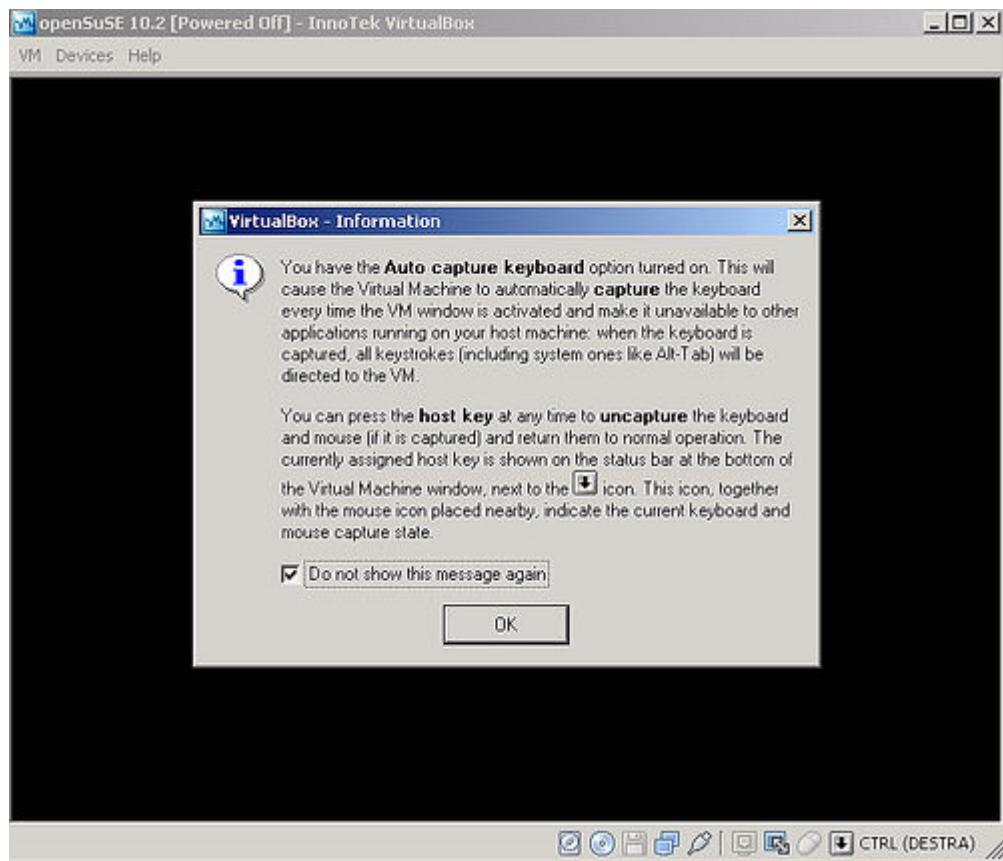


fig 16

Prima di partire compare il box informativo di **fig. 16** che ci avvisa di fare attenzione a quando clicchiamo all'interno della finestra della macchina virtuale. Il mouse e la tastiera verranno "catturati" all'interno della macchina virtuale e per uscire occorre premere prima il tasto Control di destra della tastiera (se guardate bene la barra degli strumenti in basso nella finestra c'è anche indicato in grande CTRL (DESTRA)).

Ulteriori informazioni su questo particolare comportamento del mouse potete trovarle nel post "VirtualBox: liberare il mouse intrappolato in una finestra".

Intanto proseguiamo: cliccato su **OK** del box informativo partirà finalmente la nostra macchina virtuale facendo il boot nell'ordine che avevamo indicato (ricordate? Prima il floppy, poi il CD/DVD, infine il disco rigido). Trovato il DVD o il file .iso a seconda delle scelte fatte in precedenza, partirà l'installazione che vedete in **fig 17**.

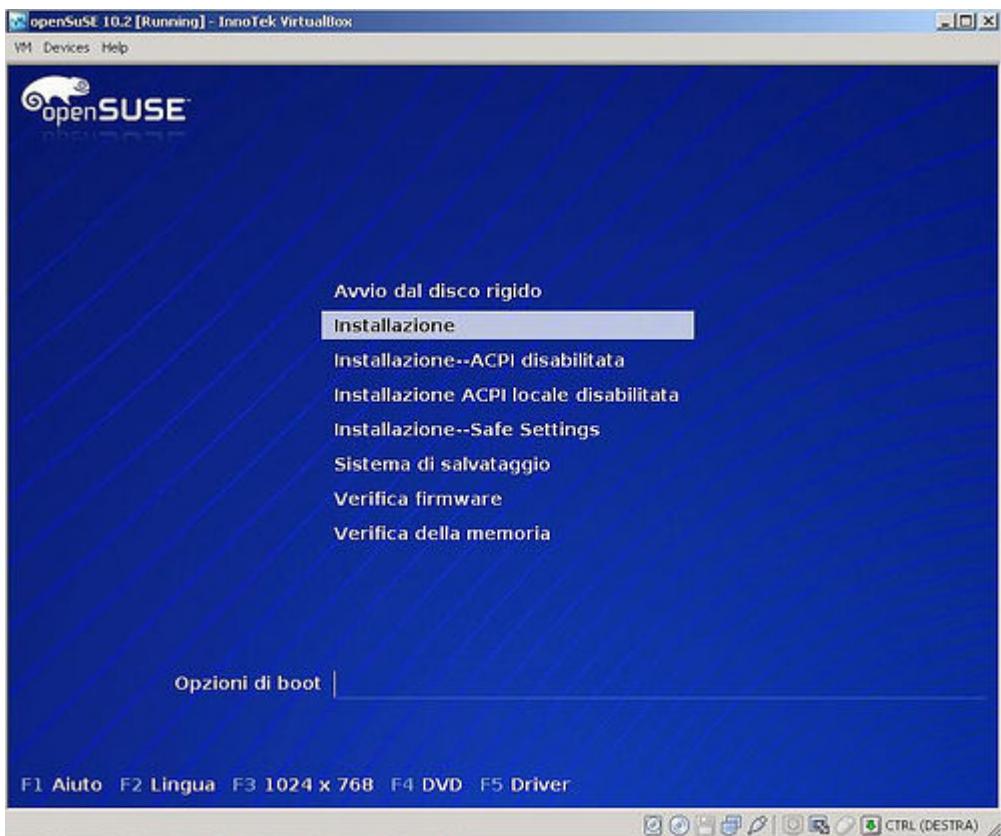


fig. 17

Ed eccoci finalmente: sta partendo l'installazione di SuSE Linux 10.2 (o comunque della vostra distribuzione preferita, lo ricordo, i passaggi per creare la macchina virtuale sono identici qualunque sia la distribuzione)!

A questo punto tutto procede come una qualunque installazione su una macchina reale ma non dovremo aver paura di far danni se sbagliamo qualcosa: l'installazione di Windows è al sicuro sulla macchina reale, quindi potremo smanettare in piena libertà su Linux ed imparare ad usarlo al meglio.

Se dovete incontrare qualche problema con il PC virtuale, vi segnalo altri post che vi possono aiutare. Tenete d'occhio la lista, è continuamente aggiornata con i nuovi post sul fondo:

- Virtualizzazione: creiamo un PC virtuale con VirtualBox (1a parte)
- VirtualBox: come aggiornarlo ad una nuova versione
- VirtualBox: come configurare le porte USB
- VirtualBox: installare le VirtualBox Guest Additions, premessa
- VirtualBox: installare le VirtualBox Guest Additions, procedura generale
- Rilasciato Virtualbox 1.4.0: tutte le novità

Bene, questo è tutto, non mi rimane che augurarvi buon divertimento con i vostri esperimenti di virtualizzazione.

@:\>

Pubblicato alle 16.58 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

2 commenti:

Anonimo ha detto...

Grazie a questa splendida guida sono riuscito a installare UBUNTU 7.04 Funziona tutto alla grande tranne le porte USB, non vengono riconosciute in fase di installazione. Mi puoi dire in base alla tua esperienza cosa posso fare? Ti ringrazio ciao
Luciano

16 maggio 2007 21.59
Giornale di Sistema ha detto...

Ciao Luciano.
Grazie, sono contento ti sia piaciuta! :)

La Ubuntu 7.04 per il momento ho avuto solo il tempo di scaricarla... appena ho la possibilità la installo e ti faccio sapere com'è andata.

A presto, ciao ciao.
@:>

giovedì 15 marzo 2007

Nuova versione per VirtualBox: la 1.3.8

Volevo segnalare l'uscita della nuova versione di VirtualBox: la 1.3.8.

Sono stati risolti alcuni bug e la versione Host Linux (cioè da installare su Linux) conta su un supporto per un maggior numero di versioni di Linux.

La lista completa delle novità sul sito di VirtualBox.

Buon aggiornamento.

@:\>

Pubblicato alle 18.14 Argomenti: news, software, Virtualbox, virtualizzazione

venerdì 16 marzo 2007

VirtualBox: come aggiornarlo ad una nuova versione

Aggiornamento del 5 settembre 2007

Il post, aggiornato nella procedura, resta valido anche nelle versioni successive di VirtualBox.

Nel caso riutilizzate macchine virtuali create con precedenti versioni di VirtualBox, ricordatevi di aggiornare anche le Guest Additions al loro interno (a breve il link al post).

@:>

Questo post prende spunto dalla notizia che riportavo ieri dell'uscita della nuova versione di VirtualBox, la 1.3.8.

Almeno per ora non esiste la possibilità di aggiornare VirtualBox direttamente dalla vecchia versione, occorre prima disinstallarlo e poi reinstallarlo. La procedura non differisce comunque dalla classica disinstallazione di qualunque programma Windows, vediamola nel dettaglio.

Prima di procedere con la disinstallazione, **salviamo le macchine virtuali già create** (non si sa mai che qualcosa vada male).

Aprite l'Esplora risorse e cercate la cartella C:\Documents and Settings\Nome Utente\VirtualBox: è qui che vengono salvate le macchine virtuali (cartella VDI) e le loro configurazioni (cartella Machines). Fate una copia di tutta la cartella .VirtualBox.

Bene, siamo pronti per **disinstallare VirtualBox**.

Andate nel *Pannello di Controllo*, cliccate su *Installazione Applicazioni e Cambia* (**fig. 1**).

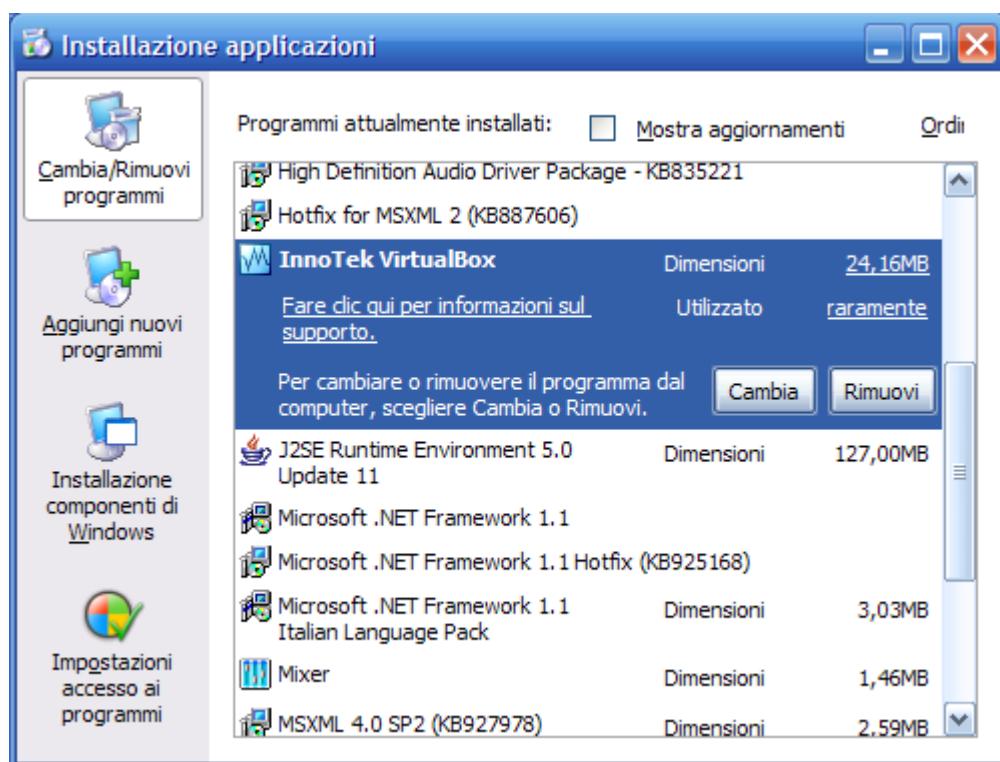


fig. 1

Partirà la disinstallazione di VirtualBox (**fig. 2**).



fig. 2

Cliccate su *Remove* per procedere con la disinstallazione. Vi verrà richiesto per conferma se volete rimuovere VirtualBox, cliccate ancora su *Remove*(fig. 3).

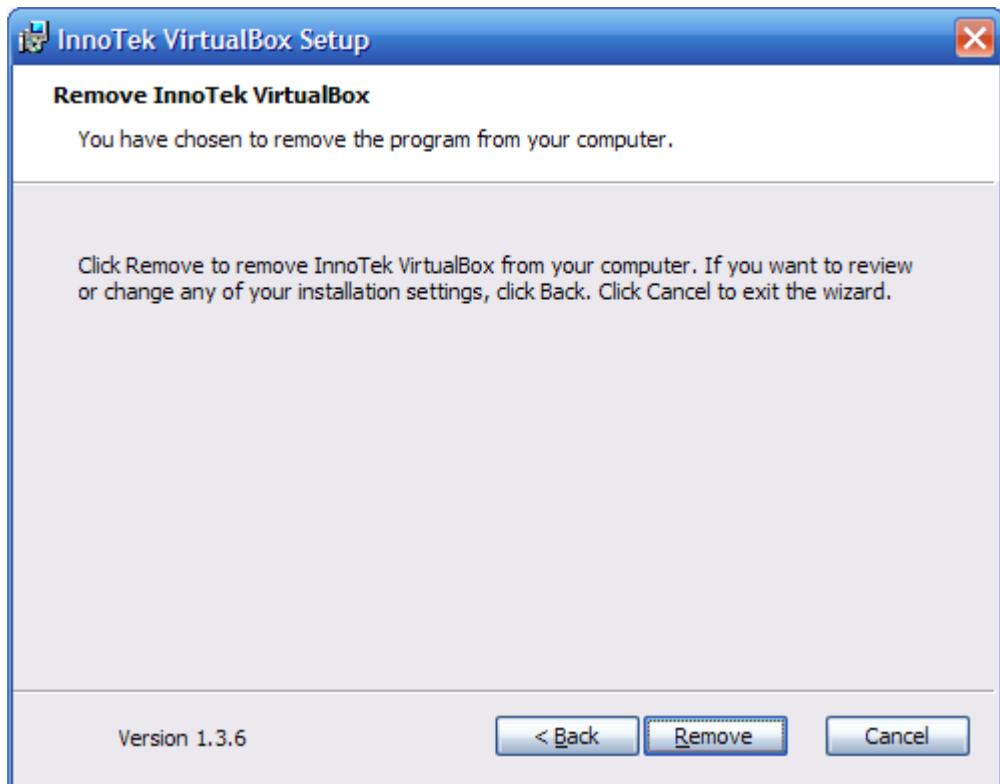


fig. 3

Partirà finalmente la disinstallazione, quando sarà conclusa cliccate su *Finish* (fig. 4).



fig. 4

La disinstallazione non cancella la cartella .VirtualBox, per cui le nostre macchine virtuali sono al sicuro. Vi verrà chiesto di riavviare Windows per concludere correttamente la disinstallazione, chiudete tutte le applicazioni eventualmente aperte e riavviate cliccando su Yes (fig. 5).

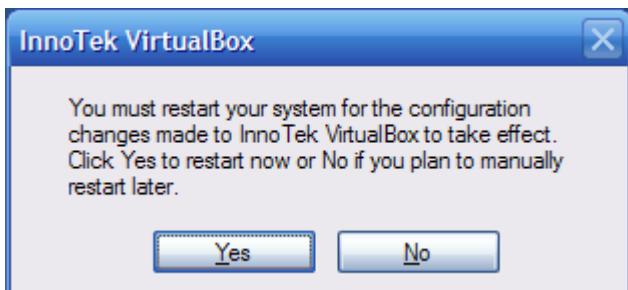


fig. 5

Bene, il tempo di riavviare il vostro PC e sarete pronti a reinstallare la nuova versione di VirtualBox nel modo che abbiamo già visto in un precedente post (Virtualizzazione: installiamo Virtualbox).

@:>

Pubblicato alle 16.28 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione,

martedì 24 aprile 2007

VirtualBox for OS X

Ottime notizie dal fronte VirtualBox: il programma di virtualizzazione open source adesso è disponibile anche per Mac OSX in versione beta.

@:|>

Pubblicato alle 12.01 Argomenti: news, Virtualbox, virtualizzazione

lunedì 4 giugno 2007

VirtualBox: come configurare le porte USB

E rieccoci, dopo un pò di tempo, a parlare nuovamente di VirtualBox.

Questa volta vi spiegherò **come configurare le porte USB**, un problema più comune di quanto si possa credere (colgo l'occasione per ringraziare Luciano che ha posto per tutti la domanda in questo commento).

E vediamo come fare.

La soluzione è valida per qualunque distribuzione Linux vogliate virtualizzare perché non è Linux a non vedere le porte USB, è nella configurazione hardware del PC virtuale che avete creato in VirtualBox che semplicemente non esistono, magari perché vi siete dimenticati di attivarle. Lo sottolineo ancora: Linux non riconoscerà mai e poi mai le porte USB se queste non sono già presenti e attive nella configurazione hardware del PC virtuale.

Non dimentichiamoci infatti una cosa basilare. VirtualBox è un programma che serve a creare un intero PC virtuale, con tutte le componenti hardware che costituiscono qualunque PC reale. Ci sarà pertanto un disco rigido, anche se solo virtuale, una scheda video, virtuale anch'essa, quella audio, e tra le altre cose, anche una o più porte USB; ma solo se le avremo esplicitamente previste mentre stavamo creando il PC virtuale, o in alternativa, le abbiamo comunque aggiunte successivamente. Perciò, prima attivate le porte USB nella macchina virtuale e solo poi potrete aspettarvi che Linux le riconosca.

Vediamo adesso come VirtualBox gestisce le porte USB reali per poi renderle disponibili nel PC virtuale. VirtualBox in realtà non riconosce le porte USB del vostro PC reale! Sorpresi? Ciò che riconosce invece sono i singoli dispositivi USB che inserite in esse, ad esempio la vostra chiavetta.

VirtualBox si accorge della sua esistenza solo quando la inserite nel PC reale e la rende disponibile nel PC virtuale, qualunque sia la porta USB reale fra quelle a disposizione a cui la collegate. Se la chiavetta la prossima volta la inserite su una porta diversa, si tratta sempre e comunque di quella chiavetta. Basta configurarla una e una sola volta in VirtualBox, la prima, e poi sarà sempre riconosciuta qualunque sia la porta in cui la inseriamo. E così l'eventuale vostro disco rigido esterno USB, la stampante USB, e così via, indifferenemente da quale porta sul PC reale state usando.

Scoperto l'arcano, non ci resta allora che avvisare VirtualBox dell'esistenza della nostra chiavetta USB. Come? Semplicemente inserendola nel vostro PC reale mentre VirtualBox non gira. Appena la chiavetta viene riconosciuta da Windows, avviate VirtualBox. Se non avete già una macchina virtuale pronta, createla seguendo le istruzioni del post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 1)".

Ora sia che si tratti del PC virtuale appena creato, sia si tratti di quello con Linux già installato e da fargli riconoscere solo i dispositivi USB, passate a configurare la vostra chiavetta USB così come indicato nel post "Virtualizzazione: creiamo un PC virtuale con VirtualBox (parte 3a)".

Bene, ora chiudete VirtualBox e sfilate la chiavetta USB dal PC reale.

Ora, solo se si tratta della macchina virtuale appena creata, riavviate VirtualBox e procedete con l'installazione di Linux nella distribuzione da voi preferita. Quando è terminata e avete Linux perfettamente funzionante davanti a voi, spegnete Linux facendo lo Shut Down, poi chiudete VirtualBox.

Siamo pronti!

Riavviate VirtualBox e avviate in esso la macchina virtuale. Quando la partenza è terminata e siete davanti al desktop di Linux, infilate la vostra chiavetta USB nel PC reale: solo la prima volta, partirà l'installazione completamente automatica di un driver *VirtualBox USB*. E' questione di pochi secondi: appena terminata, dopo qualche secondo ancora, comparirà finalmente sul desktop la chiavetta USB.

Ora che tutto funziona, per essere sicuri che la chiavetta USB venga sempre rilevata al suo inserimento, seguite questa procedura:

- prima avviate VirtualBox,
- poi avviate il PC virtuale,
- solo quando il sistema operativo installato al suo interno ha completato la partenza, inserite il dispositivo USB nella porta del PC reale.

Giusto per completezza, riporto sotto gli screenshot di come appare la mia chiavetta USB sotto Ubuntu 7.04 (**fig. 1**) e Suse 10.2 (**fig. 2**).

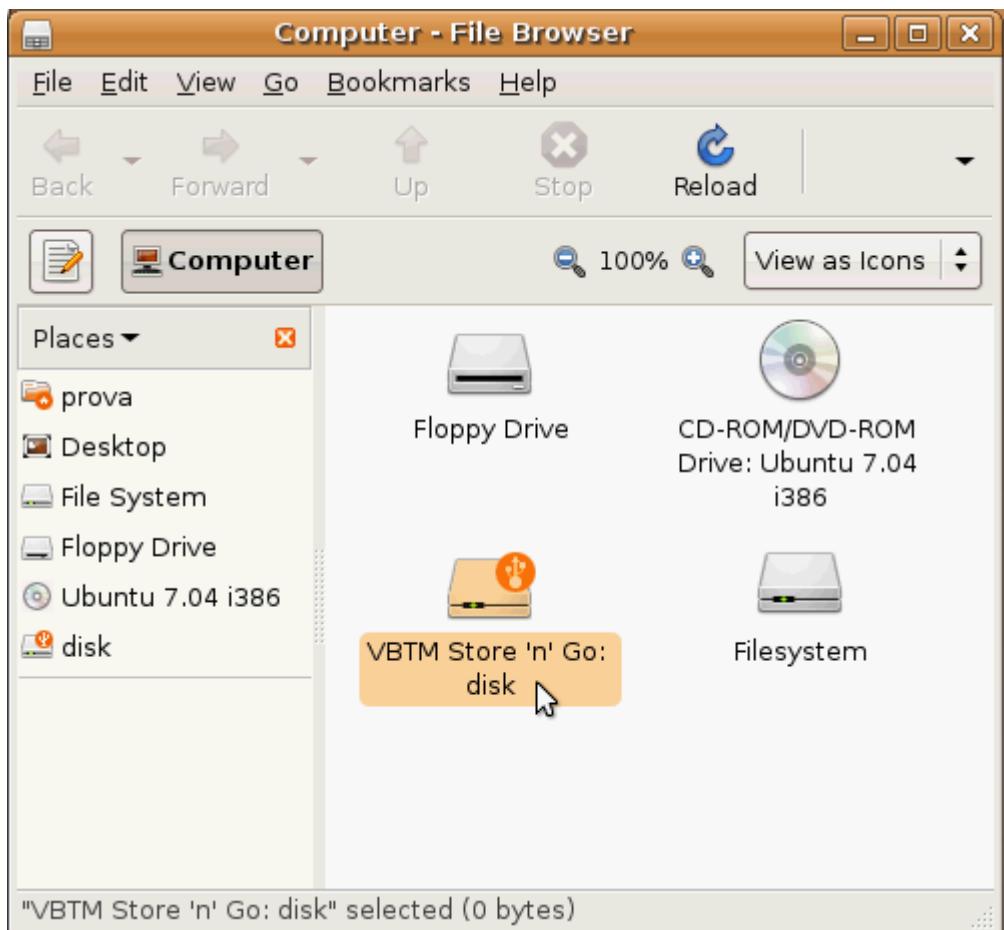


fig. 1

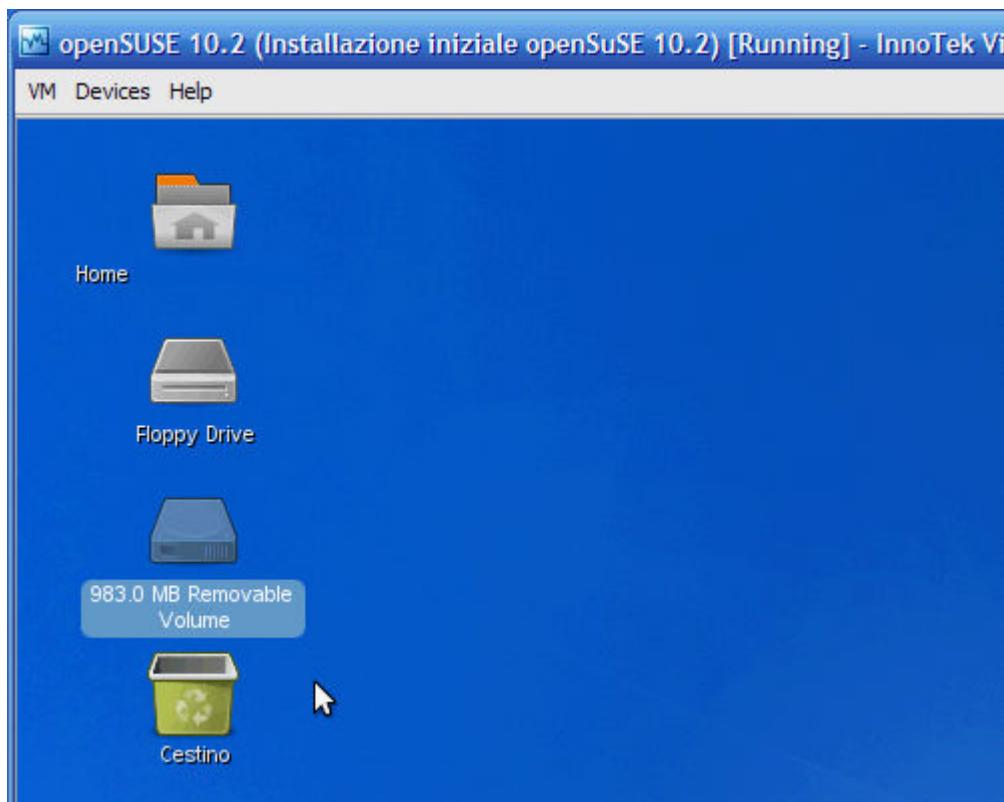


fig. 2

E' tutto. Spero di avervi aiutato a dissipare ogni dubbio in merito alla configurazione di dispositivi USB in VirtualBox, e se avete problemi ancora problemi, non mancate di segnalarceli.

@:)>

Aggiornamento di: domenica 10 giugno 2007

Volevo segnalarvi un post di Gianbi che spiega come ha risolto il problema del riconoscimento della chiavetta sotto Ubuntu Linux.

Grazie Gianbi. :)

Pubblicato alle 12.44 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

martedì 5 giugno 2007

VirtualBox: liberare il mouse intrappolato in una finestra

Una cosa che disorienta facilmente chi usa per la prima volta VirtualBox sono l'uso del mouse e della tastiera.

E' un classico: si clicca all'interno della finestra del PC virtuale, ed il mouse in un certo senso vi "casca dentro", rimane cioè intrappolato all'interno della finestra e non si riesce più a farlo uscire per poter interagire con il resto del PC reale.

Come fare allora a tirare fuori il mouse dalla finestra in cui è rimasto intrappolato?

Premete il tasto Ctrl (control) destro della vostra tastiera. Vedrete il mouse nella finestra del PC virtuale rimanere congelato nella sua posizione, mentre al di fuori di essa compare il mouse del PC reale con cui ora potete tranquillamente interagire (**fig. 1**). Se osservate bene in basso a destra di fig. 1 noterete che VirtualBox stesso indica come fare seppur con un sibilino *CTRL (DESTRA)*. Provate anche a lasciare il mouse sulla scritta e sulle icone accanto. Compariranno dei tooltip con ulteriori informazioni sul loro significato.



fig. 1

Se volete, potete fare in modo che sia solo la tastiera ad essere attiva nella finestra del PC virtuale, senza farvi "cascare dentro" il mouse (**fig. 2**).

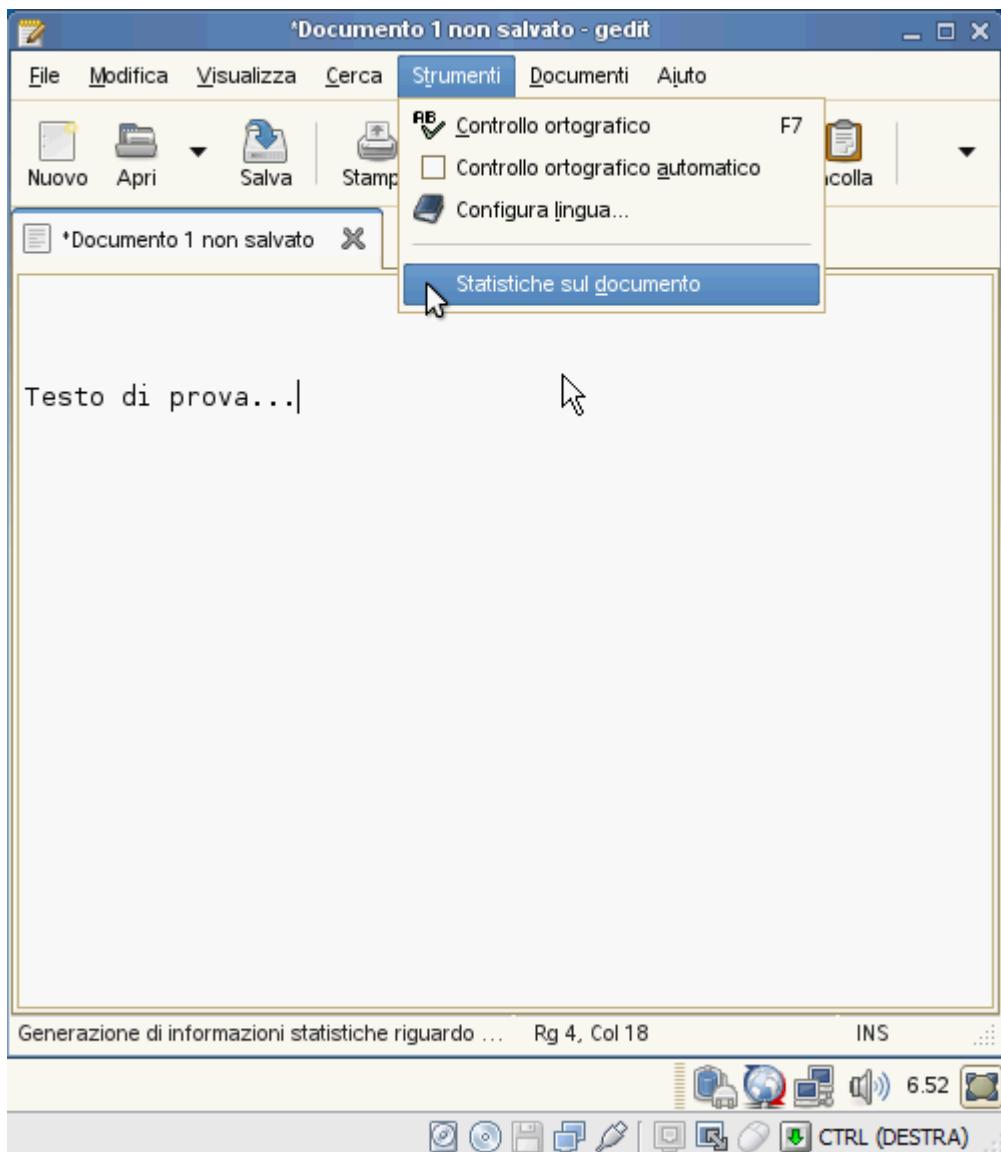


fig. 2

Qui però qualche dubbio sull'esatta procedura da seguire è rimasto. A leggere il manuale è sufficiente cliccare sulla barra del titolo della finestra del PC virtuale, ma all'atto pratico, spesso ho dovuto cliccare sull'icona della finestra che compare nella barra delle applicazioni di Windows (**fig. 3**) altrimenti la tastiera non si attivava. Se **fig. 3** vi confonde un attimo, tenete presente che ho la barra della applicazione sul lato destro dello schermo, in verticale.

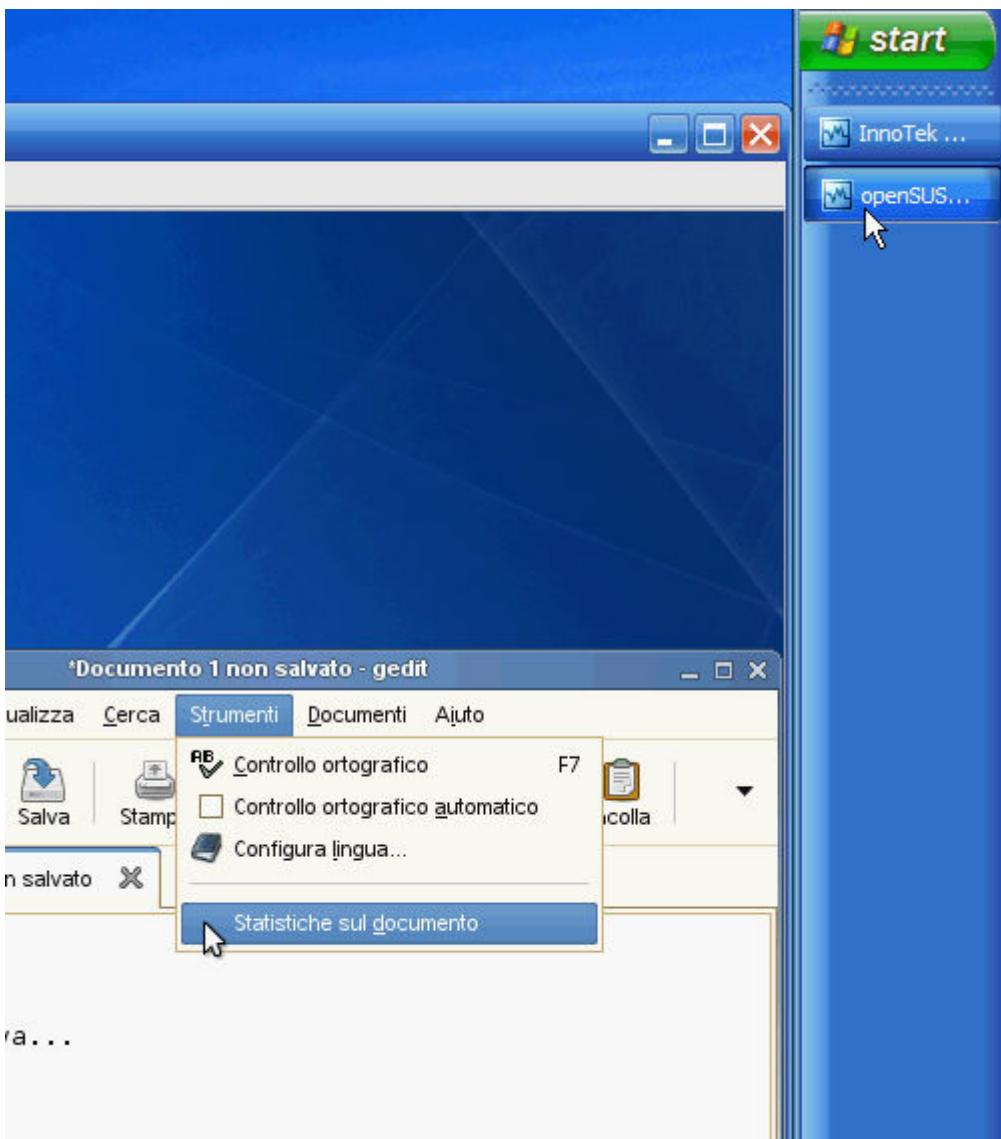


fig. 3

Provate anche voi a cliccare sia sulla barra del titolo della finestra del PC virtuale che sulla sua icona sulla barra delle applicazioni e fatemi sapere quale dei 2 sistemi di attivazione della sola tastiera funziona nel vostro caso.

Esiste in realtà un modo per avere un unico puntatore del mouse che non resti imprigionato nella finestra del PC virtuale: occorre installare le *VirtualBox Guest Additions*, delle speciali utility che aumentano le performance e l'usabilità di VirtualBox. Oltre infatti a risolvere il problema del mouse intrappolato, aumentano per esempio le performance ed il numero di risoluzioni a disposizione nella scheda video virtuale, permettono di condividere cartelle fra il PC virtuale e quello reale e altre cosette interessanti.

Si tratta di un argomento così succoso, l'installazione delle *VirtualBox Guest Additions*, che non mancherò di parlarvene in questo apposito post e nei successivi.

@:)>

Pubblicato alle 9.55 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

mercoledì 6 giugno 2007

VirtualBox: installare le VirtualBox Guest Additions, premessa

Parafrasando la nota pubblicità che dice "la potenza è nulla senza controllo", si può dire che VirtualBox è nulla senza le VirtualBox Guest Additions installate.

In breve, si tratta di driver e applicazioni di sistema da installare nella macchina virtuale che ne migliorano le performance e l'usabilità. Vediamo i punti più significativi:

- *integrazione fra mouse del PC reale e quello della macchina virtuale*. Ne parlavo nel post "VirtualBox: liberare il mouse intrappolato in una finestra", cliccando nella finestra della macchina virtuale, il mouse resta intrappolato al suo interno. Con le VirtualBox Guest Additions installate questo fastidioso comportamento scompare, ed il mouse interagisce con la finestra del PC virtuale come con qualunque altra finestra appartenente al PC reale.

- *miglioramento delle performance video*. Si possono usare più colori e risoluzione maggiori di quelle standard. Sul mio portatile ad esempio, sono passato da 1024x768 a 1280x800 con un miglior sfruttamento dello spazio a disposizione sullo schermo, ma ho disponibili risoluzione fino a 3200x2400 con 16 milioni di colori.

- *condivisione delle cartelle*. Diviene finalmente possibile condividere cartelle tra PC virtuale e PC reale con i vantaggi e la comodità operativa che questo comporta.

- *condivisione degli appunti*. Potrete usare il normale copia-incolla anche tra sistema host e sistema guest, cioè tra PC reale e virtuale.

A questo punto non ci resta che vedere come installarle nella vostra macchina virtuale con la solita guida per immagini.

Prima però una doverosa premessa. Le VirtualBox Guest Additions vengono installate nella macchina virtuale, pertanto, non esiste una singola procedura di installazione uguale per tutte le distribuzioni Linux che è possibile virtualizzare. Esiste piuttosto una procedura generale che è da adattare di volta in volta alla particolare distribuzione che abbiamo virtualizzato. Quello che farò allora è prima illustrare questa procedura generale, e poi, come questa è svolta in concreto in una serie di macchine virtuali di esempio, la prima delle quali sarà una openSUSE, ma poi via via anche altre.

Tenete inoltre presente un'altra cosa: è possibile che siate utenti Windows alle prime armi con Linux e anzi, usiate la virtualizzazione proprio per poter provare Linux senza dover sfasciare un PC fra dual boot, partizionamento del disco rigido e possibili perdite di dati che ne può derivare se qualcosa va male.

Ebbene, se siate utenti Windows alle prime armi con Linux, potrete avere qualche difficoltà in più: Linux infatti non è così immediato come Windows e non perché sia fatto male, anzi. In Linux la sicurezza viene prima di tutto, fino al punto da sacrificare anche un pò di usabilità. Vi troverete così a trafficare con password di accesso, diritti amministrativi, utente root, sudo (il comando, non l'eccesso di traspirazione), la riga di comando, tutte cose nuove e misteriose per il tipico utente "home" di Windows.

Tutto questo ben lungi dal volervi scoraggiare, è un invito invece a seguire con attenzione i prossimi post. Vi illustrerò passo passo come procedere e grazie anche alle immagini le difficoltà dovrebbero essere davvero minime.

Accendete le vostre macchine virtuali allora, stiamo per iniziare.

@:\>

Pubblicato alle 10.18 Argomenti: guida, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione

giovedì 7 giugno 2007

VirtualBox: Guest Additions in Linux, procedura generale

Dopo la presentazione delle VirtualBox Guest Additions nel precedente post "VirtualBox: installare le VirtualBox Guest Additions, premessa", vediamo adesso la procedura generale da seguire per la loro installazione su una generica macchina virtuale Linux, in successivi post vedremo come questa si svolge concretamente su alcune distribuzioni; su macchine Windows invece la procedura è unica.

Vi ricordo infatti che le Virtual Guest Additions vengono installate all'interno del PC virtuale, non su quello reale.

L'installazione è in tre punti:

1 - attivazione al supporto dei moduli di kernel esterni. Questo perché le VirtualBox Guest Additions per funzionare hanno bisogno di un loro modulo di kernel. Grazie a questa attivazione, da fare lo ricordo ancora nel Linux all'interno della macchina virtuale, questi sarà in grado di costruire e usare questo modulo esterno.

I moduli kernel da installare per alcune distribuzioni sono:

- per le distribuzioni Debian e Ubuntu più nuove, le versione più recenti di *linux-headers* e se esiste *linux-kbuild*;
- per le distribuzioni Debian e Ubuntu più datate, *kernel-headers*;
- per le distribuzioni Fedora e Redhat, *kernel-devel*;
- per le distribuzioni SUSE e OpenSUSE, le versione più recenti di *kernel-source* e *kernel-syms*.

Vi servirà inoltre, se non è già presente, il compilatore *GNU compiler* per il linguaggio C, per poter procedere alla compilazione di questi moduli dal loro sorgente.

2 - installazione delle VirtualBox Guest Additions: in pratica viene solo fatto il mount del file *VBoxGuestadditions.iso*. Per chi viene da Windows ed è all'oscuro di cose come il *mount di un'unità*, potete immaginarlo come l'analogo dell'inserire una chiavetta USB nel vostro PC. Windows la riconosce, le assegna una lettera di unità e la visualizza in risorse del computer. Ecco, Windows non ha fatto altro che il mount, o montaggio, dell'unità. Fatto il mount del file *VBoxGuestadditions.iso*, questo comparirà sul desktop di Linux come una nuova unità CD-ROM.

3 - compilazione delle VirtualBox Guest Additions: da utente root (l'utente con i massimi privilegi sul sistema), dopo essersi spostati sull'unità appena montata, occorre dare il comando:

```
sh ./VBoxLinuxAdditions.run
```

Le distribuzioni Linux ufficialmente supportate sono:

- Fedora Core 4 e 5;
- Redhat Enterprise Linux 3 e 4;
- Suse Linux 9 e 10;
- Ubuntu 5.10 e 6.06

e tutte le distribuzioni che usano release equivalenti del software.

Per ora ci fermiamo qui.

Nei post "VirtualBox: installare le Guest Additions in SUSE, fase 1", "fase 2" e "fase 3" vedremo come questa procedura generale d'installazione delle VirtualBox Guest Additions, si concretizza su una distribuzione SUSE Linux 10.2, quali imprevisti ci si può ritrovare a fronteggiare, e la relativa soluzione.

@:\>

Pubblicato alle 13.10 Argomenti: guida, Linux, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione

3 commenti:

Michele Benvegnù ha detto...

Complimenti per la competenza di "Giornale di sistema"; approfitto per 2 quesiti.

Ho installato VirtualBox 1.4 su host Vista Business, ed ho installato Ubuntu come sistema guest. Ho installato le VBoxLinuxAdditions, che funzionano correttamente (lo conferma il comportamento del mouse, che non necessita

di essere "catturato"). Tuttavia la risoluzione schermo non va oltre le classiche 640x480, 800x600 e 1024x768. Dove sbaglio? Cosa devo fare o impostare per arrivare a 1280x1024?

Ho condiviso una cartella della macchina host (sudo mount -t vboxsf nomecondivisione puntodimount) e vedo correttamente le cartelle di Windows, ma quando spengo e riavvio Ubuntu la condivisione si perde e devo ridare il comando. Come fare per mantenere la condivisione anche al successivo riavvio? (una specie di aurorun di Windows!).

Ringrazio chi saprà rispondermi (in termini semplici perchè di linix mastico poco)

21 giugno 2007 22.59

Giornale di Sistema ha detto...

Ciao Michele, grazie per i complimenti. :-D

E colgo l'occasione per rivolgermi anche a tutti gli altri lettori: se avete già incontrato il problema di Michele e siete riusciti a risolverlo con successo, avete voglia di condividere la vostra soluzione?

Grazie in anticipo! :)

@:\>

22 giugno 2007 19.11

Giornale di Sistema ha detto...

Ciao Michele.

A proposito del tuo primo quesito, non so se hai già risolto.

Se sì, cos'era? sono curioso... :)

Se no, allora mi è venuto un dubbio: quanta memoria video hai assegnato al sistema guest? Perché se è troppo poca, risiede lì il motivo della risoluzione così bassa.

Di default VirtualBox assegna 8 MB di Ram alla scheda video, proprio ad andarci al risparmio, raddoppiala. Sennò non lesinare, dai 32 a 64 MB trova il valore che è migliore per te.

Ciao ciao!

@:\>

venerdì 8 giugno 2007

Rilasciato Virtualbox 1.4.0: tutte le novità

Sono sempre affascinato dalle sincronicità, nome molto più significativo di coincidenza. Proprio l'altro giorno cercavo di installare in VirtualBox la versione 7 di Fedora, la distribuzione Linux di casa Redhat. Niente da fare, l'installazione non andava a buon fine. Ricordo che ho pensato: "beh... sarebbe quasi ora di una nuova versione di VirtualBox che supporti anche Fedora 7, e magari aggiunga qualche altra gustosa novità alle tante cose buone che già offre".

Ebbene: detto, fatto! E' uscita la nuova versione di VirtualBox, la 1.4.0 (fig. 1).

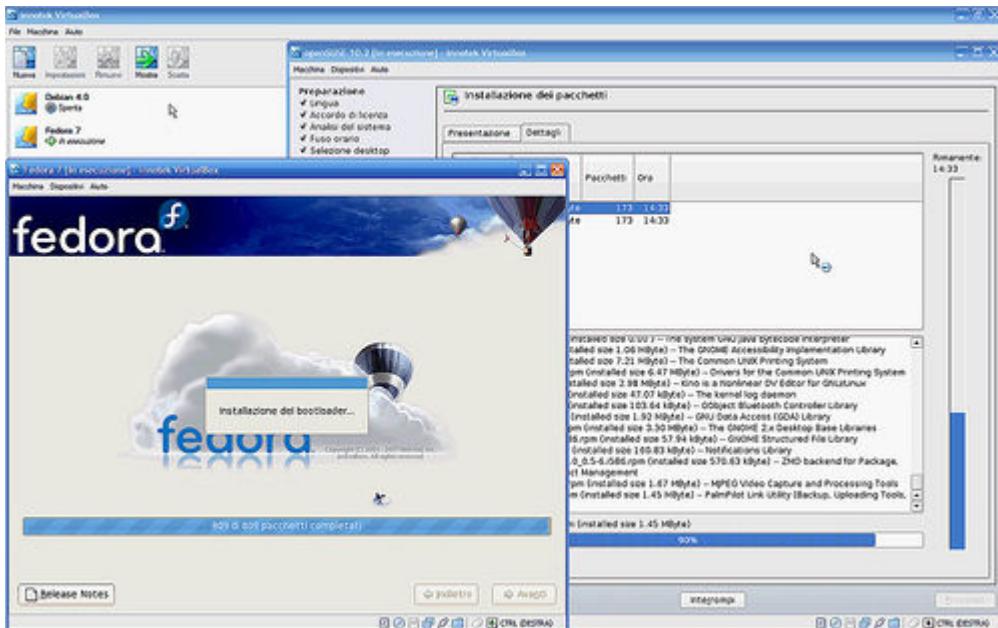


fig. 1

Per ingrandire l'immagine e vedere la novità più eclatante di VirtualBox 1.4.0, cliccateci sopra e poi su *ALL SIZES* (oppure sull'icona della lente di ingrandimento), infine su *LARGE*: non notate nulla?

Si ok, preso dall'entusiasmo ho esagerato! In primo piano sul lato sinistro dell'immagine sta procedendo l'installazione di Fedora 7, a destra in secondo piano quella di openSUSE 10.2, e in alto a sinistra, sotto tutte le altre, la finestra principale di VirtualBox.

Ebbene, come potete vedere, VirtualBox, ora è in italiano!

Ma questa è solo la prima di una numerosa ed interessante serie di novità, la lista completa come al solito la trovate qui nel Changelog. Vediamone subito altre:

- è in italiano, compresa la procedura d'installazione, ma questo si era già detto;
- ora è possibile fare il copia-incolla fra guest e host, cioè fra PC virtuale e reale, e viceversa;
- è stata aggiunta l'interfaccia grafica per impostare la condivisione delle cartelle fra host e guest;
- è stato aggiunto il supporto per OS X e AMD 64;
- un miglioramento generale delle prestazioni.

Se avete consultato il Changelog vi sarete accorti che questi sono solo alcuni dei miglioramenti, delle aggiunte o correzioni apportate.

Una cosa però continua a mancare: la possibilità di fare l'aggiornamento dell'installazione preesistente. Occorre invece disinstallare la versione precedente come indicato nel post "VirtualBox: come aggiornarlo ad una nuova versione" e procedere poi con la nuova installazione. Niente paura però: la disinstallazione elimina solo VirtualBox, le macchine virtuali già create restano al loro posto. A questo proposito stò facendo un pò di prove e per ora è andato tutto bene: le macchine virtuali create con la versione precedente di VirtualBox continuano a girare regolarmente.

Solo una cosa è andata male. Nonostante continuassero a funzionare quelle già installate, ho provato ugualmente ad installare le VirtualBox Guest Additions della nuova versione direttamente su quelle vecchie. Risultato: la macchina virtuale si blocca durante l'avvio di Linux una volta si e l'altra pure. Ora sto procedendo con una reinstallazione ex-novo della Suse e poi riproverò ad installare le VirtualBox Guest Additions; vi farò sapere com'è andata nel seguito del tutorial

che ho linkato sopra.

Bene, non vi resta che fare il download di VirtuaBox e provarlo anche voi, ne vale davvero la pena.

@:)>

Pubblicato alle 11.44 Argomenti: news, software, Virtualbox, virtualizzazione

12 commenti:

Gianbi ha detto...

Ciao e complimenti vivissimi per il tuo blog (che è già finito tra i miei segnalibri). Qualsiasi cosa cercassi su VirtualBox l'ho trovata qui, e se posso permettermi di darti un consiglio continuerei su questa linea! ;)

Ho provato per la prima volta ieri Virtualbox (subito con la nuova versione), funziona a meraviglia, ma non ho assolutamente capito come far funzionare il discorso del copia-incolla tra host e guest e le cartelle condivise. Inoltre non riesco ancora a far funzionare la chiavetta USB.

In ogni modo davvero un ottimo programma!

9 giugno 2007 10.22

Giornale di Sistema ha detto...

Ciao Giambi!

Grazie, e per il consiglio, e per i complimenti. :)

Venendo ai tuoi quesiti:

- per poter far funzionare il copia-incolla fra il PC virtuale e quello reale, occorre che siano installate le VirtualBox Guest Additions. Porta un pò di pazienza: sto preparando alcuni post con tutte le immagini sulla loro installazione, appena pronti saranno online. ;)

- questa benedetta chiavetta USB è la croce-delizia di VirtualBox, vedo che in molti hanno avuto problemi. Personalmente mi è andato tutto liscio come l'olio nella configurazione Windows host e Linux guest, non ho sperimentato la configurazione contraria. Saresti così gentile da postarmi qualche informazione in più sulla tua configurazione software, sulla procedura utilizzata per tentare di attivare la chiavetta, e tutto quello che ritieni utile per inquadrare la situazione?

Mi rivolgo anche a tutti gli altri lettori: qualcuno ha già incontrato e risolto problemi di funzionamento della chiavetta USB? Avete voglia di postare la vostra soluzione? :-D

Grazie in anticipo. :)

9 giugno 2007 19.30

Gianbi ha detto...

Ciao e grazie per aver risposto.

In realtà il problema chiavetta USB l'ho risolto. E' bastato aggiungere una riga nel file etc/fstab di linux.

Per Ubuntu:

```
#usbfs
none /proc/bus/usb usbfs devgid=46,devmode=664 0 0
```

Ora mi manca solo il fatto di poter modificare i file contenuti nel sistema host dal sistema guest (si intende questo come cartelle condivise o no?)... non ho ancora capito come fare! Le VirtualBox Guest Additions comunque le ho già installate, è bastato un click!

10 giugno 2007 3.29

Giornale di Sistema ha detto...

Ciao Gianbi.

Sono contento finalmente funziona anche questa benedetta chiavetta USB, e per dare maggiore risalto alla tua soluzione a beneficio di tutti, ho linkato il tuo commento al post "VirtualBox: come configurare le porte USB".

Grazie davvero. :)

10 giugno 2007 9.52

Gianbi ha detto...

Felice di essere stato di aiuto.

Ho risolto anche il problema delle cartelle condivise.

Volevo però chiederti una cosa: con la nuova versione sai se c'è modo di aumentare lo spazio assegnato al sistema guest? Mi sono reso conto di averne assegnato troppo poco... Come posso fare per non perdere l'installazione che ho già?

Grazie

10 giugno 2007 18.15

Giornale di Sistema ha detto...

Se parli di memoria RAM si, è sufficiente cambiarne la quantità assegnata alla macchina virtuale, dopo averla selezionata, dal tab "Dettagli" alla voce "Generale", menù "Memoria di base".

Se parli della capienza del disco rigido virtuale, sembra proprio di no, non è ancora possibile aumentarne la capienza massima una volta allocato di una certa grandezza. Se per te è importante non reinstallare tutto, puoi però allocarne un secondo e "attaccarlo" alla macchina virtuale.

@:)>

11 giugno 2007 12.45

marcos ha detto...

@gianbi: ringrazio anche io gianbi per l'aiuto alla configurazione delle porte USB in virtualbox e visto che è stato così brillante ti chiedo per quale motivo è così lento quando apro la chiavetta? alle volte si impianta tutto!!!!

....

12 giugno 2007 12.45

Gianbi ha detto...

@ marcos

Devi fare attenzione: per prima cosa inserisci la chiavetta e aspetta che il sistema host la rilevi, dopodichè "smontala" e avvia il sistema virtuale.

Maggiori chiarimenti li trovi qui: <http://www.gianbzone.it/forum/viewtopic.php?p=10520#10520>

P.S. Se non si può mettere il link editate pure.

P.P.S. Colgo l'occasione per invitare "giornale di sistema" ad iscriversi, mi farebbe piacere.

12 giugno 2007 14.35

Giornale di Sistema ha detto...

Grazie Gianbi, volentieri. :)

13 giugno 2007 9.00

marcos ha detto...

@gianbi: ho una chiavetta USB e lui me la vede e mi setta correttamente i parametri, riprovando sono riuscito anche a vederla nel sistema virtuale ma la lentezza per aprirla è immensa.....
anzi nemmeno al apre e si impalla.....

cosa ne pensi???

ho virualizzato win_2000 su ubuntu 7.04 con 1GB di ram di cui 300 mb condivisi con la macchina virtuale. 32MB di scheda video condivisa e HD fisso e non dinamico.

.....

grazie

14 giugno 2007 15.43
Anonimo ha detto...

Ciao a tutti ho un problema con virtualbox non riesco a mettere nella stessa rete il computer host (winxp professional) con il computer guest (linux debian). Qualcuno di voi sa indicarmi come settare la scheda di rete della macchina virtuale e i parametri ifconfig da eseguire.

8 agosto 2007 17.41
Giornale di Sistema ha detto...

Ciao. :-)

A breve pubblicherò la serie dei post riguardo a VirtualBox e la rete. Non mi occuperò specificatamente di Debian (uso una Suse come riferimento), ma mi auguro ugualmente tu possa trovare spunti per risolvere il problema.

E se qualche altro lettore del blog avesse già incontrato e risolto il problema, e ha voglia di condividere la soluzione... grazie in anticipo! :-D

9 agosto 2007 9.43

domenica 10 giugno 2007

Anche i piccioni sanno di Ubuntu?

Erano le 6 di stamane quando due piccioni in vena di scherzi hanno cominciato a tubare come forsennati proprio sul cornicione davanti alla camera da letto. E' un suono che normalmente non noto per niente, ci sono abituato, ma stamane si, l'ho sentito e mi si sono drizzate le orecchie.

I due piccioni non erano mica normali: uno faceva uubuuuntu... uubuuuntu... uubuuuntu... e l'altro gli rispondeva: kuubuuuntu... kuubuuuntu... kuubuuuntu...

E lì, ho capito quanto sono grave! 

Sarà che ho passato troppo tempo a virtualizzare distribuzioni Linux con VirtualBox 1.4.0? 

Comunque sia, mi sono alzato e ho fatto colazione. E ho cominciato a chiedermi: che faccio?
Faccio finta di nulla e lascio il tutorial su come creare PC virtuali con VirtualBox così com'è? Oppure lo aggiorno sostituendo le vecchie immagini con quelle relative alla nuova versione 1.4.0? O ancora, lo rifaccio ex-novo?

Poi ho portato lo sguardo sulla descrizione sotto il nome del blog: "Appunti ed immagini dai miei viaggi nel mondo dei bit... e non solo". Sono sempre stato convinto che al di là delle parole mostrare un'immagine aiuta tantissimo nel comprendersi. Dove ho potuto in questo blog ho sempre fatto così: poco testo, l'indispensabile, corredata da immagine, a cui il testo per la verità finisce quasi a far da didascalia.

E così ho deciso: sì, occorrono nuove immagini che siano relative alla nuova versione di VirtualBox. E' troppo diversa: intanto è in italiano, poi vi sono nuove opzioni, wizard, insomma, meglio mostrarlo così com'è ora VirtualBox 1.4.0. Consuetudine vuole poi che i vecchi post non vadano modificati, quasi a testimonianza di ciò che è stato, come un diario di cui non andiamo a modificare le pagine già scritte, ma ne aggiungiamo di nuove.

Pagine nuove che magari riprendono le vecchie, ne ampliano il discorso, lo completano, oppure lo trasformano. Il tutto legato da un filo invisibile, il tempo, che le dispone lungo un percorso lineare, sequenziale.

Bene, la decisione è presa. Nuovi post corredati di nuove immagini! Tra l'altro, avevo già pronte le immagini relative al post dell'installazione delle VirtualBox Guest Additions versione 1.3.8, ho buttato anche quelle.

Beh, buona domenica a tutti voi, spero risvegliati dall'aroma di un buon caffè e non da due piccioni con il pallino dell'informatica. 

@:)>

Pubblicato alle 8.49 Argomenti: blogging, segnalazioni, Virtualbox, virtualizzazione

martedì 12 giugno 2007

Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 1)

Nel post "Virtualizzazione: installiamo Virtualbox" abbiamo visto come installare VirtualBox, un potente programma di virtualizzazione open source. Grazie ad esso diventa realtà il sogno di far girare più sistemi operativi simultaneamente su un solo PC, ognuno nella sua finestra, e far in modo che condividano dati fra loro.

Con VirtualBox potete infatti creare dei PC virtuali completi di tutte le loro periferiche: disco rigido, scheda grafica, audio, periferiche USB e così via; su ognuno di essi installare poi un sistema operativo e farlo funzionare contemporaneamente a tutti gli altri. Potete anche fare in modo che i PC, quello reale detto *host* e quelli virtuali detti *guest*, comunichino tra loro condividendo file, cartelle ed anche il contenuto del classico copia-incolla; se avete poi una scheda di rete installata sul PC host (quello reale), potrete persino metterli in rete fra di loro.

Un'altra caratteristica molto interessante di VirtualBox è la possibilità di controllare i PC guest (quelli virtuali) da remoto tramite VRDP (VirtualBox Remote Desktop Protocol). Immaginate una rete aziendale (ma anche la vostra rete casalinga): su un PC si fa girare VirtualBox con le sue macchine virtuali e protrete collegarvi ad esse da qualunque altro PC della rete.

Sono molti gli scenari di utilizzo di VirtualBox e della virtualizzazione in generale, per una breve panoramica potete leggere il post "Virtualizzazione: cos'è e quando usarla".

Vediamo adesso come creare un PC virtuale.

Visto che la macchina host (quella reale) è dotata di Windows, una scelta carina mi sembra quella di creare una macchina guest (virtuale) su cui installare una distribuzione Linux. La creazione della macchina virtuale, tranne che per pochi dettagli, è uguale per tutti i sistemi operativi previsti; pertanto, anche se in questo tutorial useremo la Suse Linux 10.2, usate questo post come guida per la creazione di una qualunque fra le macchine virtuali ufficialmente supportate.

Cominciamo.

Dal pulsante *start* ---> *Programmi* ---> *InnoTek VirtualBox* ---> *Virtual Box*, lanciate con un clic il programma di virtualizzazione. Vi ritroverete così davanti alla schermata principale di VirtualBox (**fig. 1**). Vi sono già delle macchine virtuali pronte all'uso, giusto per darvi un'idea della cosa. Ora cliccate sull'icona *Nuova* per iniziare la creazione di una nuova macchina virtuale (**fig. 1**):

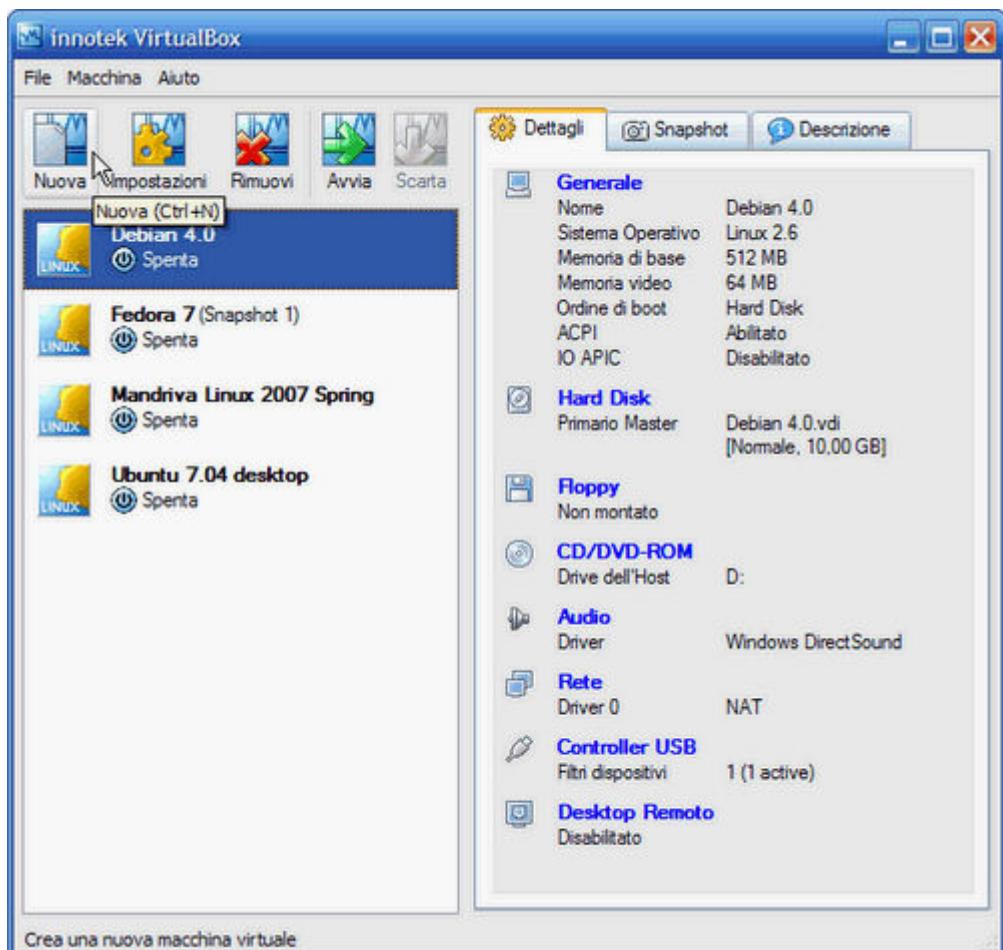


fig. 1

Si aprirà la schermata di benvenuto al wizard che vi guiderà nel creare la nuova macchina virtuale (**fig. 2**), clicchiamo su Successivo:



fig. 2

Nel campo **Nome** (fig. 3) inserite il nome della macchina virtuale, ad esempio "openSUSE 10.2"; nel campo **Sistema Operativo** selezioniamo Linux 2.6 o il nome del sistema operativo che volete virtualizzare, e andiamo avanti cliccando su **Successivo**:



fig. 3

Scegliamo ora la quantità di memoria che la nostra macchina virtuale avrà a disposizione per funzionare: 256 MB sono da considerarsi il minimo perché tutti funzioni ad una velocità accettabile, se invece avete un PC con almeno 1 GB di memoria allora 512 MB è la quantità ottimale; fatta la vostra scelta cliccate su **Successivo** (fig. 4).



fig. 4

Il nostro PC virtuale ha bisogno di un disco rigido virtuale per funzionare; cliccate su **Nuovo** (fig. 5):



fig. 5

Il wizard che si è aperto vi guiderà nella creazione del disco rigido del vostro computer virtuale, cliccate su **Successivo** (fig. 6) per continuare:

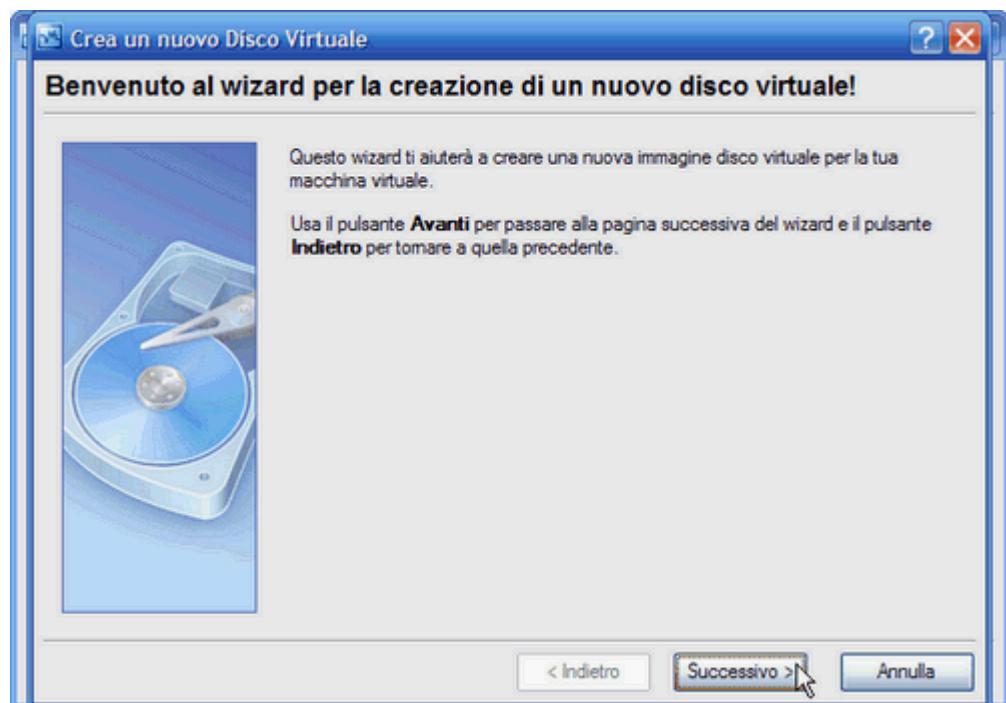


fig. 6

A questo punto dovete scegliere il tipo di disco virtuale da allocare fra i due disponibili:

- **Ad espansione dinamica;** inizialmente alloca solo una piccola quantità di spazio sul disco reale, pochi kilobyte. Man mano il sistema guest (quello virtuale) avrà bisogno di più spazio su disco, questo viene allocato in quel momento e fino a quando non avrà raggiunto il massimo stabilito, oltre la quale non va.
- **A dimensione fissa;** la quantità di spazio prescelta viene allocata subito tutta.

Quale dei due scegliere?

Valutate voi; la prima opzione permette di occupare meno spazio sul disco reale, la seconda però evita i rallentamenti che si verificano nel momento dell'allocazione "al volo" dello spazio; fatta la vostra scelta (personalmente ho scelto la prima per motivi di spazio), cliccate su *Successivo* per continuare(**fig. 7**):



fig. 7

Infine scegliete la capienza del disco rigido virtuale. Per provare comodamente Linux 10 GB saranno più che sufficienti, tenete però a mente due cose: lo spazio che destinerete al disco rigido virtuale viene sottratto al disco rigido reale del vostro computer, pertanto assicuratevi di avere spazio sufficiente su quest'ultimo. Inoltre, una volta creato il disco rigido virtuale di una certa dimensione, questa non potrà più essere modificata. L'unico modo per aumentare lo spazio a disposizione del sistema guest (PC virtuale) sarà creare un secondo disco rigido virtuale, così come su un qualunque PC, quando il disco rigido si riempie ne aggiungete un secondo. Ancora una volta, per proseguire clicchiamo su *Successivo* (**fig. 8**).



fig. 8

Siete arrivati alla schermata riepilogativa del disco rigido virtuale (tipo di disco rigido virtuale, cartella del disco rigido reale dove verrà collocato, spazio occupato), se tutto è a posto, cliccate su *Termina* (fig. 9).

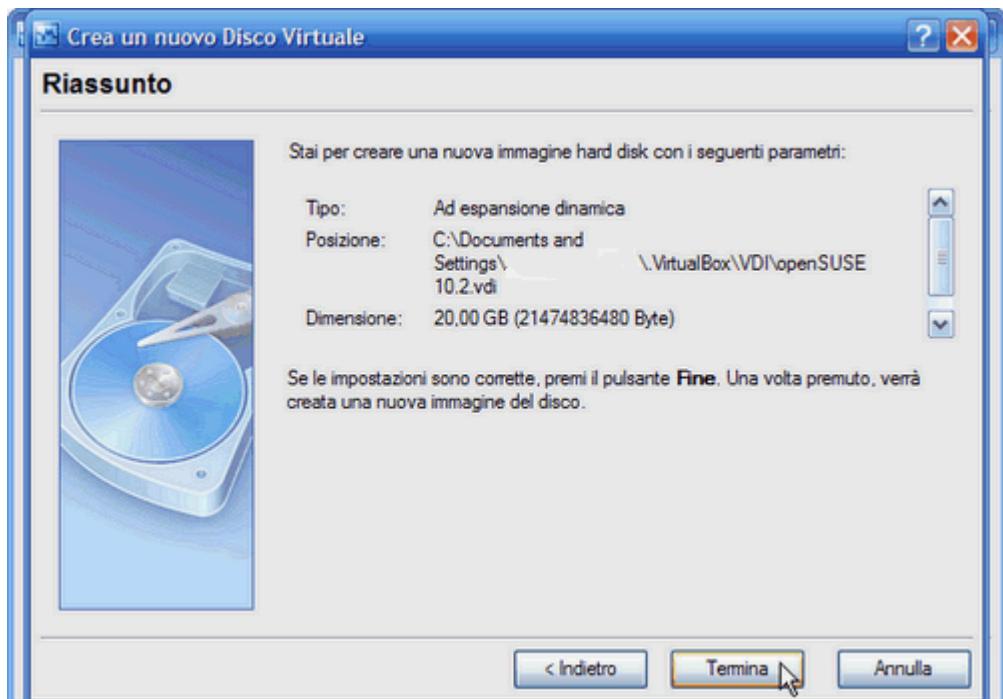


fig. 9

Bene, il disco rigido virtuale è pronto, cliccate su *Successivo* (fig. 10).



fig. 10

Anche il computer virtuale su cui installare Linux è pronto, terminiamo la procedura di setup cliccando su *Termina* (fig. 11).



fig. 11

Ed eccolo finalmente (fig. 12), la macchina virtuale è pronta.

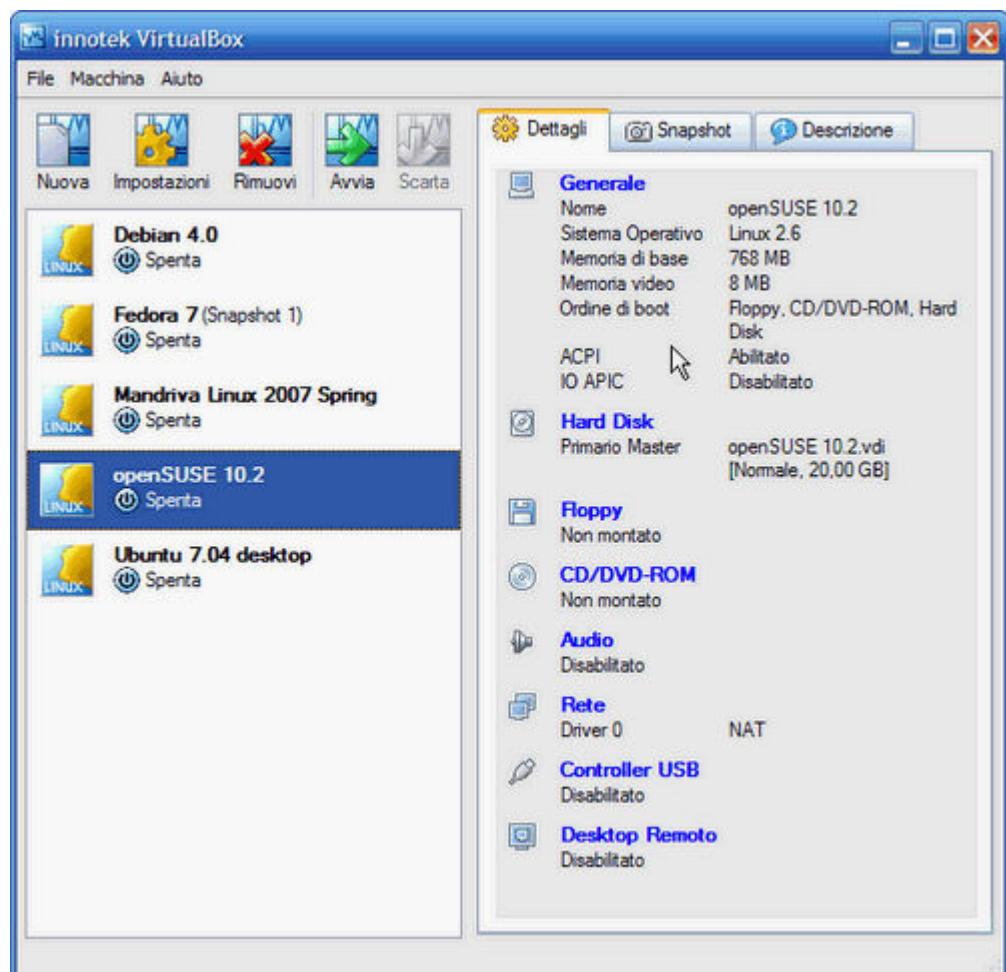


fig. 12

Come potete notare in questo momento la macchina virtuale è spenta (*Spenta*): in realtà non è ancora configurata del tutto. Se guardate infatti la scheda *Dettagli*, a destra nella schermata, noterete che Audio e controller USB sono disabilitati, il CD/DVD-ROM ed il Floppy idem, la scheda grafica ha appena 8 MB di memoria, l'ordine di boot (l'ordine in cui sulle varie periferiche viene cercato il sistema operativo da avviare) è migliorabile... insomma, bisogna ancora lavorare un pò per affinare la configurazione.

Ed è quello che faremo nel post: "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 2)".

@:)>

Pubblicato alle 14.55 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

2 commenti:

jakaiser ha detto...

Bell'articolo :) Ora sbrigati a preparare il 2... la rete devo ancora capire come configurarla :)
In questo blog, mi sento... a casa :)

12 giugno 2007 18.05
Giornale di Sistema ha detto...

AHAHAHAHAH ... grande jakaiser! Davvero, quando ho letto il post mi sono sganasciato dalle risate. :-D
Grazie per la simpatia, ma soprattutto per una cosa davvero bella che hai detto: di sentirti... a casa. Scalda il cuore. :)

12 giugno 2007 21.59

giovedì 14 giugno 2007

Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 2)

Nel post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 1)" avevamo iniziato la creazione di una macchina virtuale, in questo post proseguiamo il lavoro di configurazione.

Vi ricordo che la procedura di creazione e configurazione di una virtual machine (macchina virtuale) che vi sto descrivendo in questa serie di post, è uguale per tutti i sistemi operativi guest ufficialmente supportati da VirtualBox; la distribuzione Suse Linux che stiamo usando è stata presa solo come esempio.

Riavviate VirtualBox, vi ritroverete davanti al pannello di controllo (**fig. 1**). Selezionate la macchina virtuale *openSuse 10.2* e cliccate sul pulsante *Impostazioni* (da tastiera *Ctrl+s*) nella barra degli strumenti:

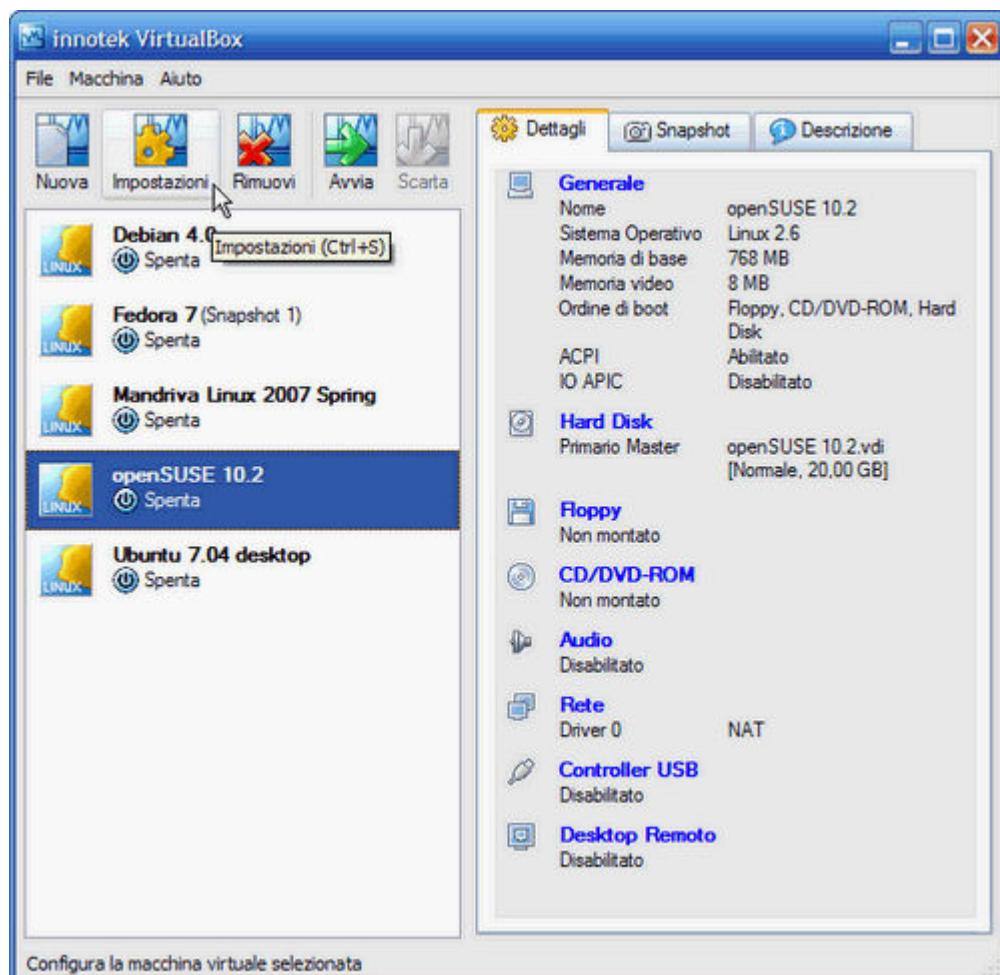


fig. 1

La voce *Generale* nella barra di navigazione, a sinistra in alto nell'immagine (**fig. 2**), è costituita da 3 schede visualizzate a destra:

- *Base*; contiene una serie di voci quali il nome della macchina virtuale ed il sistema operativo da installarvi all'interno; queste voci risultano già presenti grazie alla configurazione tramite wizard fatta la volta scorsa. A noi interessa la voce *Dimensione della memoria video* (sempre **fig. 2**); cliccate sul cursore e fate lo scorrere fino a raggiungere la quantità desiderata di Megabyte da destinare alla scheda video. Tenete presente che in linea di massima, più sarà alta questa quantità e più sarà alta la risoluzione video ed il numero di colori a disposizione nel PC virtuale. Vi è però un limite: finché non installate le VirtualBox Guest Additions, non andrete comunque oltre la risoluzione massima di 1024 x 768 con 65.000 colori, una volta installate invece potrete arrivare fino a 3200 x 2400 con 16 milioni di colori. Quando avete finito, cliccate sulla scheda *Avanzato* (**fig. 2**):

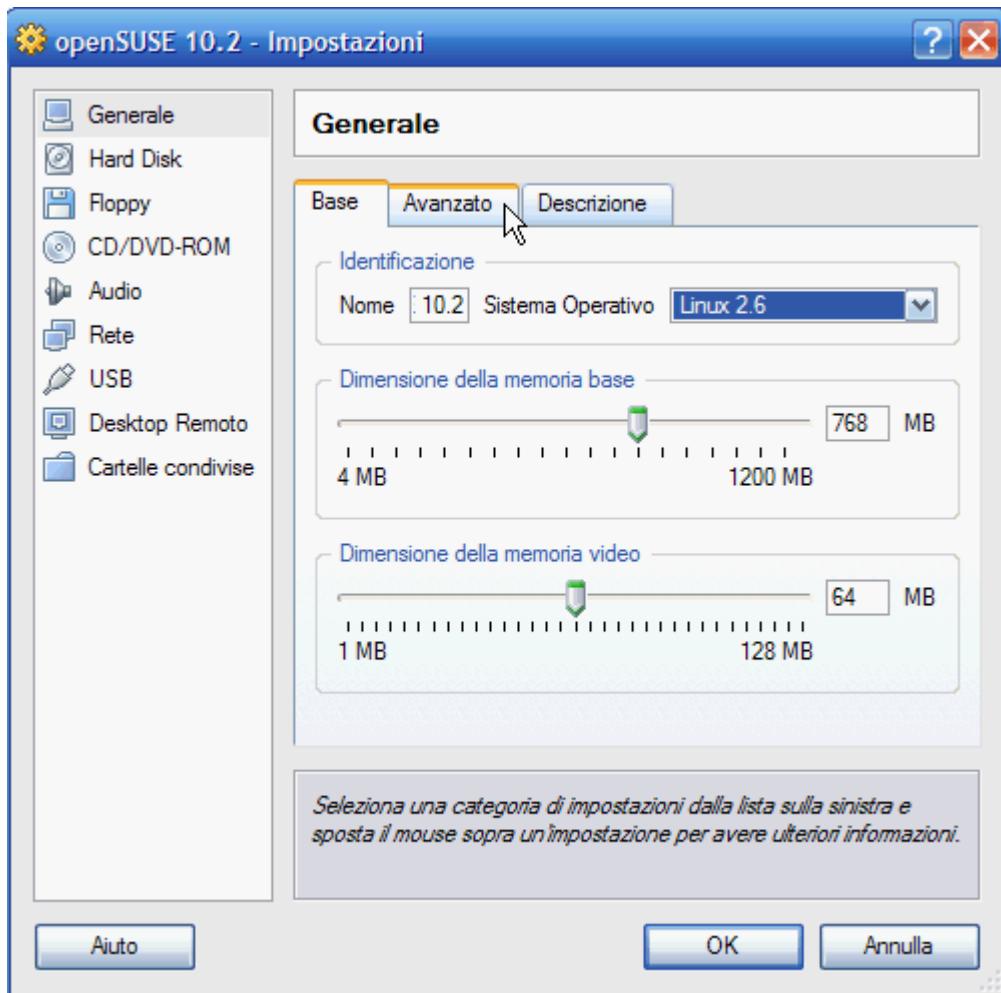


fig. 2

- **Avanzato (fig. 3).** La voce *Ordine di Boot* è la lista dei dispositivi da cui la macchina virtuale cercherà di far partire il Sistema Operativo nell'ordine specificato: se, come è normale, il vostro ordine è diverso, configuratevelo allo stesso modo di fig. 3: la macchina virtuale cercherà prima sul disco rigido virtuale, se non trova nulla cerca sul CD/DVD-ROM. Da Floppy e da Rete potete anche deselezionarne le relative opzioni, in genere non vi è necessità di fare il boot da questi dispositivi. Le impostazioni di default della voce *Funzionalità avanzate* in genere vanno bene, lasciatele così come sono. La funzionalità *Appunti condivisi* attivatela come in fig. 3, vi permetterà di usare il copia-incolla tra sistema host e guest, cioè tra PC reale e virtuale. Ricordatevi però: funzionerà davvero solo quando saranno state installate anche le VirtualBox Guest Additions nel sistema guest, cioè nel PC virtuale.

- **Descrizione (fig. 3).** Vi permette di aggiungere un commento alla vostra macchina virtuale, ad esempio le indicazioni di una configurazione particolare che volete tenere a mente.

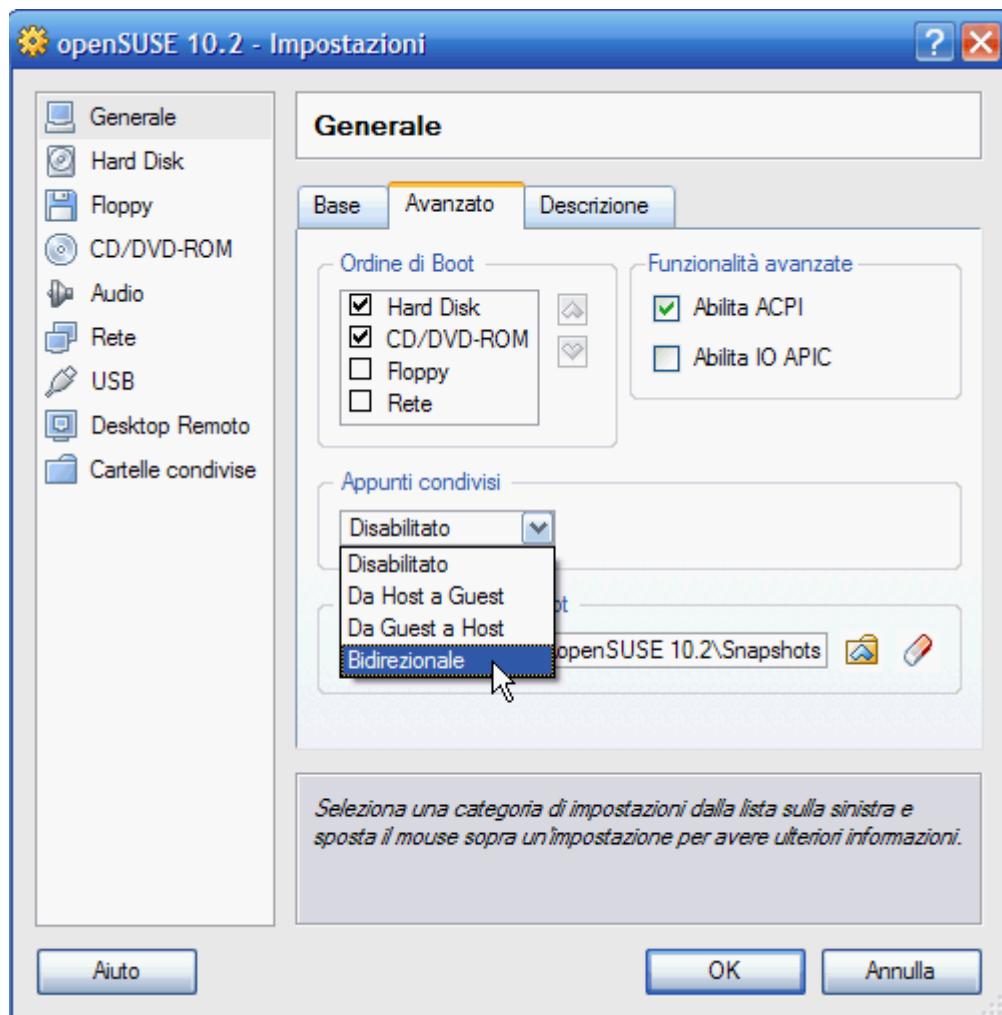


fig. 3

Terminata la configurazione della voce *Generale*, cliccate su *Hard Disk*, la voce successiva nella barra di navigazione sulla sinistra(**fig. 4**). Quello che vedete è il disco rigido principale del vostro PC virtuale, la sua capienza (20 GB), ed il nome del singolo file sulla macchina reale in cui è memorizzato.

Come potete notare, vi è la possibilità di connettere alla macchina virtuale ancora altri 2 dischi rigidi virtuali. Questa possibilità torna utile quando lo spazio sul disco rigido principale termina: VirtualBox non supporta, almeno fino a questa versione, il ridimensionamento della grandezza di un disco rigido virtuale una volta creato. Tramite il Gestore dei dischi virtuali (a breve un post specifico) potete però allocare un secondo disco rigido virtuale più capiente e connetterlo alla macchina virtuale, aumentando così lo spazio totale a disposizione.

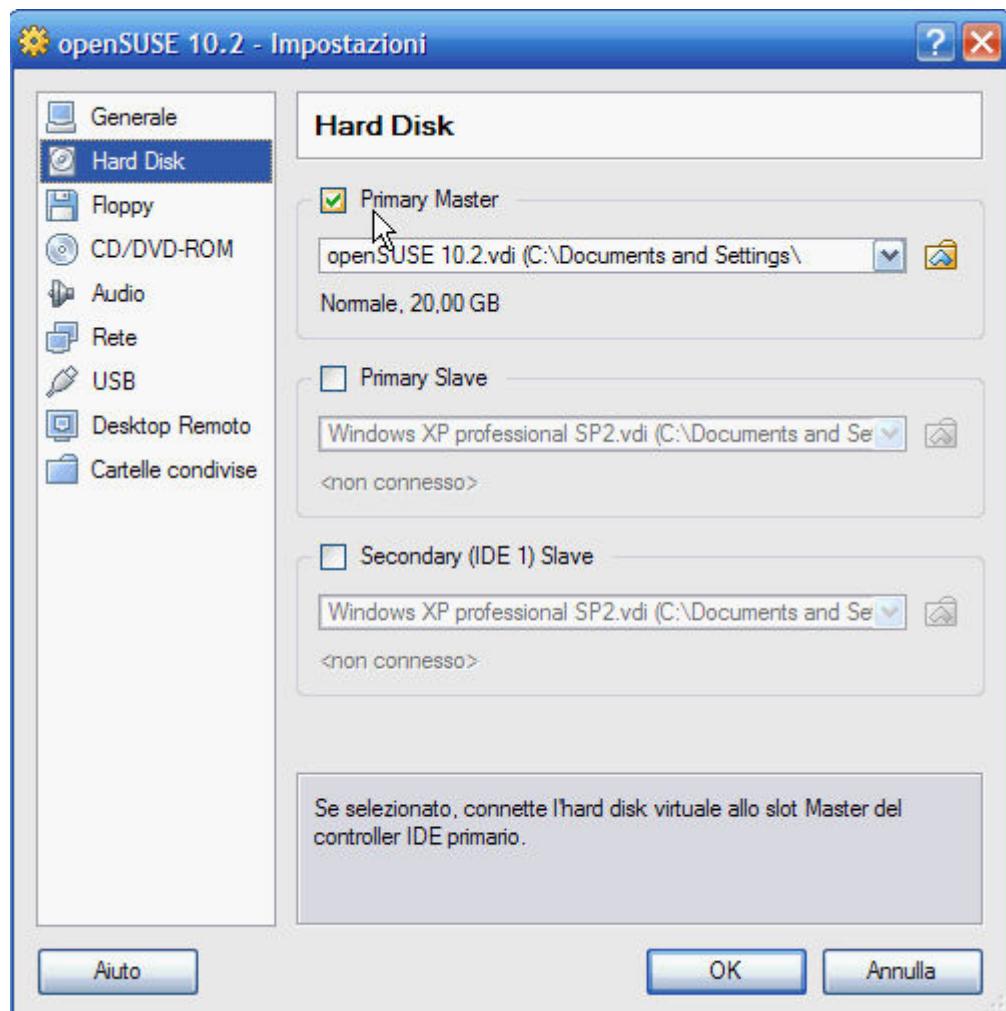


fig. 4

Se avete configurato bene lo spazio iniziale del disco rigido virtuale (almeno 10 Gb) non dovreste avere la necessità di cambiare nulla. Nel caso invece vi foste accorti che lo spazio riservato al disco rigido virtuale è insufficiente potete usare il Gestore dei dischi virtuali per cancellare e riallocare il disco rigido virtuale di dimensioni adeguate. Se è tutto a posto così com'è, andiamo avanti cliccando sulla voce *Floppy* (fig. 5) nella barra di navigazione a sinistra.

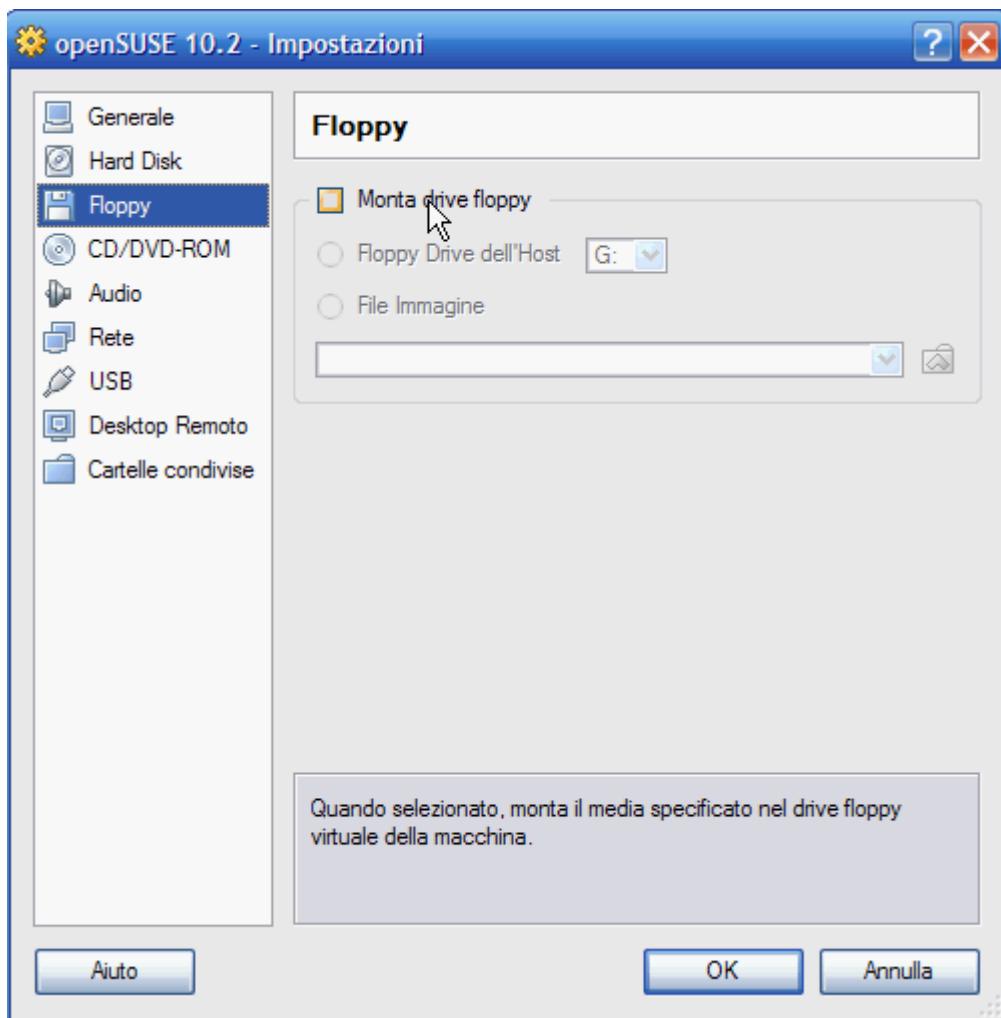


fig. 5

Del floppy generalmente se ne può fare a meno (sulla mia macchina, un notebook, non è neanche presente), pertanto lasciate deselezionata la casella *Monta drive floppy* e cliccate sulla voce successiva, **CD/DVD-ROM** (fig. 6). Se invece il floppy proprio vi dovesse servire, tenete presente che la configurazione è analoga alla quella del CD/DVD-ROM, usate quella come riferimento.

Spuntate la casella *Mounta drive CD/DVD* (sempre fig. 6), avrete così disponibile anche il CD/DVD nel vostro PC virtuale. Se il Linux (o comunque il sistema operativo guest) da installare è su un CD/DVD allora scegliete la prima opzione, *Drive CD/DVD dell'Host*, e la sua lettera di unità sul PC reale (in genere è una D, se non siete sicuri aprite Gestione Risorse di Windows e controllate la lettera con cui appare visualizzato, in genere da D a Z, e con una scritta del tipo: unità CD/DVD o cose simili).

Se invece, come spesso accade, avete a disposizione il file iso, allora VirtualBox ci mette a disposizione una comodissima opzione per installare Linux direttamente da quel file senza doverlo più masterizzare. E' sufficiente cliccare sulla seconda opzione, *file immagine ISO*, e poi nel punto dove vedete il mouse (ancora fig. 6).

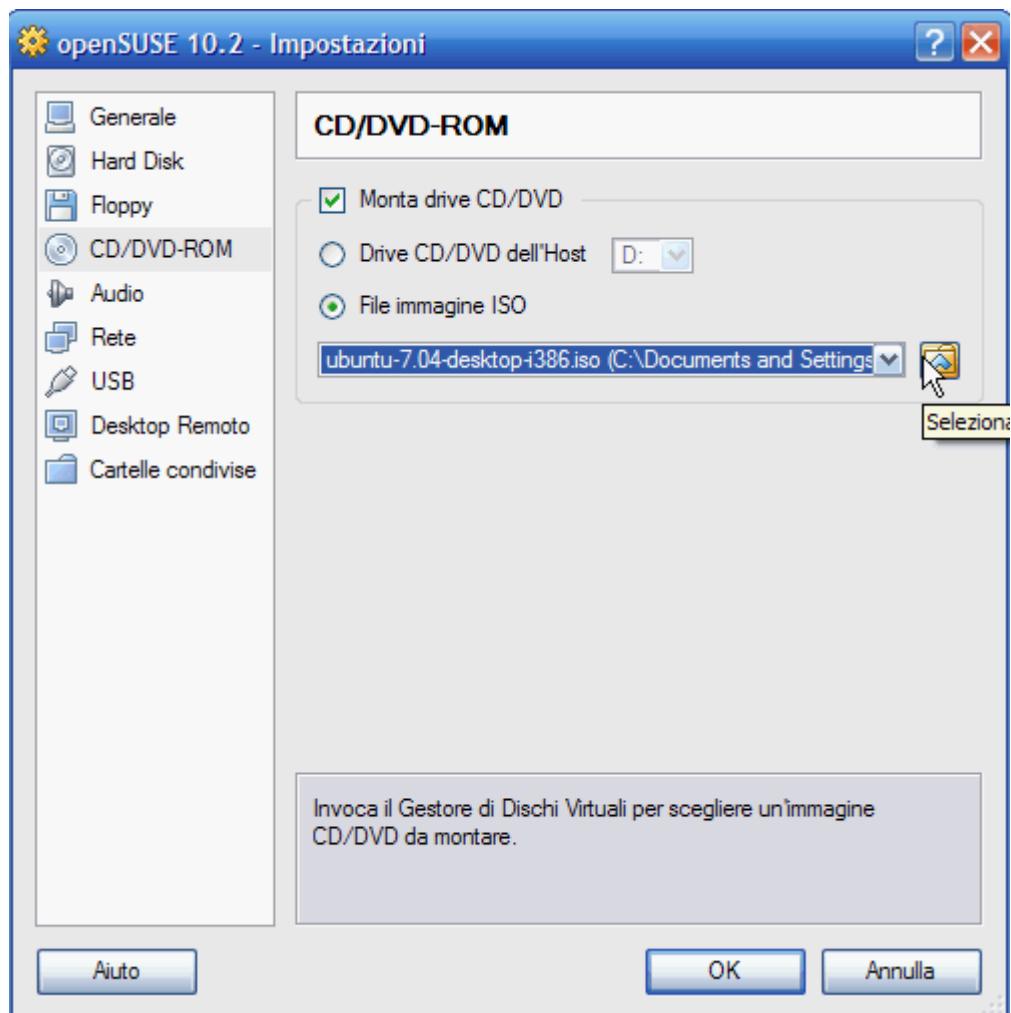


fig. 6

Si aprirà il Gestore dei dischi virtuali (vi ricordo, a breve un post specifico), in **fig. 7**. Come potete osservare, sono già presenti alcuni file ISO, ma non quello relativo alla distribuzione Suse che stiamo usando qui, come esempio, per questa guida alla creazione di macchine virtuali con VirtualBox. Cliccate allora sul pulsante *Aggiungi*:

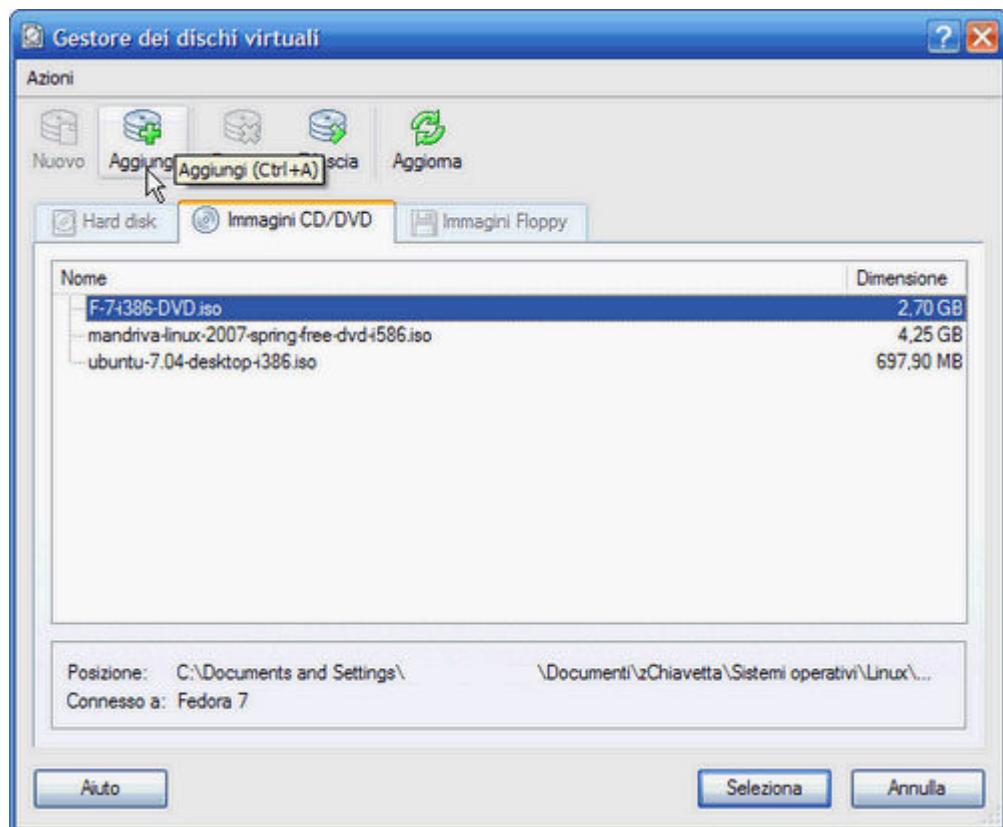


fig. 7

Si aprirà la finestra di ricerca file, navigate fino alla cartella dove si trova il vostro file iso, selezionatelo e cliccate su *Apri* (fig.8).

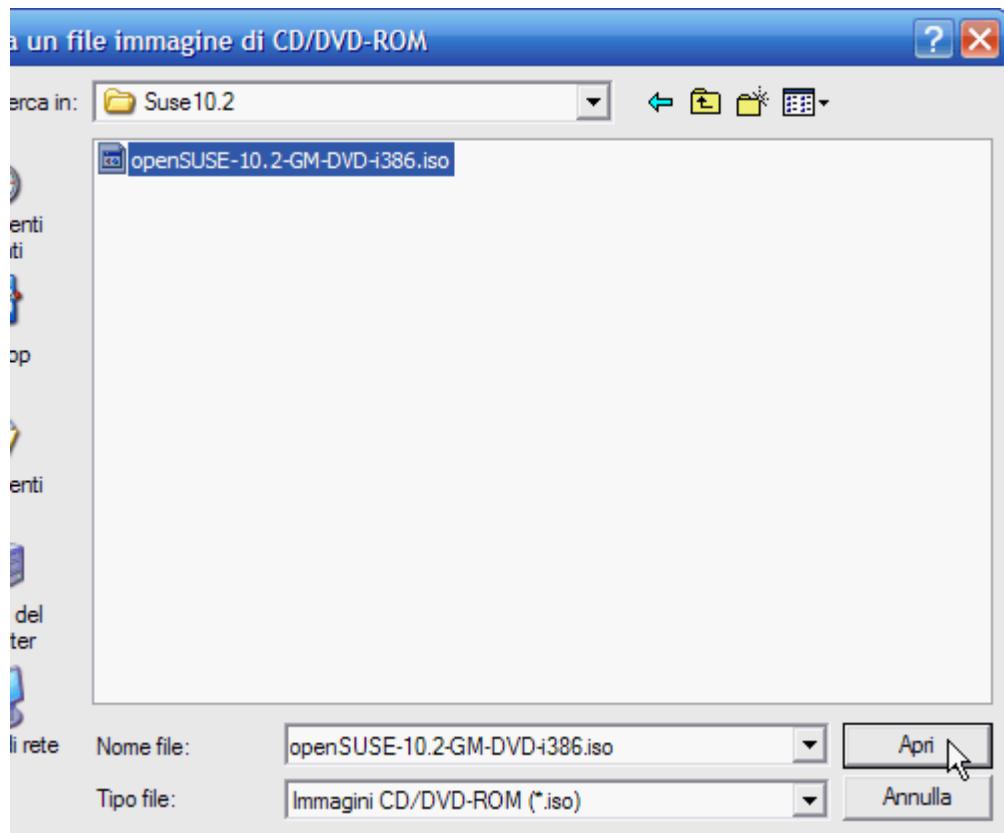


fig. 8

Sarete riportati indietro al Gestore dei dischi virtuali con il file iso inserito già selezionato: cliccate su **Seleziona** per andare avanti (**fig. 9**):

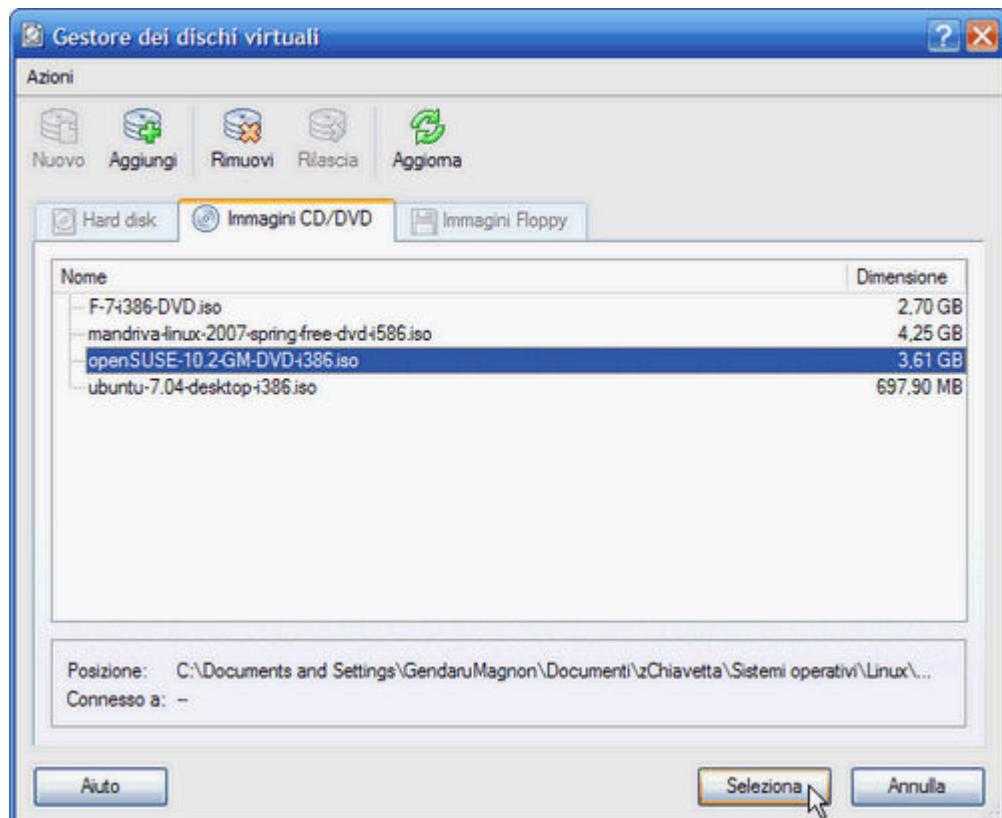


fig. 9

Ed eccola finalmente, l'immagine iso da cui installare la Suse Linux è caricata nel CD/DVD virtuale (**fig. 10**). Questa è indubbiamente una grande comodità di VirtualBox: non ci sarà più bisogno di perdere tempo a masterizzare una distribuzione e sprecare cd/dvd nel caso non fossero riscrivibili. Basterà leggerne la iso ed il gioco è fatto. Inoltre, leggendo un file direttamente dal disco rigido del PC host (quello reale) invece che dal suo cd/dvd, notoriamente la periferica più lenta tra le due, anche l'installazione procederà più spedita.

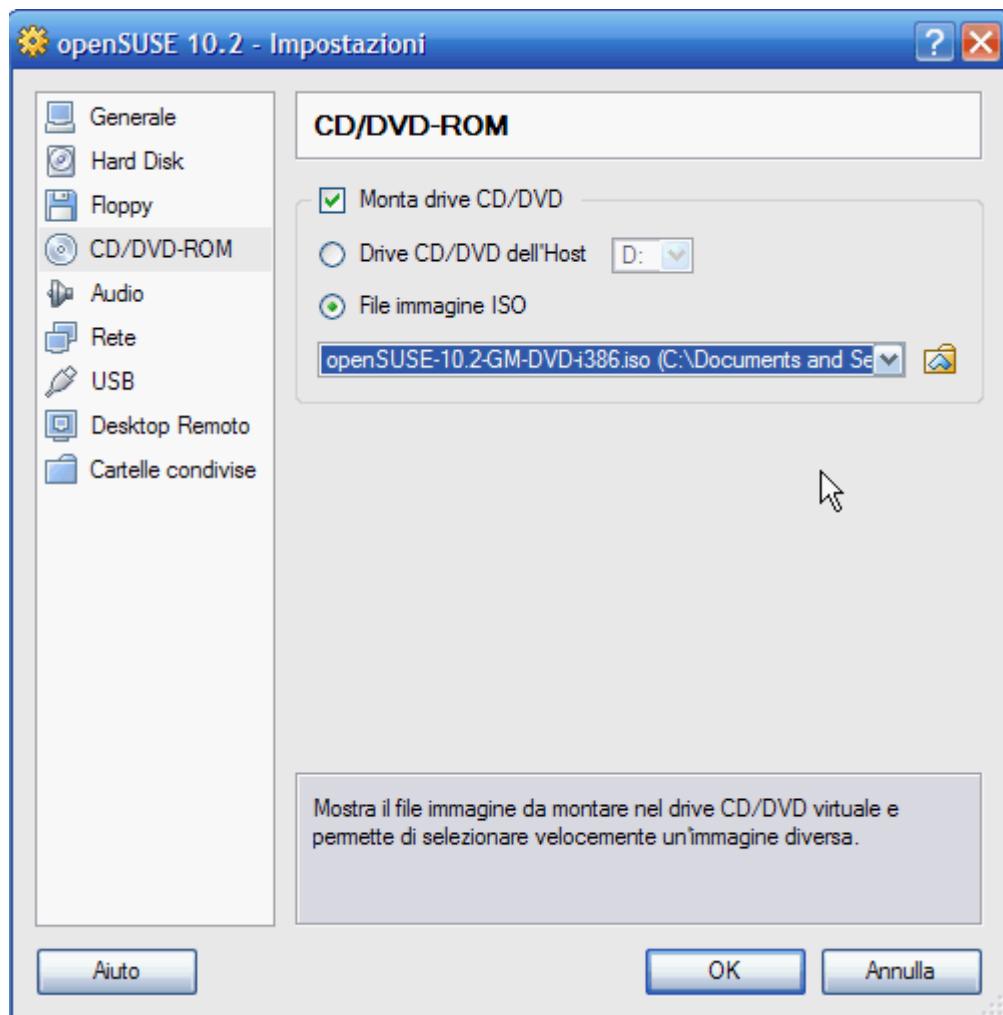


fig. 10

Ok, la carne messa a cuocere è tanta, e non è ancora finita. Mancano argomenti topici come la configurazione della rete e la configurazione delle porte USB, meglio fermarci qui e dedicar loro tutto il prossimo post, "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 3)".

@:>

Pubblicato alle 12.40 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

mercoledì 20 giugno 2007

VirtualBox: problemi con i dischi virtuali a dimensione fissa?

Il tutto è nato da una chiacchierata con Gianbi a proposito delle prestazioni del disco rigido di una macchina virtuale a seconda del tipo: "Ad espansione dinamica" oppure "A dimensione fissa" (fig. 1).

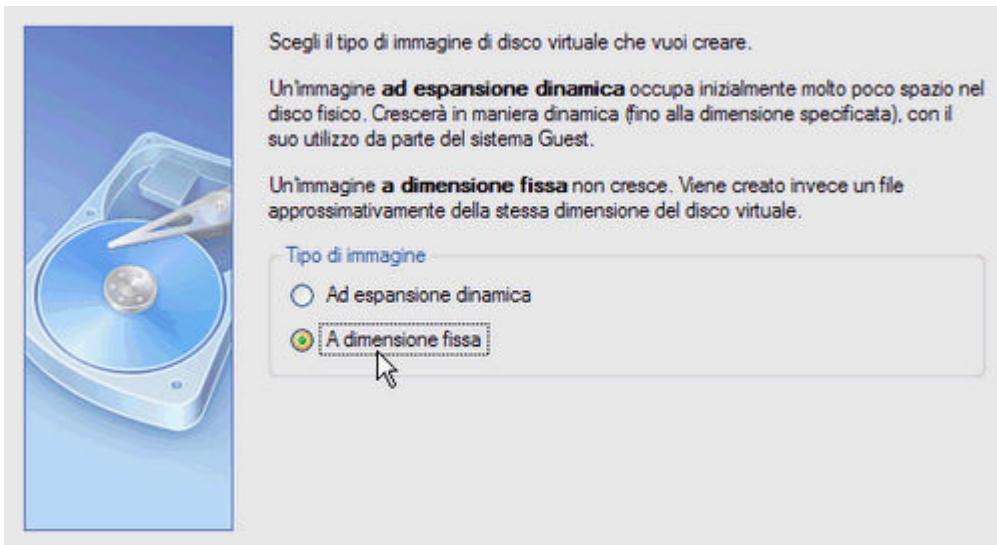


fig. 1

Ebbene: ripetute prove di installazione di una Suse 10.2, e per verifica una Mandriva 2007 Spring, usando un disco virtuale di tipo "A dimensione fissa", hanno portato sempre allo stesso errore -3009 durante la formattazione delle partizioni d'installazione.

Per ora non ho trovato nulla a riguardo nel forum ufficiale di VirtualBox; qualcuno ha avuto la stessa difficoltà? E' riuscito a risolvere il problema? Se sì, gli va di condividere con tutti la soluzione?

Grazie. :-)
@:)>

Pubblicato alle 9.44 Argomenti: segnalazioni, software, Virtualbox, virtualizzazione

2 commenti:

Gianbi ha detto...

Ciao!
Come ti ho detto, io non ho avuto problemi di questo tipo, quindi non saprei come aiutarti. Tu hai scoperto qualcosa di nuovo?

3 luglio 2007 11.43
Giornale di Sistema ha detto...

Ciao Gianbi. :-)

Purtroppo è mancato il tempo per approfondire la questione, se hai avuto modo di leggere il post 'Il blog, il lavoro, la vita...' saprai sicuramente anche il perché. Ci ritorno su appena ho la testa più sgombra da altri pensieri e ti faccio sapere.

Ciao ciao!
@:)>

4 luglio 2007 17.57

mercoledì 20 giugno 2007

Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 3)

Nel post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 2)", avevamo proseguito il setup di una macchina virtuale configurando:

- la quantità di memoria base e video del PC virtuale;
- l'ordine di boot;
- la condivisione degli appunti per rendere possibile il copia-incolla fra guest e host;
- la connessione (mount) di uno o più dischi rigidi virtuali al PC virtuale;
- il mount del Floppy;
- il mount del CD/DVD con la possibilità di installazione diretta da file ISO caricato tramite il Gestore dei dischi virtuali.

In questo post proseguiremo con la configurazione della scheda audio, della scheda di rete e delle porte USB.

Vi ricordo che la procedura di creazione e configurazione di un PC virtuale (virtual machine o anche macchina virtuale) che vi sto descrivendo in questa serie di post, è uguale per tutti i sistemi operativi guest ufficialmente supportati da VirtualBox; la distribuzione Suse Linux che stiamo usando è stata presa solo come esempio.

Cliccate sulla voce *Audio* nella barra di navigazione a sinistra (**fig. 1**) e configurate la sezione audio del vostro PC virtuale come indicato in figura, in questo modo il PC virtuale potrà usare la scheda audio presente sul sistema host (PC reale).

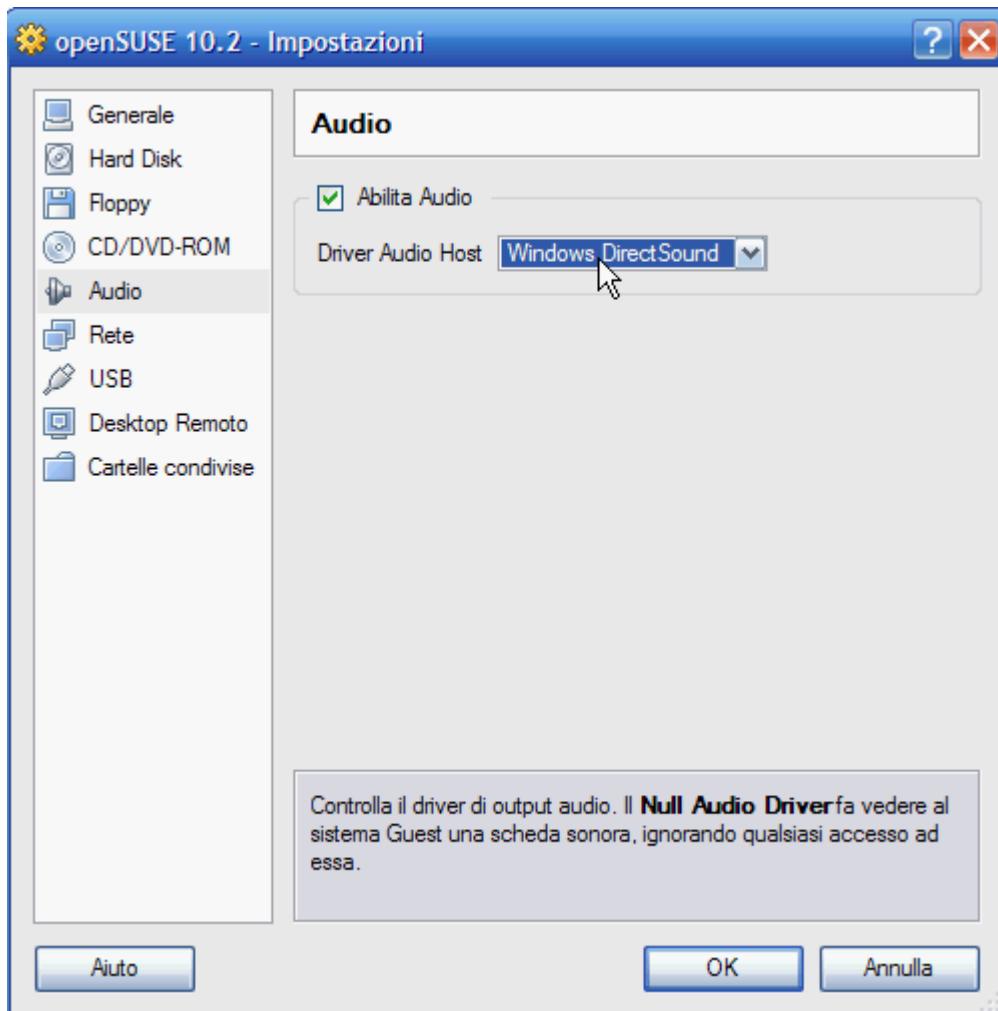


fig. 1

Sulla scheda audio non c'è molto da aggiungere, perciò cliccate sulla voce *Rete* nella barra di navigazione a sinistra e passiamo alla configurazione base della rete (**fig 2**). La prima cosa che potete notare è il numero di schede di rete

virtuali che VirtualBox è in grado di simulare: quattro per ogni PC virtuale, di tipo *AMD PCNet Ethernet card*, ognuna con la sua configurazione personalizzata.

L'impostazione di default che vedete in **fig. 2**, è in genere tutto ciò che vi serve per poter navigare in Internet direttamente dall'interno del PC virtuale, pertanto, se non avete voglia di capire come funzionino le cose, ne di vedere le altre opzioni disponibili, lasciate le impostazioni così come sono e proseguite. Se invece volete approfondire le conoscenze sulla configurazione della rete in VirtualBox, allora fate riferimento al post di approfondimento "VirtualBox: configurare la rete" (... in preparazione).

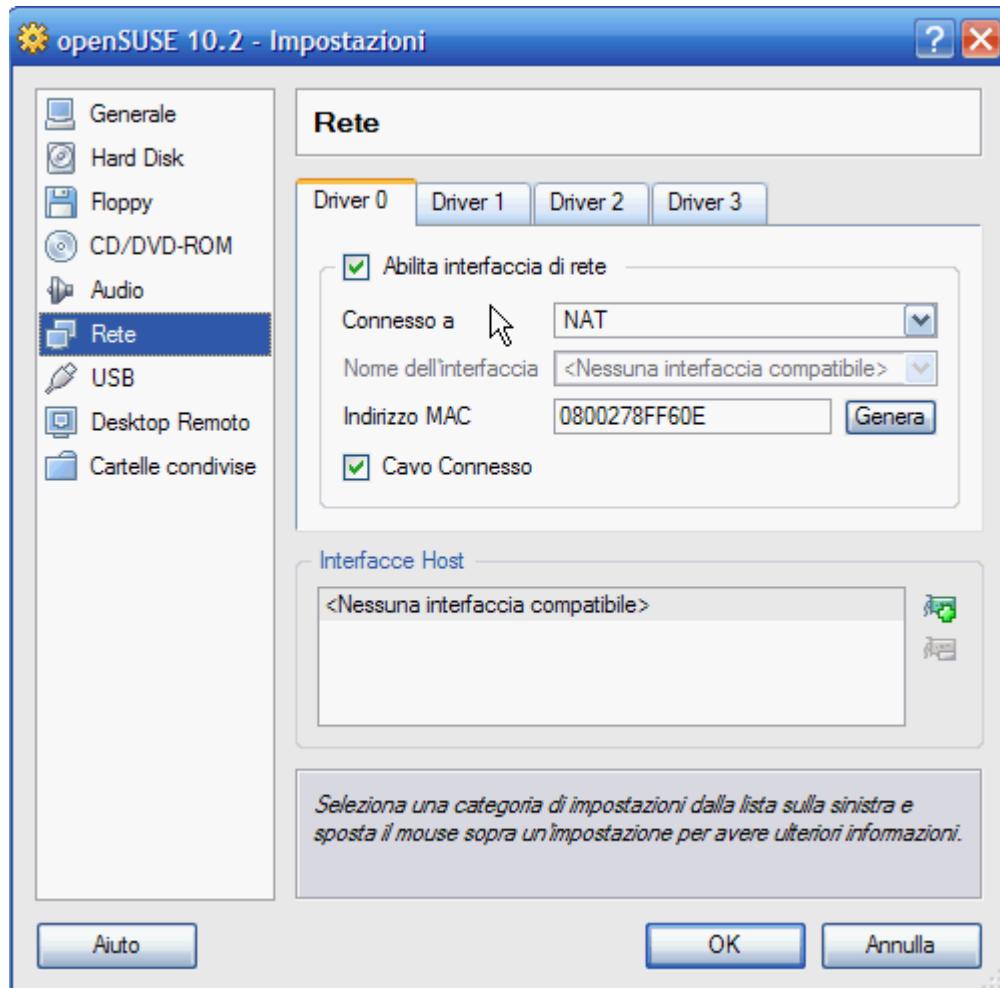


fig. 2

Una volta che avrete installato il sistema operativo sulla macchina virtuale, se avete **problemi con la rete**, ad esempio non riuscite a navigare in Internet, verificate che:

- il sistema operativo guest sia configurato per accettare l'assegnazione del nome in rete (il nome con cui viene visualizzato in una rete di PC) tramite il protocollo DHCP (**fig. 3**, esempio per SUSE, della scelta in fase di installazione).

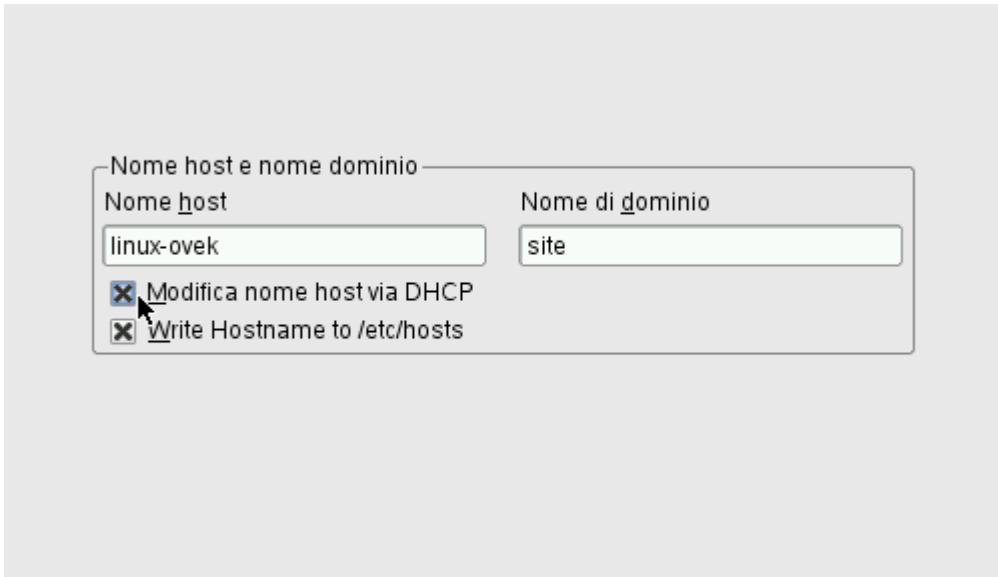


fig. 3

- che la scheda di rete sia stata correttamente riconosciuta (**fig. 4**, sempre una SUSE: notate il nome della scheda correttamente riconosciuta e l'uso del DHCP):

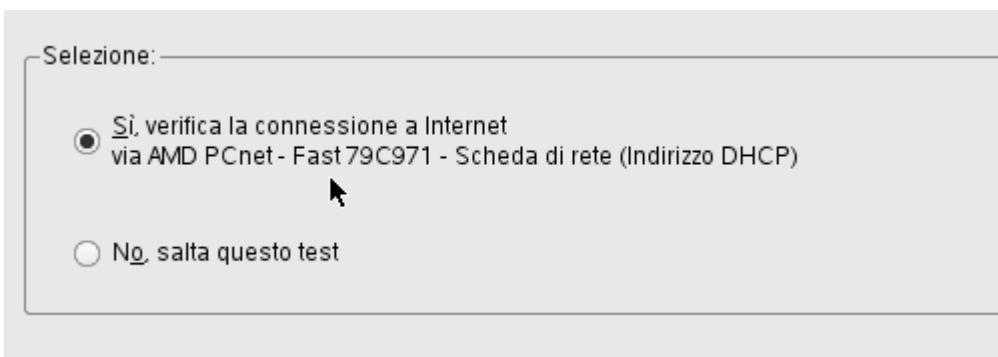


fig. 4

Terminata la configurazione base della rete, passiamo alla configurazione dell'USB. Cliccate sulla voce USB nella barra di navigazione a sinistra, sulla destra vedrete tutte le opzioni di configurazione dei dispositivi USB collegati al vostro PC reale (**fig. 5**). E sottolineo: dispositivi USB collegati.

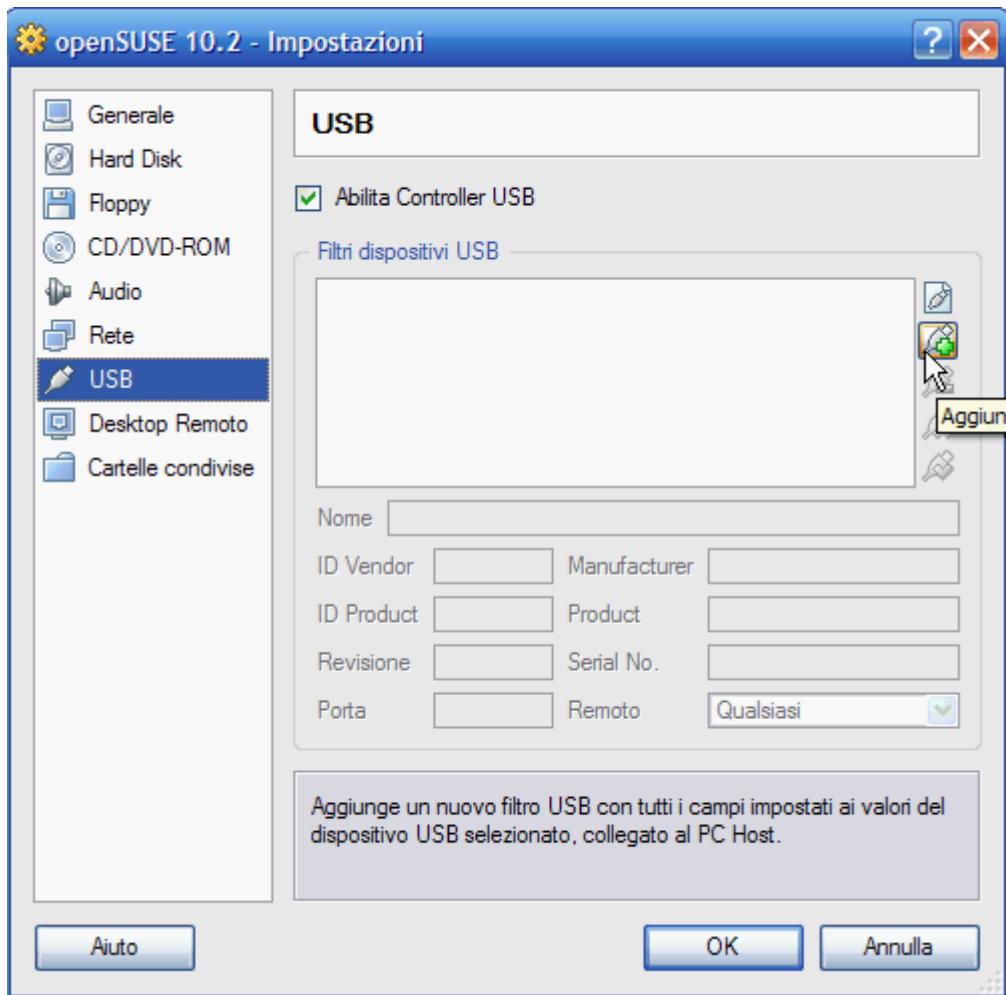


fig.5

Infatti, una volta selezionata l'opzione *Abilita Controller USB*, cliccando sul pulsante *Aggiungi* che vedete in fig. 5, si aprirà una schermata ove appariranno tutte e solo le periferiche USB collegate in quel momento alle porte del vostro PC reale (host). Pertanto assicuratevi prima che la vostra chiavetta USB ad esempio, sia già collegata e venga correttamente riconosciuta dall'host, solo dopo avviate VirtualBox; in quel momento diverrà visibile e selezionabile dal pulsante *Aggiungi*. Il risultato della selezione è in **fig. 6**:

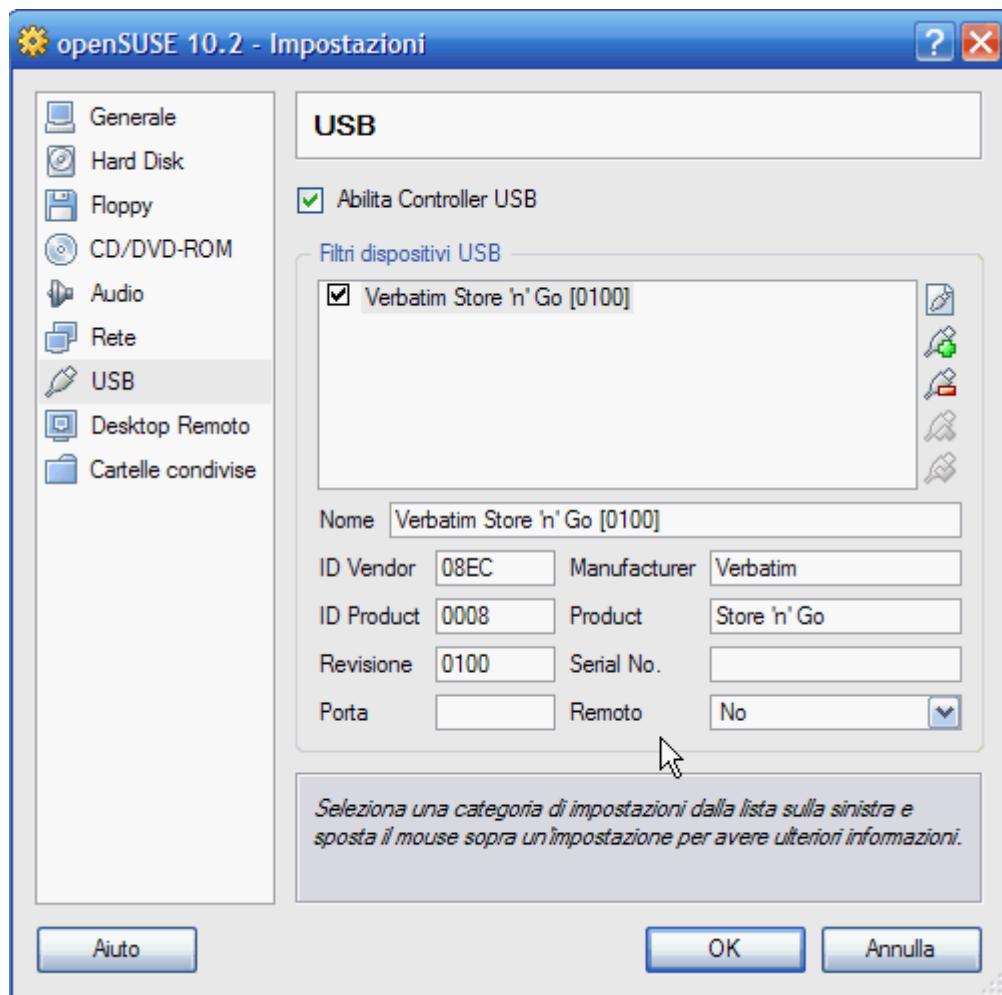


fig. 6

Se, dopo aver seguito tutte le indicazioni sulla configurazione dell'USB, permangono problemi di funzionamento delle porte USB, provate a leggere il post di approfondimento "VirtualBox: come configurare le porte USB".

Della configurazione del *Desktop Remoto* (fig. 7) non ce ne occuperemo in questo post, è possibile che lo faccia in un successivo post di approfondimento; per ora vi dirò soltanto che tramite l'attivazione del Desktop Remoto, è possibile controllare le macchine virtuali non solo dal PC reale (host) su cui girano, ma anche da altri computer reali connessi alla stessa rete di cui l'host fa parte.

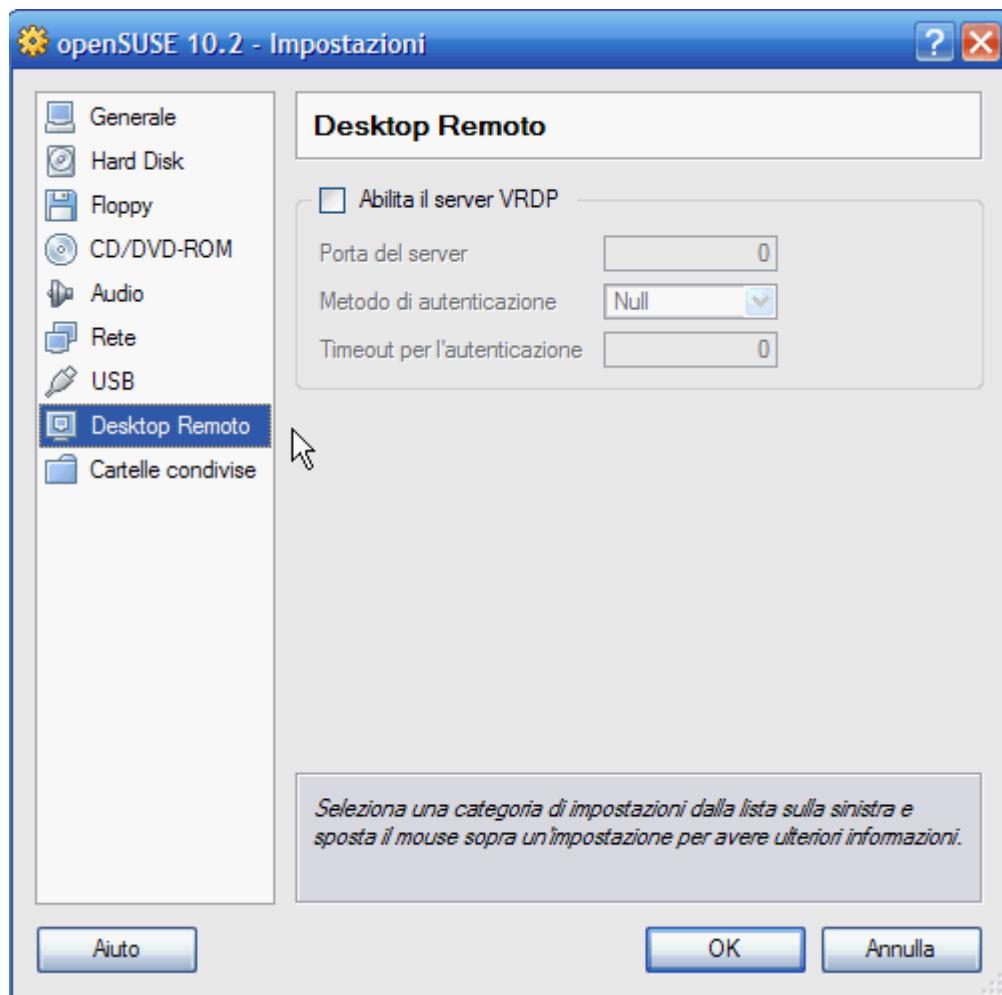


fig. 7

Bene, anche per questa volta è tutto, non vi resta che seguire il post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 4)", l'ultimo della serie, per terminare la configurazione del PC virtuale.

@:>

Pubblicato alle 14.46 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

venerdì 22 giugno 2007

Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 4)

Nel post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 3)", avevamo proseguito il setup di una macchina virtuale configurando:

- la scheda audio,
- la scheda di rete nella modalità di default,
- il protocollo DHCP (Dynamic Host Configuration Protocol),
- i dispositivi USB;

del Desktop Remoto invece ne avevamo solo accennato. In questo post termineremo la configurazione del PC virtuale con l'ultimo argomento rimasto: le *Cartelle condivise* (**fig. 1**).

Vi ricordo che la procedura di creazione e configurazione di un PC virtuale (virtual machine o anche macchina virtuale) che vi stò descrivendo in questa serie di post, è uguale per tutti i sistemi operativi guest ufficialmente supportati da VirtualBox; la distribuzione Suse Linux che stiamo usando è stata presa solo come esempio.

Ma veniamo alle cartelle condivise: tramite queste è possibile accedere ai file dell'host direttamente dall'interno del guest, senza doverlo fare tramite la rete. Fate attenzione però, questa possibilità è attiva solo per i guest a scelta tra:

- Windows 2000,
- Windows XP,
- Linux con kernel 2.4,
- Linux con kernel 2.6.

Per aggiungere la cartella che volete condividere fra host e guest, cliccate sul pulsante su cui si trova il mouse in **fig. 1**:

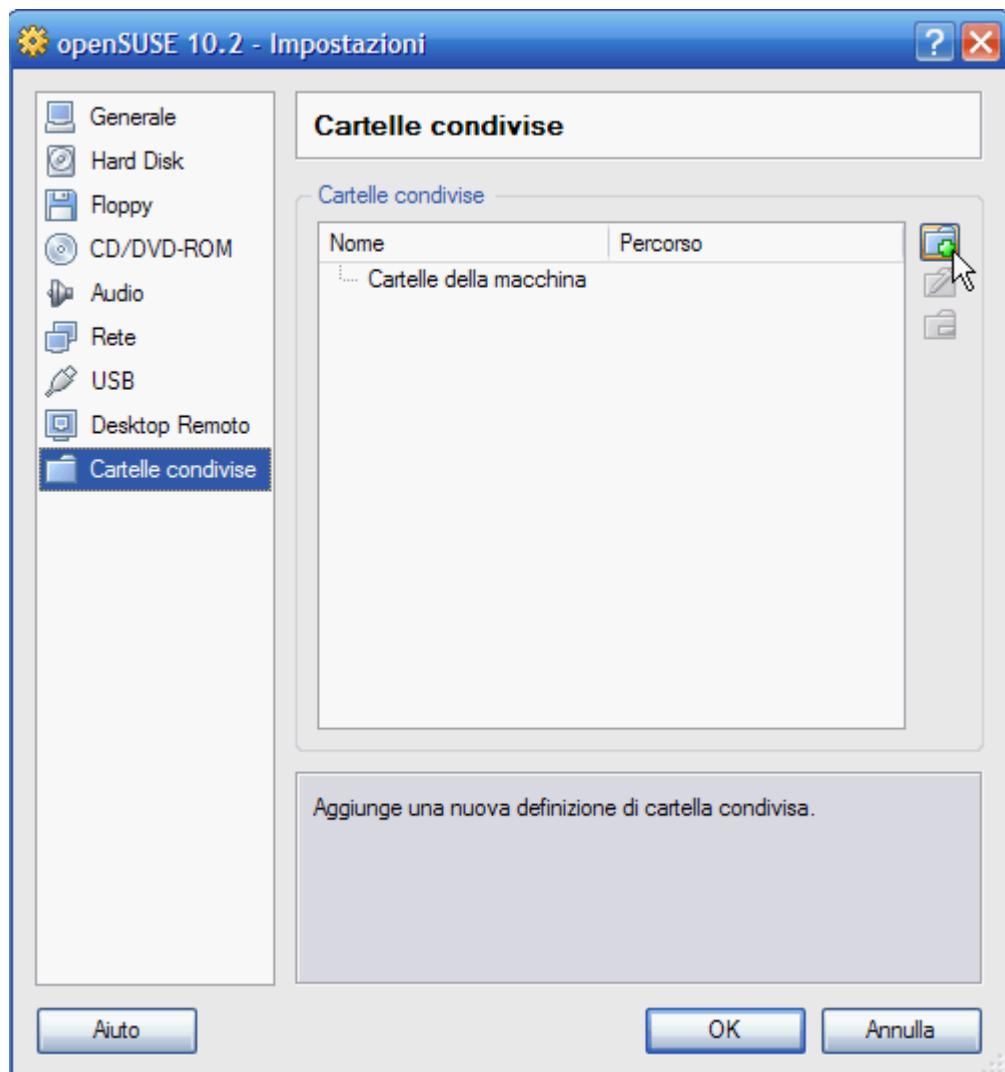


fig.1

Si aprirà un box in cui indicare il percorso ed il nome della cartella: cliccate sul pulsante in **fig. 2** per individuarne una già esistente o crearne una nuova.

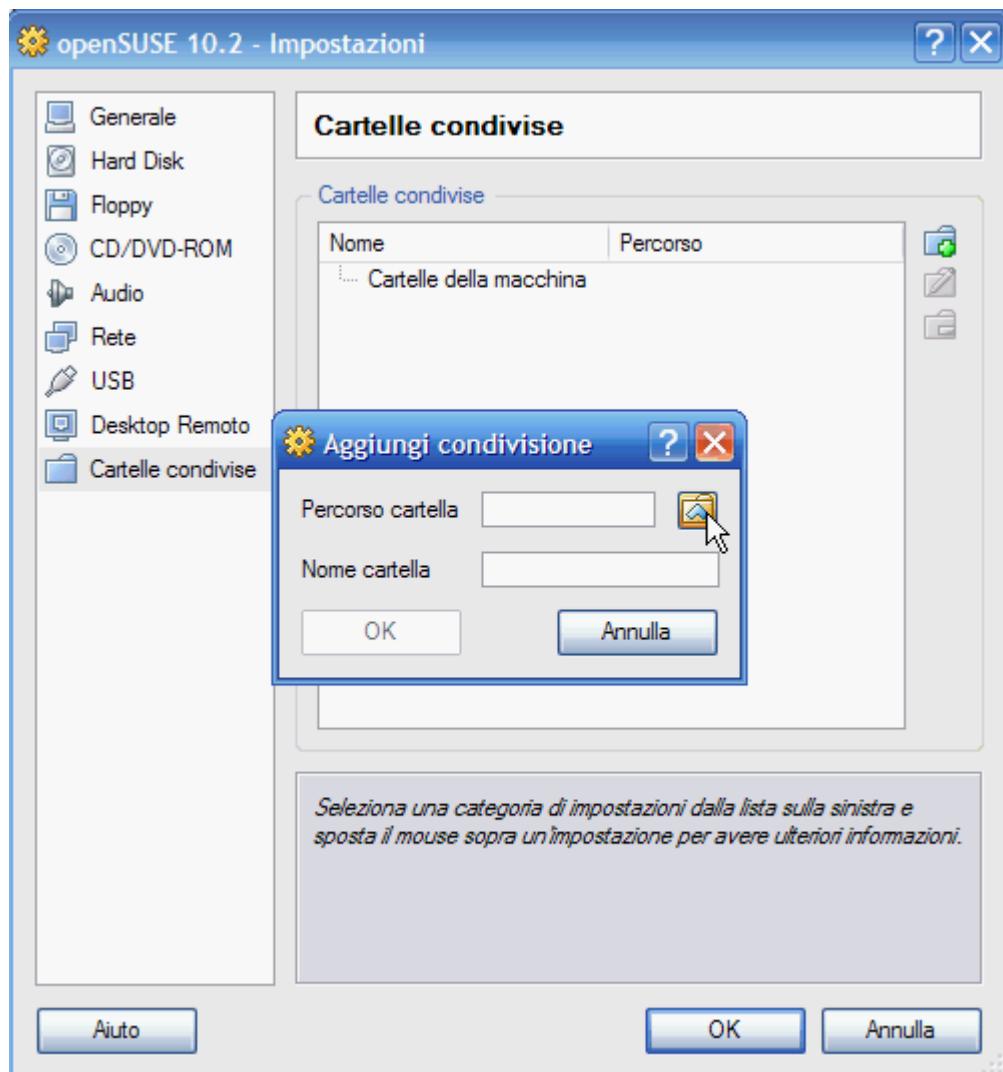


fig. 2

Quando avrete terminato, cliccate su OK. (fig. 3).

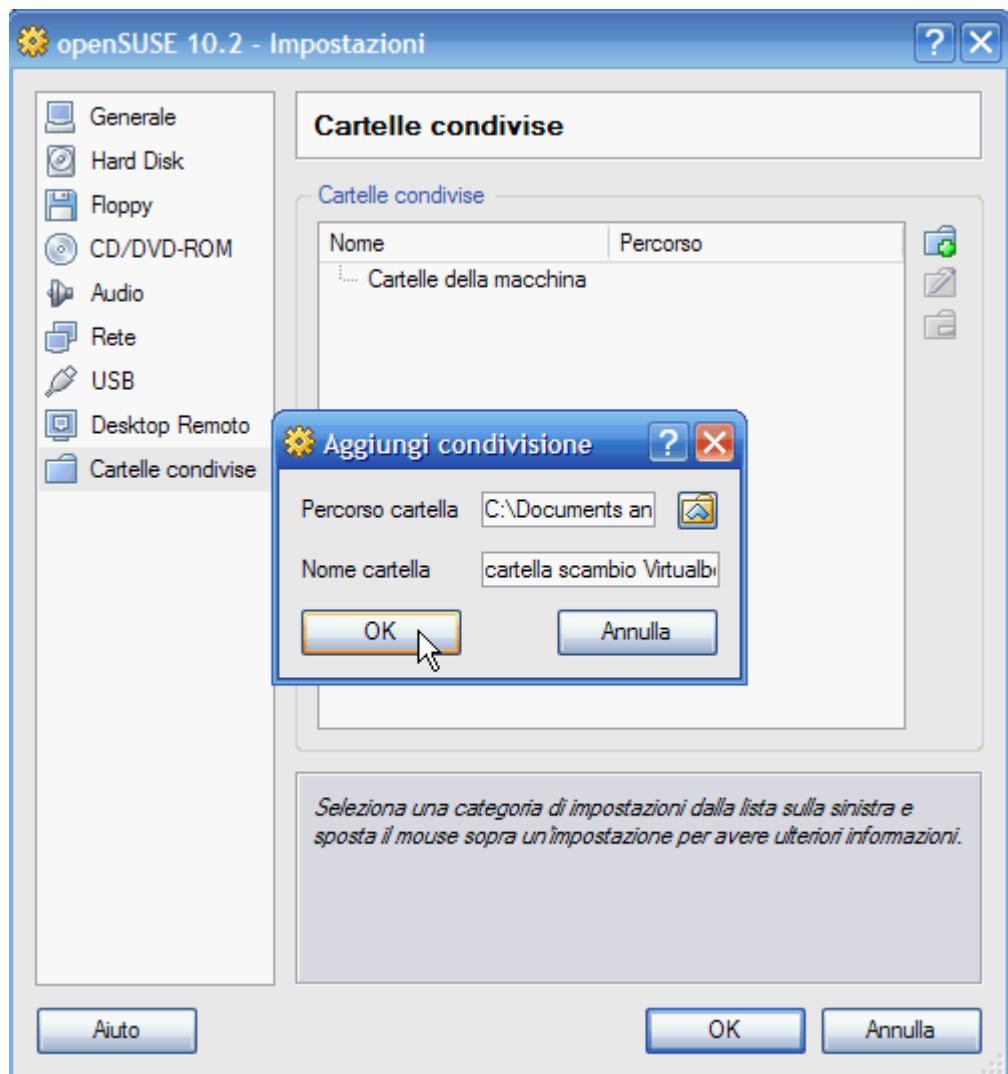


fig. 3

Cliccate ancora su **OK** (fig. 4) per terminare.

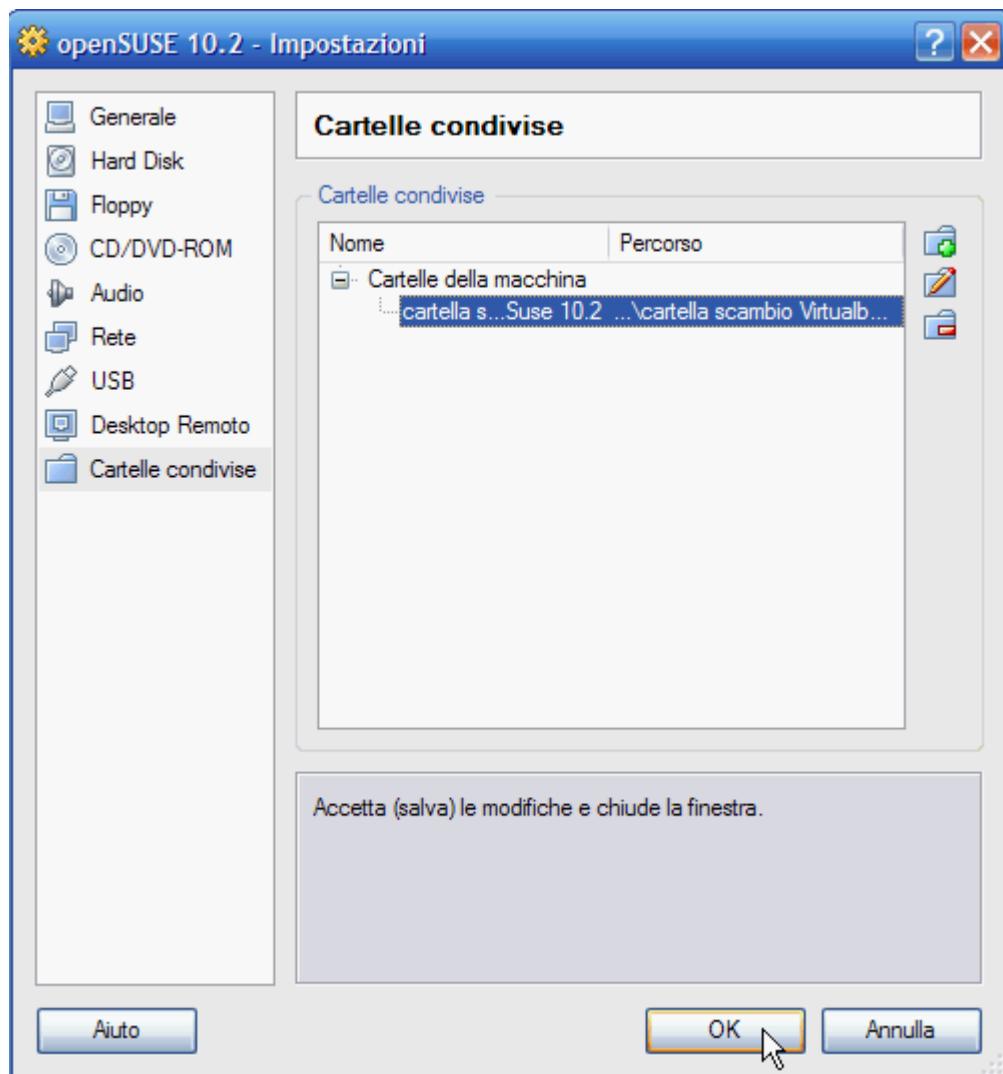


fig. 4

Benissimo, e soprattutto, complimenti: avete appena terminato la configurazione della vostra prima macchina virtuale con VirtualBox (**fig. 5**)!

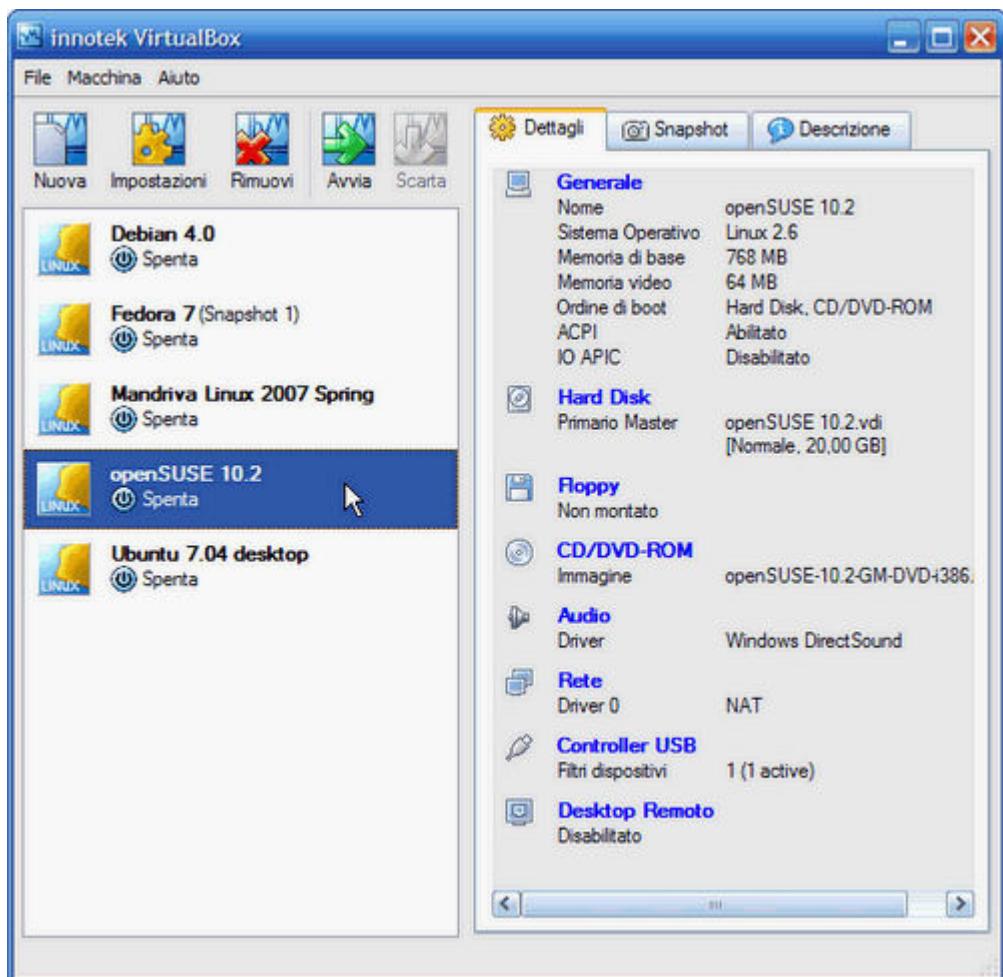


fig. 5

Ora non vi resta che avviare la macchina virtuale cliccando su **Avvia** (fig. 6).

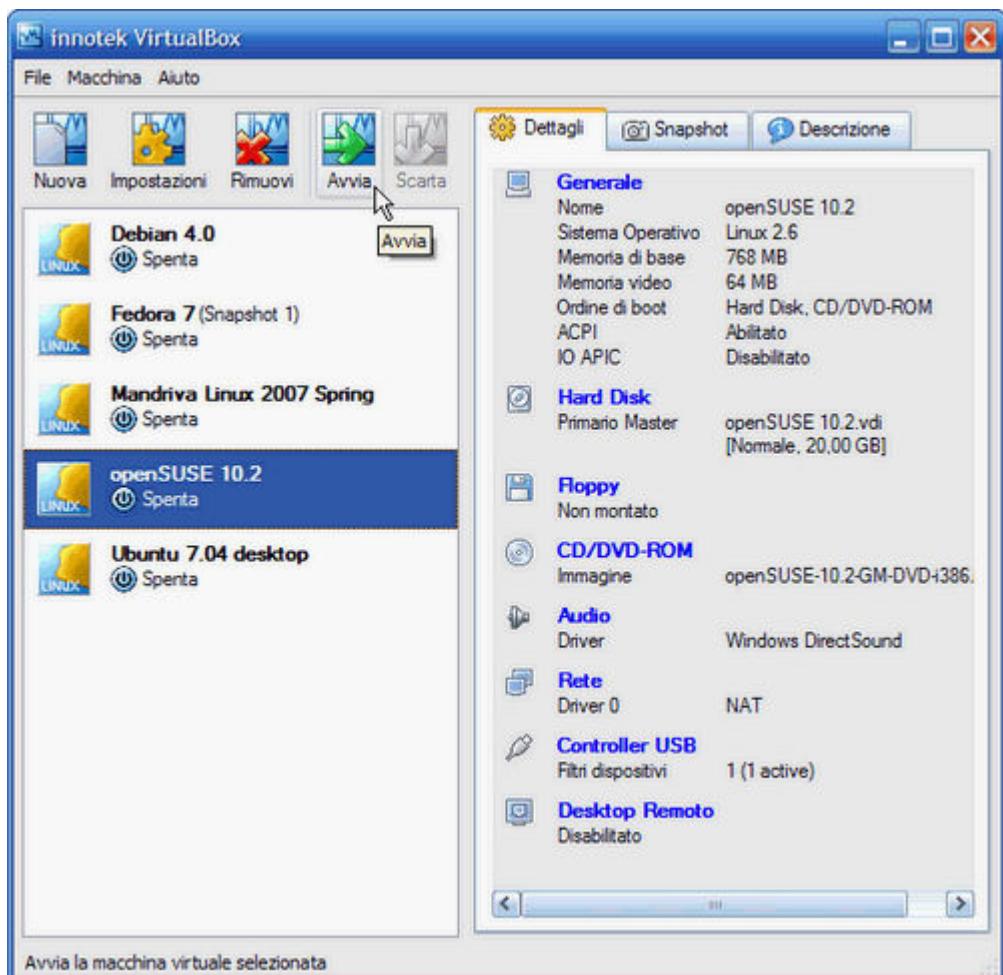


fig. 6

Il PC virtuale sta partendo (fig. 7), notate che, come un vero PC, potrete scegliere anche l'ordine di boot premendo F12

...



fig. 7

... non vi resta che iniziare ad installare il sistema operativo (fig. 8).

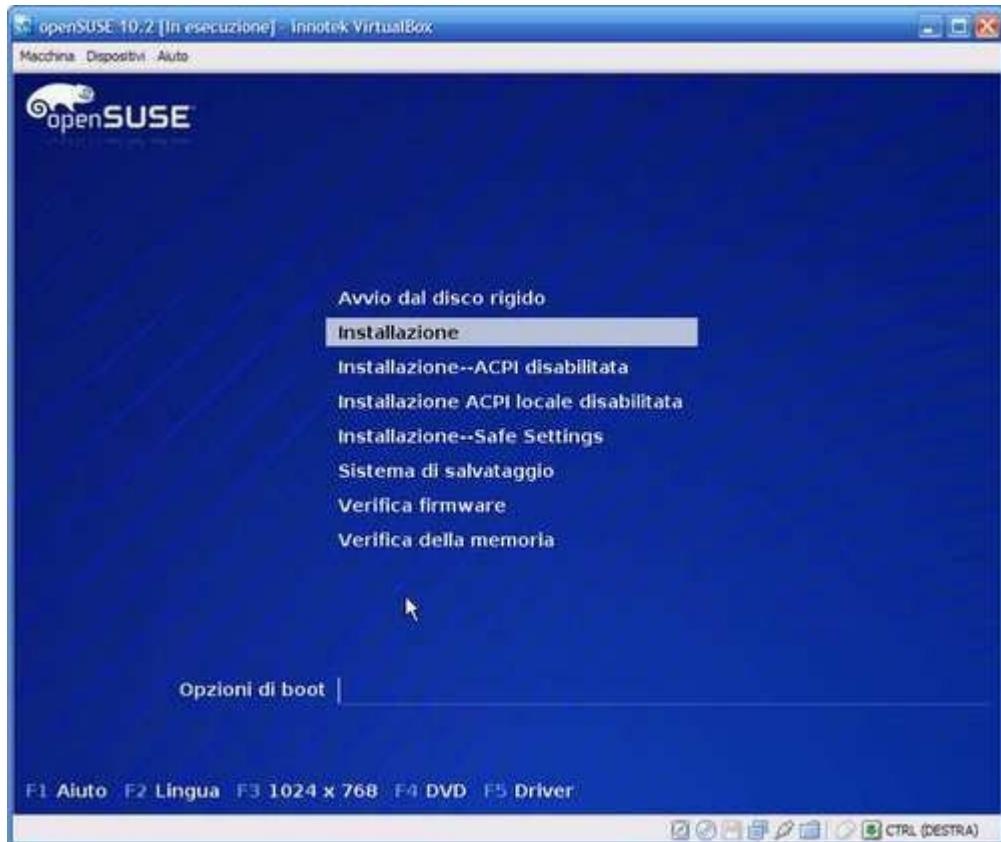


fig. 8

Buon lavoro!

@:\>

Aggiornamento di: Domenica 24 Giugno 2007

Mi segnalava Fabio nei commenti al post che, dopo aver terminato la configurazione della macchina virtuale e installato il sistema operativo guest, provando a condividere le cartelle con l'host, la condivisione non funziona. Come spiego nella risposta a Fabio, a questo punto della configurazione della macchina virtuale, ciò è normale. Pensavate di aver già finito tutta la configurazione, eh? Ma questa è solo una configurazione base. ;)

Ci sono ancora da installare le VirtualBox Guest Additions e vedere la configurazione avanzata della rete. Solo al termine di entrambi, tutte le possibilità di VirtualBox, e sono tante, saranno pienamente utilizzabili. Ma tranquilli, non dovete aspettare molto, i nuovi post sono quasi pronti. :)

Nel frattempo, date un'occhiata alla chiacchierata con Fabio: mi sono infatti accorto che non è possibile condividere una cartella qualunque del sistema host, occorre che abbia caratteristiche ben precise. Non so se si tratta di un vero e proprio bug, sicuramente è un'idiosincrasia di VirtualBox.

Buona lettura! :)

@:\>

Aggiornamento di: Martedì 17 Luglio 2007

Risolto il problema delle Cartelle condivise con un guest Windows, fate riferimento alla chiacchierata con baldo per i particolari.

@:\>

Pubblicato alle 15.29 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

6 commenti:

Michele Benvegnù ha detto...

Questo è un blog e non un forum, quindi non so dove inserire il mio intervento; lo inserisco qui come commento, in quanto l'argomento riguarda Virtual Box.

Credo che l'utilizzo più frequente dei software di virtualizzazione sia, per gli utenti Microsoft, sperimentare i sistemi Linux; tuttavia è interessante anche installare guest Windows, per testare programmi senza sporcare il PC originale. Io ho installato Vista come guest sul mio host Vista (Vista dentro Vista: scatole cinesi!).

Il problema che ho riscontrato è il seguente: VirtualBox emula la scheda di rete AMD PCNET Family Ethernet Adapter (PCI); ebbene, Windows Vista non supporta tale scheda di rete, o quantomeno non dispone nativamente dei relativi driver. Pertanto se installo Vista come guest, questo non riconosce la scheda di rete e pertanto non riesco a navigare in internet.

SOLUZIONE

1. scaricare i driver dal sito del produttore: (http://www.amd.com/us-en/assets/content_type/utilities/V4.51.zip)
2. aprire lo zip e quindi la cartella WinXP_SignedDriver, masterizzare i 3 file in essa contenuti in un file ISO
3. avviare la macchina virtuale Vista
4. Gestione dispositivi, ▷ Sistema ▷ andare su Pannello di Controllo cercare l'adattatore di rete (sarà contrassegnato come non funzionante), installare i driver con una ricerca manuale (cerca il software del driver nel computer, indicando il percorso del Cd iso)

Fatto ciò, la scheda di rete viene riconosciuta e la connessione ad internet è immediata.

(da <http://forums.virtualbox.org/viewtopic.php?t=224>)

22 giugno 2007 16.33

Giornale di Sistema ha detto...

Ciao Michele.

Vai tranquillo, la tua scelta di segnalare un problema e la sua soluzione, alla fine della serie di post sulla creazione di una macchina virtuale con VirtualBox, mi sembra perfetta! :)

E mentre sto scrivendo mi è venuta un'idea: creare un post in cui segnalare problematiche che abbiano anche trovato soluzione all'interno di commenti sparsi per il blog.

In questo modo si riesce a dare il giusto risalto, tanto alla soluzione, quanto a chi si è prodigato nel cercarla e ha avuto il desiderio di condividerla con tutti.

In effetti sì: la virtualizzazione in ambiente Windows è spesso usata per sperimentare distribuzioni Linux. Ritrovarsi con un Windows già pre-installato che occupa tutto il disco e pensare di mettersi a giocare con le partizioni per ridimensionarle, installare la distribuzione Linux, accorgersi in tutto questo di aver rischiato i propri dati e neanche ci piace, beh, magari lo facciamo una volta, ma non di più. E qualcosa, prima o poi, capita. Lo sa bene un mio collega di lavoro: nel tentativo di sfruttare un'oretta di tempo prima di andare in discoteca, installando una Ubuntu 7.04 (se ricordo bene), beh, un click di troppo, complice la fretta e la distrazione, e si è fumato tutto il disco rigido.

Certo, se poi l'intenzione è quella di provare prodotti Windows, allora non è da disdegnare anche un'altra possibilità di cui ho spesso parlato: SVS (Software Virtualization Solution). Per Vista esiste solo in versione 32 bit, ma indubbiamente ha dei grossi vantaggi quando è possibile utilizzarlo. Prima di tutto la velocità: sottraendo solo un 5-6% di risorse di CPU, i programmi sono comunque più performanti che l'analogo virtualizzato su VirtualBox e affini. Anche la trasportabilità è un elemento da non sottovalutare: un layer SVS contenente l'intera installazione di OpenOffice.org è grande poco più di 200 MB. Entra comodamente in un chiavetta USB e si può pensare tranquillamente di portarsi dietro il solo layer SVS fra varie macchine su cui si lavora. Portarsi dietro l'intera installazione di un sistema operativo più un applicativo, è sicuramente meno agevole.

Ma, al solito, sono entrambi due possibili soluzioni, resta solo da guardare il contesto per rendersi conto di quale è preferibile in quel momento.

Un'ultima cosa prima di salutarti: grazie per la soluzione che hai segnalato. :)

Ciao ciao!
@:>

22 giugno 2007 20.22

Fabio ha detto...

Ciao, ho seguito con interesse i 3 articoli ed aspettavo il quarto per capire come rendere operative le directory condivise tra host e guest. Nel mio caso host=XPpro e guest=XPpro. Avevo già eseguito i passaggi che hai illustrato nel quarto articolo, ma mi manca la fine. Mi spieghi: entrando nella macchina guest XPpro, non riesco a visualizzare le cartelle dell'HOST che ho condiviso come hai spiegato. Dove si dovrebbero trovare? Andando (nella macchina GUEST) su risorse di rete -> tutta la rete -> virtualBox Shared Folder, ottengo solo un messaggio di errore "impossibile sfogliare la rete. richiesta non supportata". Sto usando 1 rete NAT. Riesco a navigare in Internet.

Grazie.
Fabio

23 giugno 2007 16.51

Giornale di Sistema ha detto...

Ciao Fabio.

Ti ringrazio per la domanda, mi ha fatto render conto che nel post ho omesso un'informazione importante: così com'è configurata la macchina virtuale, ancora non è pronta per le "Cartelle condivise", esattamente come succede per gli "Appunti condivisi".

In entrambi i casi occorre che siano installate le VirtualBox Guest Additions, ma per la Cartelle condivise occorre anche una diversa modalità di rete, quella chiamata "Interfaccia Host". E neanche questo basta. Il nome della cartella condivisa in VirtualBox non deve avere spazi (nome cartella in fig. 2) ed il suo percorso deve essere qualcosa del tipo C:\nome cartella (percorso cartella sempre in fig. 2, in questo caso non è importante che non vi siano spazi), in altre parole non puoi condividere sottocartelle, ma solo cartelle principali.

Insomma, considera l'attivazione della condivisione cartelle solo preventiva, così come quella degli appunti condivisi, solo quando saranno online i post sull'installazione delle VirtualBox Guest Additions e sulle modalità avanzate di rete (già da domani si comincia), il quadro sarà finalmente completo e si potrà procedere anche

all'attivazione delle cartelle condivise.

Nel frattempo, dai un'occhiata alla configurazione della tua macchina virtuale: il nome della cartella host condivisa in VirtualBox appare senza spazi? Se è come in fig. 3 allora correggila, quello scelto di default da VirtualBox purtroppo è errato. Usa un nome a tua scelta, l'importante è che non abbia spazi, al limite sostituisci con il carattere underscore "_" o trattino basso altrimenti detto.

Ok, a presto allora, abbiamo ancora tanto lavoro da fare, ma chi la dura, la vince! :-D

E naturalmente aggiorno il post per segnalare la nostra chiacchierata, tornerà utile a molti. Grazie. :)

24 giugno 2007 16.21
baldo ha detto...

Ciao a tutti

sto provando Virtualbox da un paio di giorni e stavo "tribolando" anch'io con le cartelle condivise tra xpro host e xpro guest e ho fatto diverse prove: attivare le cartelle condivise non è servito a nulla :-(allora ho anche provato ad aggiungere una risorsa di rete nel guest, ma ancora non c'ero ...poi il colpo di fortuna, prendo la cartella desktop dell'host e attivo una normalissima condivisione in lettura/scrittura ed ecco che nelle risorse di rete del guest (chiaramente dopo aver installato le additions e attivato nei driver di rete l'"interfaccia host" magicamente è comparsa l'icona "desktop" dell'host tranquillamente utilizzabile con le sue sottocartelle :-)))scusate la spiegazione confusa e salutoni (ma a questo punto la scelta in virtualbox delle cartelle condivise a cosa serve?)

12 luglio 2007 8.52
Giornale di Sistema ha detto...

Ciao baldo.

Mi dispiace tu ti sia imbattuto in queste difficoltà, ed in effetti la domanda "sorge spontanea": se le cartelle fra host e guest possiamo condividerle via rete come hai fatto tu, a che serve farlo con l'apposita voce "Cartelle condivise"?

L'idea dei ragazzi dalla innotek credo fosse (posso solo ipotizzare naturalmente) proprio quella di semplificare la condivisione delle cartelle. Così, dopo aver sistemato un pò di cose, ho ripreso in mano tutta la questione e ho fatto prove più approfondite: beh, ne ho scritte di castronerie nel commento precedente (di cui chiedo venia a tutti i lettori del blog), segno evidente di quanto avessi la testa da un'altra parte.

A rettifica di quanto dicevo nel commento precedente, l'unica cosa che resta valida è questa: il nome della cartella condivisa in VirtualBox non deve avere spazi (nome cartella in fig. 2).

Ed infatti, con questo unico prerequisito soddisfatto, le cose funzionano egregiamente anche se la rete non è stata attivata nella configurazione della macchina virtuale: sì, esatto, anche senza la rete, la cartella host condivisa è comunque visibile anche nel guest!

Ed ecco che allora i conti tornano e la mia ipotesi sui ragazzi della innotek è verificata. ;-)

Spiegherò tutto per filo e per segno in un prossimo post (a proposito: ho sistemato in qualche modo le mie cose e sto per ricominciare ad aggiornare il blog :-D), ma se vuoi portarti avanti, dopo aver attivato la condivisione cartelle e installato le Guest Additions, procedi così:

- nel sistema guest apri una finestra "Prompt dei comandi", la trovi in Accessori;
- dai questo comando: net use X: \\vboxsvr\nome_cartella_condivisa_HOST

dove X è il nome dell'unità che verrà assegnato alla cartella condivisa; nome_cartella_condivisa_HOST è invece il nome della cartella condivisa in VirtualBox (nome cartella in fig. 2). Dopo pochi istanti vedrai comparire la cartella con la lettera di unità prescelta in Esplora Risorse.

@:>

17 luglio 2007 22.58

martedì 26 giugno 2007

VirtualBox Guest Additions: riflessioni mattutine

La mia maestra alle elementari (una donna d'altri tempi per cui provo ancora uno smisurato affetto e gratitudine) diceva sempre: se la sera non riuscite a imparare qualcosa, se tutto vi sembra confuso, lasciate perdere e andate a dormire. Il mattino dopo, come per magia, tutto vi sarà chiaro e cristallino.

Ieri sera rileggevo il post pubblicato ieri sulle Guest Additions, e qualcosa continuava a non tornarmi; e così sul nuovo post che volevo mettere online oggi, più lo rileggevo e più mi suonava strano. Leggi che ti rileggi, alla fine mi sono arreso e sono andato a dormire.

Questa mattina, davanti ad una bella tazza di caffè, il miracolo! :-D

I due post erano un ghirigori di istruzioni, di cose da fare, un andare avanti per poi tornare indietro a fare qualcosa che già poteva esser fatta prima ed evitare così il problema. Insomma, era più un percorso ad ostacoli che una guida a come evitarli. Infatti avevo descritto passo passo come ero arrivato io ad installarle correttamente, con tutte le difficoltà incontrate ed il modo per superarle. Ma perchè complicare la vita anche a voi? Perchè non descrivervi direttamente la giusta sequenza di operazione che porta direttamente all'installazione corretta delle Guest Additions?

E così ho preso una decisione drastica!

Ripubblicare il post di ieri, "VirtualBox: installare le Guest Additions in SUSE, fase 1 e 2" dopo averlo opportunamente modificato (ecco perchè non è visibile in questo momento, ma tranquilli, già oggi lo sarà di nuovo) e modificare quello non ancora pubblicato "VirtualBox: installare le Guest Additions in SUSE, fase 3".

Il tutto per rendere i post una guida chiara, semplice, lineare, ed evitare le difficoltà in cui incappereste se come me, seguite passo passo il manuale di VirtualBox.

Grazie per la comprensione. :-)

Aggiornamento delle ore 11.00

Nel renderne più semplice e godibile la lettura, il post "VirtualBox: installare le Guest Additions in SUSE, fase 1 e 2" è stato diviso in due post separati; saranno entrambi online oggi. ;-)

@:)>

Pubblicato alle 9.29

martedì 26 giugno 2007

VirtualBox: installare le Guest Additions in SUSE, fase 1

Nel post "VirtualBox: installare le VirtualBox Guest Additions, procedura generale", avevamo visto la loro procedura d'installazione per una generica distribuzione Linux; in questo invece vedremo un esempio concreto: entreremo nel dettaglio della loro installazione su una distribuzione SUSE Linux 10.2 in configurazione di default, ossia installata senza modificare le scelte proposte da YAST.

La **fase 1** della procedura prevede l'attivazione al supporto dei moduli di kernel esterni e, almeno per SUSE in configurazione di default, anche l'installazione del compilatore GNU, quest'ultimo necessario per poter procedere nella fase 3, alla compilazione delle Guest Additions.

Cominciamo!

Per prima cosa **attiviamo il supporto dei moduli di kernel esterni**.

Avviate VirtualBox e fate partire la macchina virtuale con sopra la SUSE. Terminato il caricamento, quando appare il desktop cliccate su *Computer*, poi su *Software di installazione* (**fig. 1**).

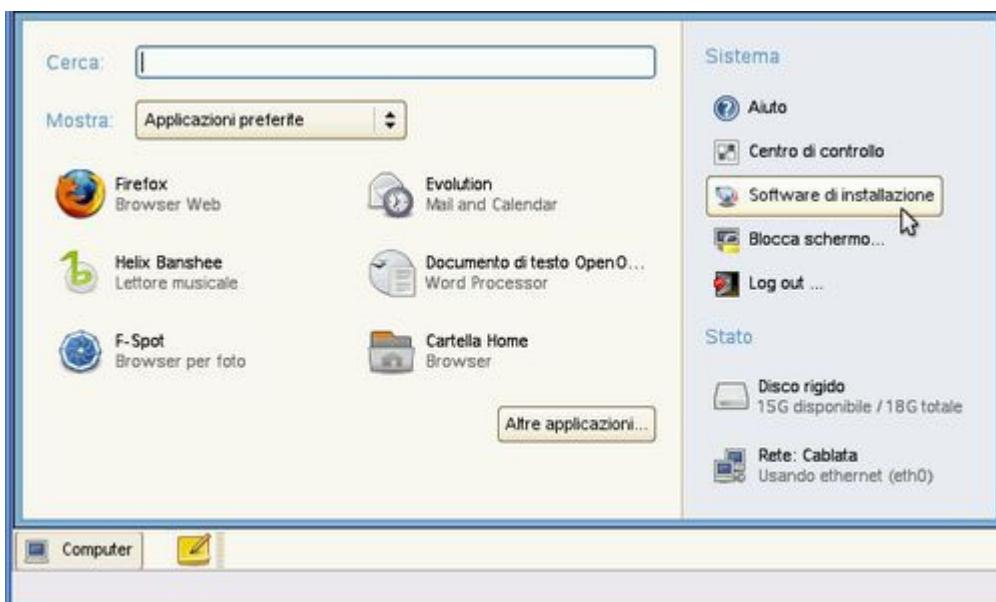


fig. 1

Nel box di ricerca scrivete "kernel" e cliccate su *Cerca*: compariranno tutti i moduli che nel nome hanno il termine kernel, selezionate *kernel-source* e *kernel-syms*, poi cliccate su *Install* (**fig. 2**).

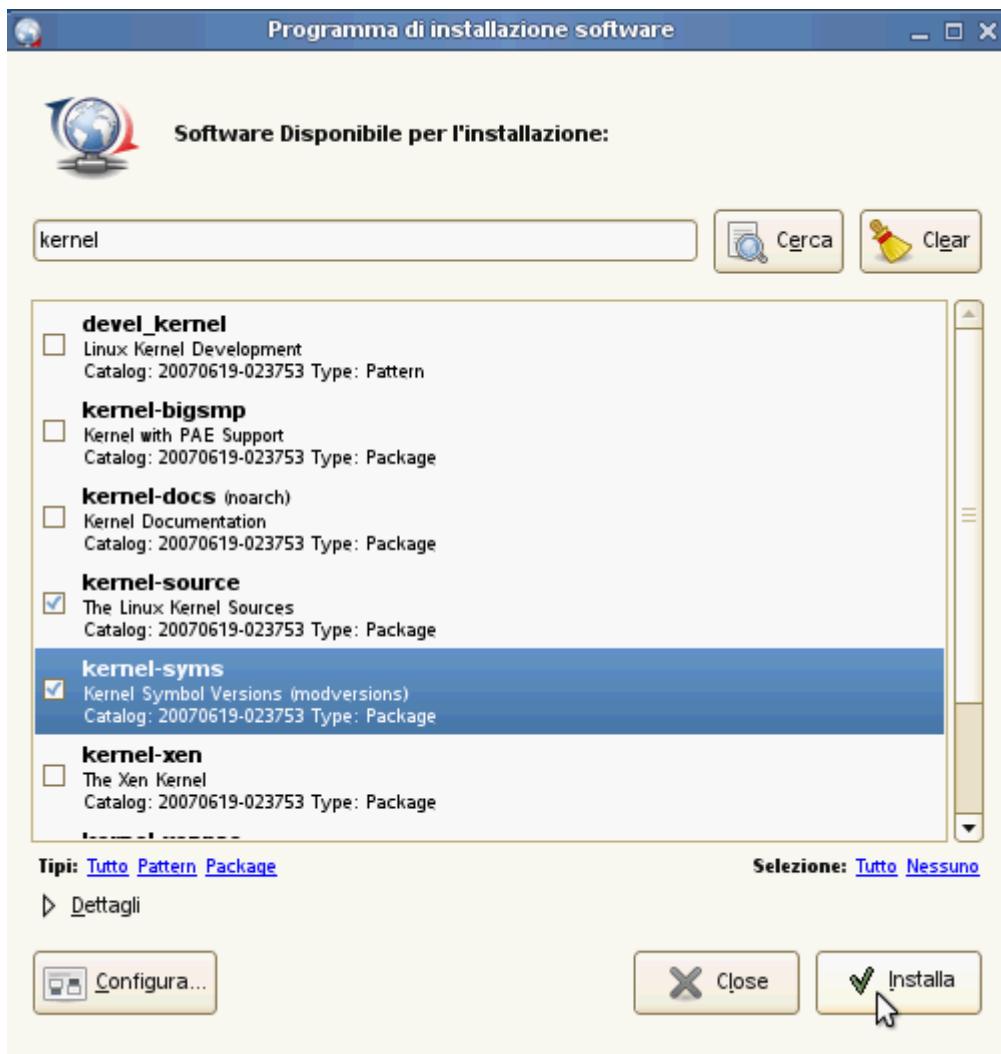


fig. 2

Se, come è probabile, ottenete un messaggio simile a quello di **fig. 3**, cliccate sul pulsante *Aggiungi utente privilegiato* per proseguire:

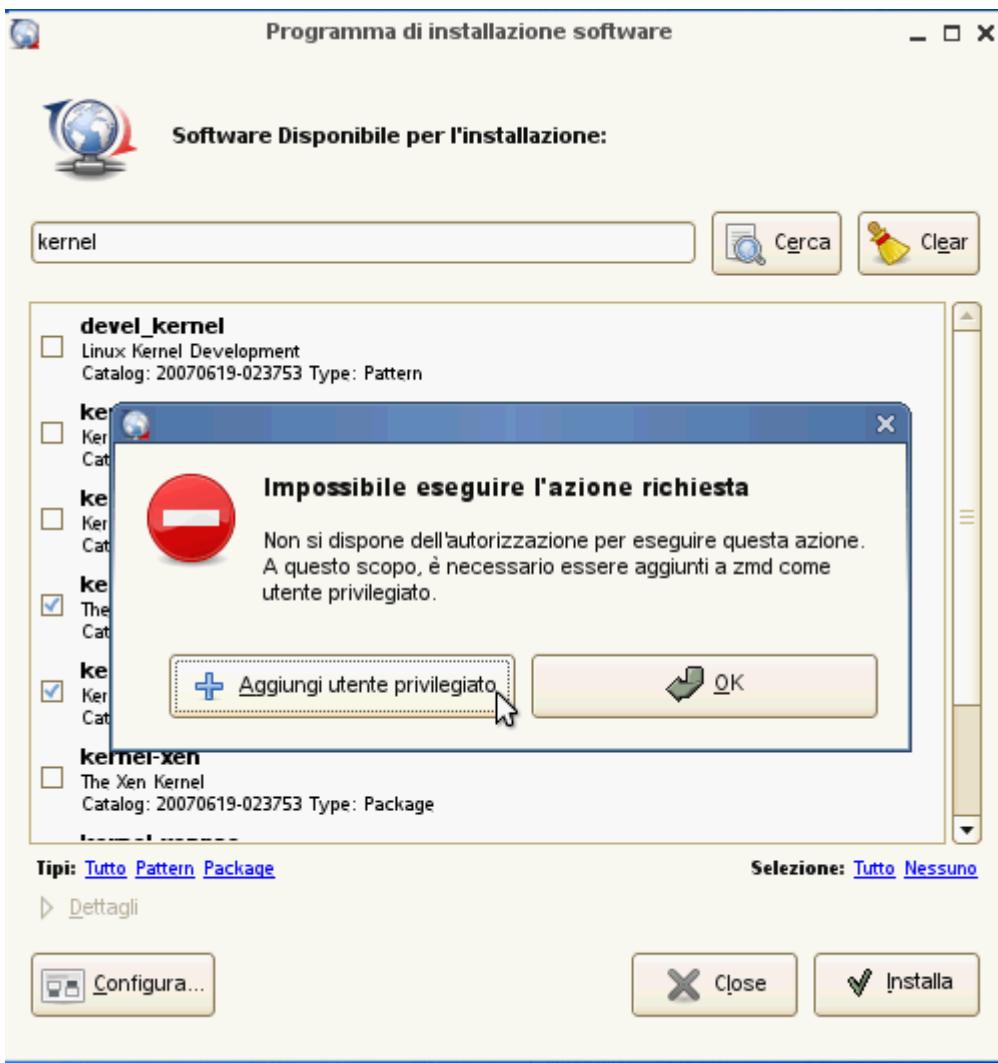


fig. 3

Vi verrà chiesta la password di root, inseritela e cliccate su *Continua* (fig. 4).

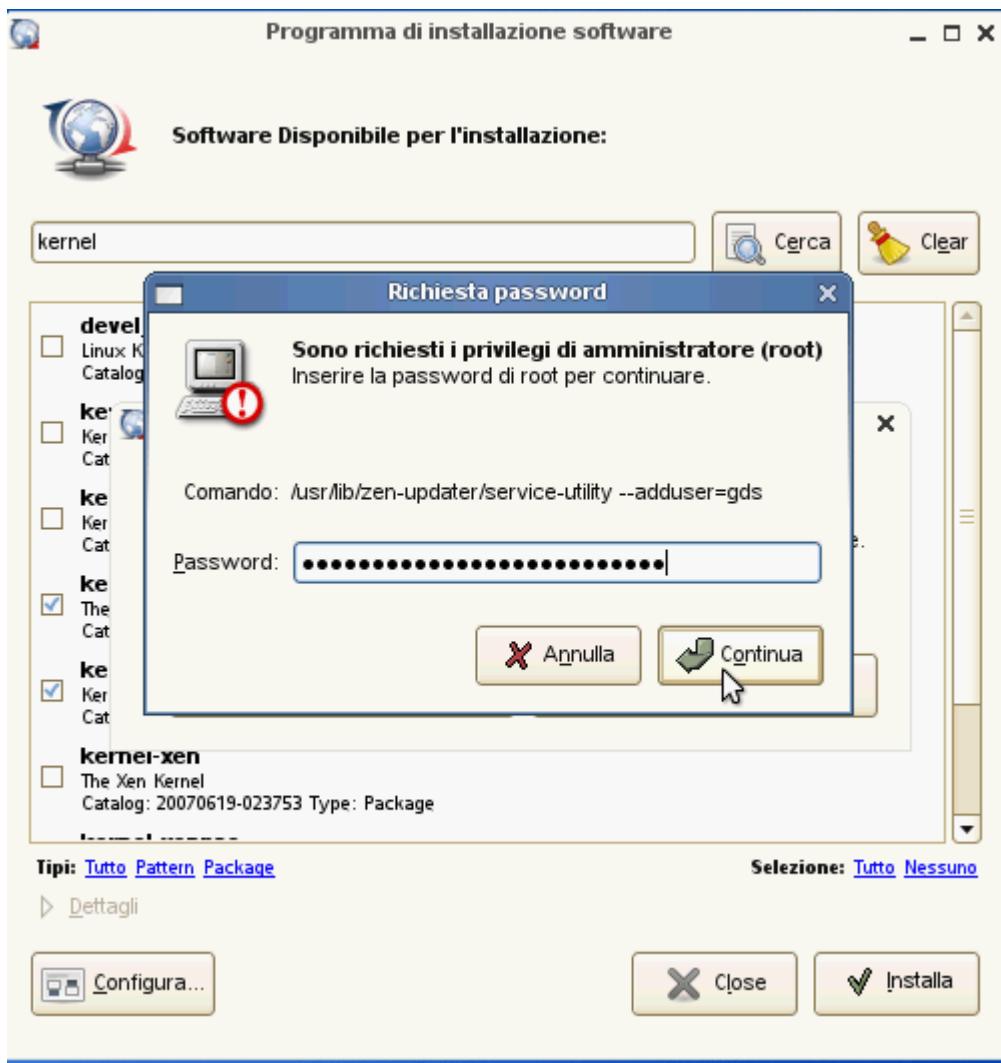


fig. 4

Se tutto si conclude correttamente, sarete stati aggiunti come utente a ZMD (ZENworks Management Daemon), pertanto ora potrete procedere con l'installazione dei moduli kernel; cliccate su *Chiudi*, inizierà il processo di installazione (fig. 5).

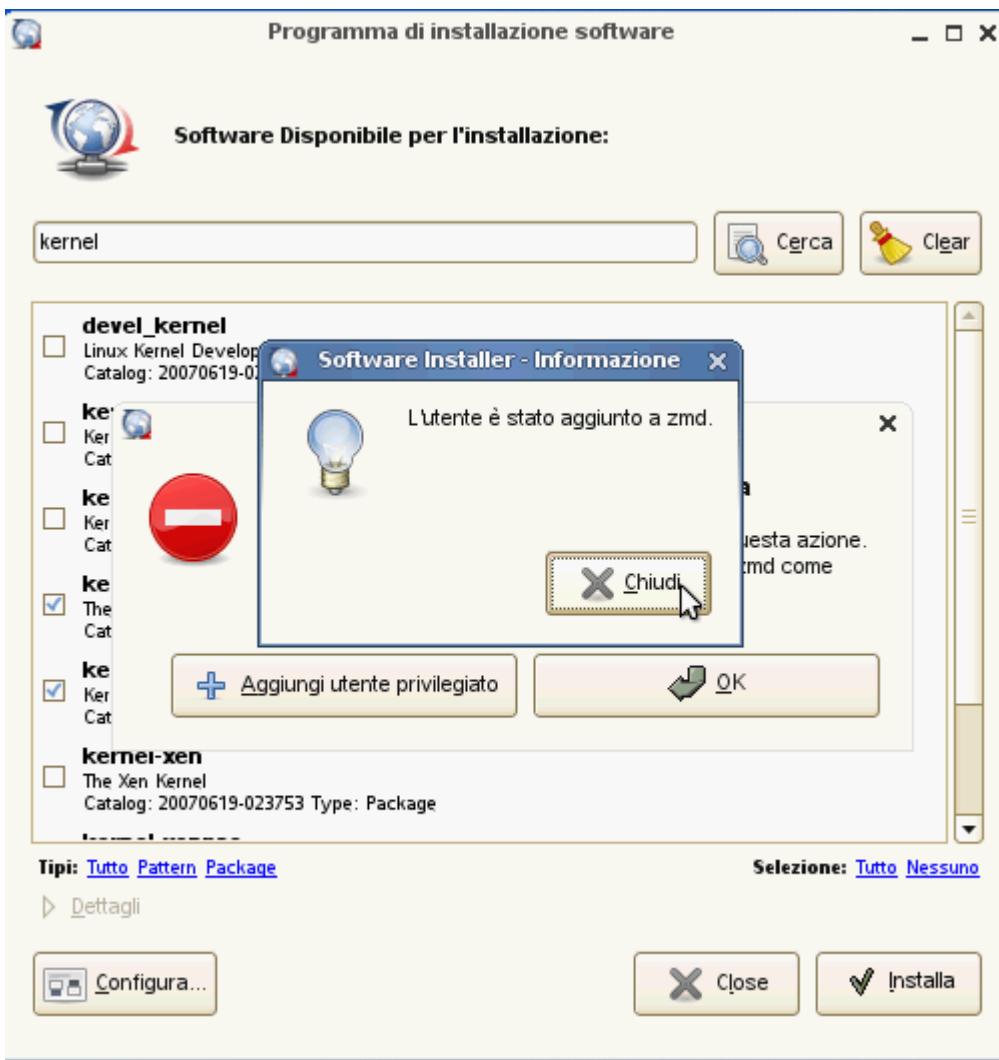


fig. 5

Quando l'installazione sarà terminata, cliccate sul pulsante *Chiudi* per terminare (fig. 6).



fig. 6

Ora che il supporto ai moduli di kernel esterni è installato, procederemo ad **installare il compilatore GNU**.

Se avete già chiuso la finestra del "Programma di installazione software", ricominciate da fig. 1, altrimenti, nel box di ricerca scrivete "gcc" e cliccate sul pulsante *Cerca*, selezionate la voce "gcc" e cliccate su *Installa* (fig. 7).

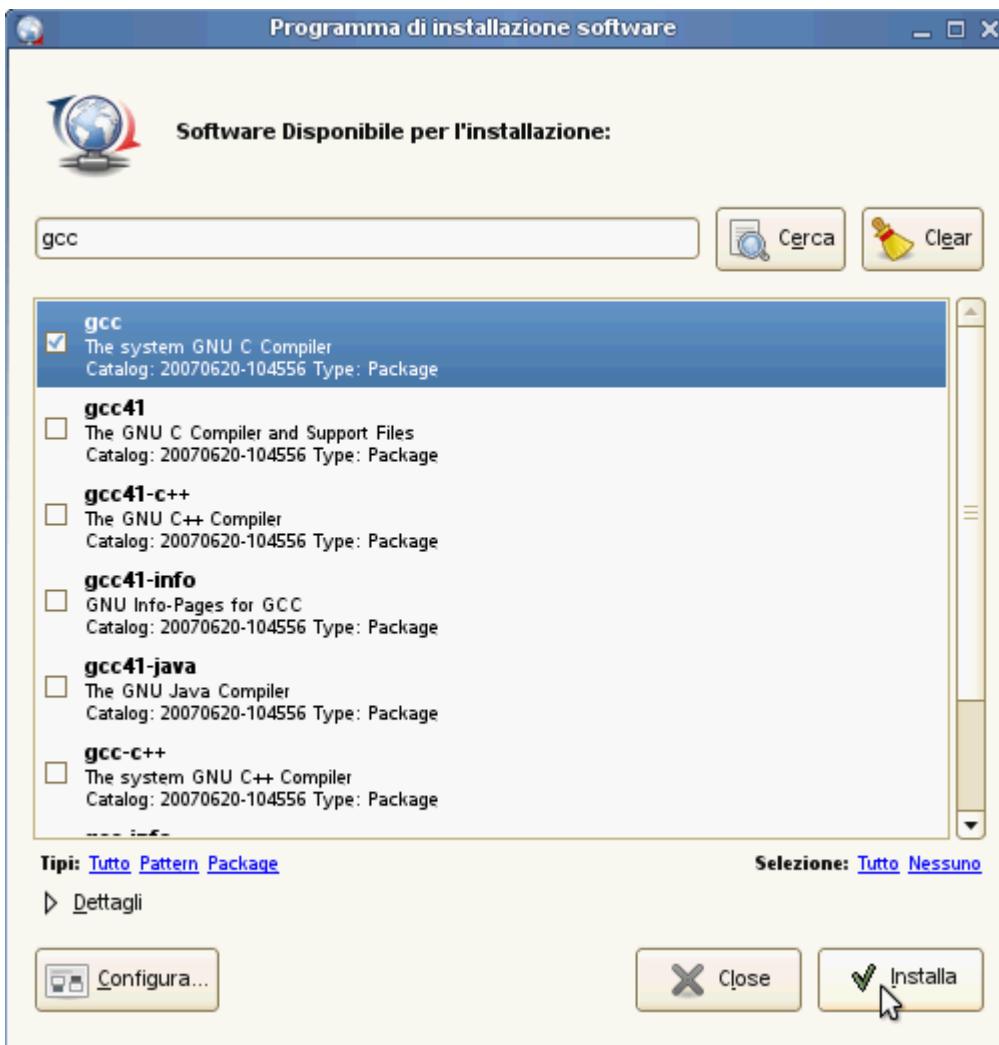


fig. 7

Il programma di installazione del compilatore GNU cercherà e vi mostrerà tutte le dipendenze, ossia i programmi aggiuntivi da installare affinché tutto funzioni correttamente; accettate sulla fiducia quanto vi mostra e cliccate su *Applica* (fig. 8).

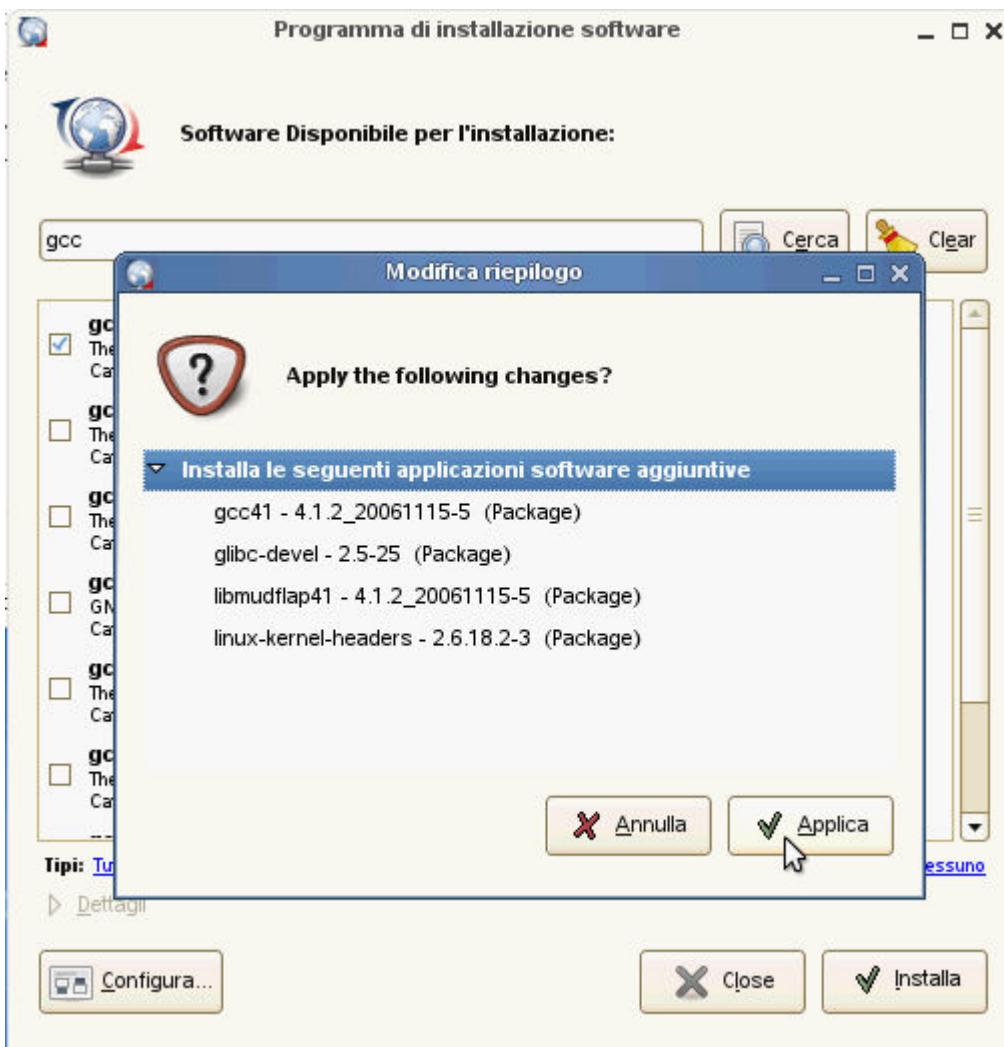


fig. 8

Quando l'installazione è terminata, chiudete tutto.

Bene, la fase 1, l'attivazione al supporto di moduli di kernel esterni e l'installazione del compilatore GNU, è terminata con successo.

Nel prossimo post, "VirtualBox: installare le Guest Additions in SUSE, fase 2", ci occuperemo di installare le Guest Additions sul vostro sistema cosicché nella fase 3, potremo finalmente compilarle e renderle attive.

@:>

Pubblicato alle 12.00 Argomenti: guida, Linux, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione

martedì 26 giugno 2007

VirtualBox: installare le Guest Additions in SUSE, fase 2

Nel post "VirtualBox: installare le Guest Additions in SUSE, fase 1" abbiamo visto:

- l'attivazione del supporto ai moduli di kernel esterni,
- l'installazione del compilatore GNU.

Proseguiamo ora con la **fase 2**, l'installazione delle VirtualBox Guest Additions.

Se è rimasto intrappolato dentro, tirate fuori il mouse dalla finestra della macchina virtuale (premete il tasto CTRL destra sulla tastiera), cliccate sul menù *Dispositivi* della finestra, poi su *Installa Guest Additions* (**fig. 1**).

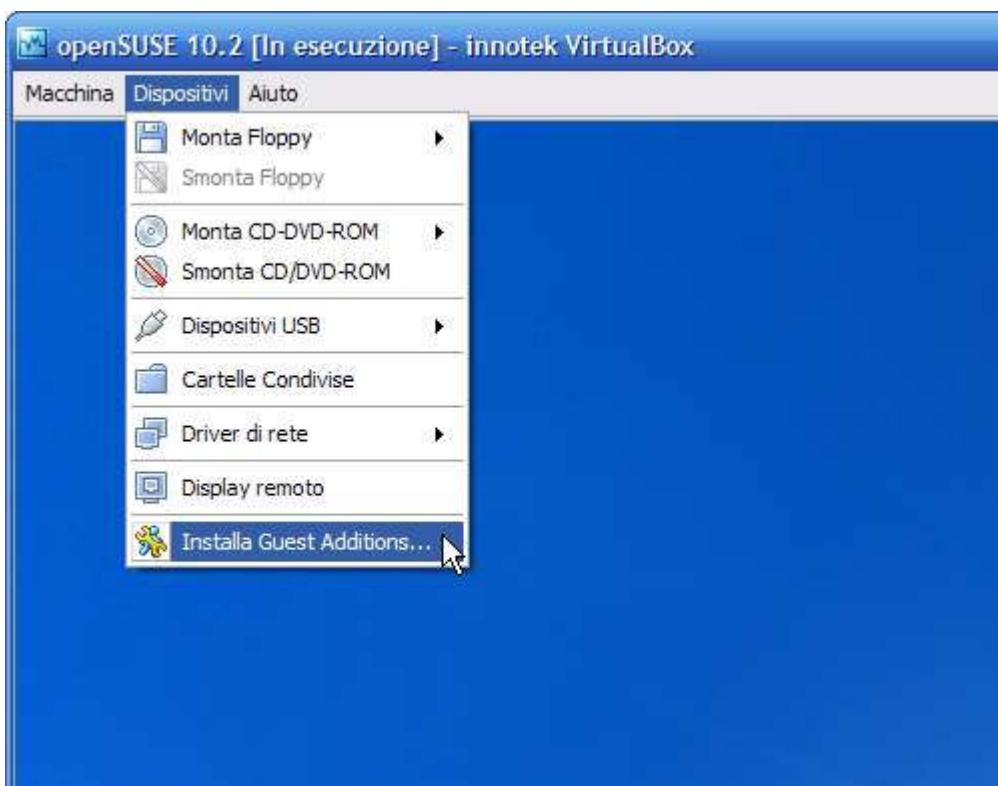


fig 1

Ora, se nel CD/DVD della macchina virtuale non vi è nulla, queste compariranno immediatamente sul desktop, se invece com'è probabile, è ancora caricato il file ISO d'installazione della vostra SUSE, allora sembrerà non accadere nulla; tra poco però vedremo che qualcosa, ancorché dietro le quinte, è avvenuto.

Intanto, provvedete a scaricare dal CD/DVD la ISO d'installazione in questo modo: sul desktop, cliccate con il tasto destro sull'icona del CD/DVD e poi su *Espelli* (**fig. 2**).



fig. 2

Aspettate che l'icona scompaia dal desktop, poi, dal menù *Dispositivi*, cliccate su *Smonta CD/DVD-ROM* (fig. 3).

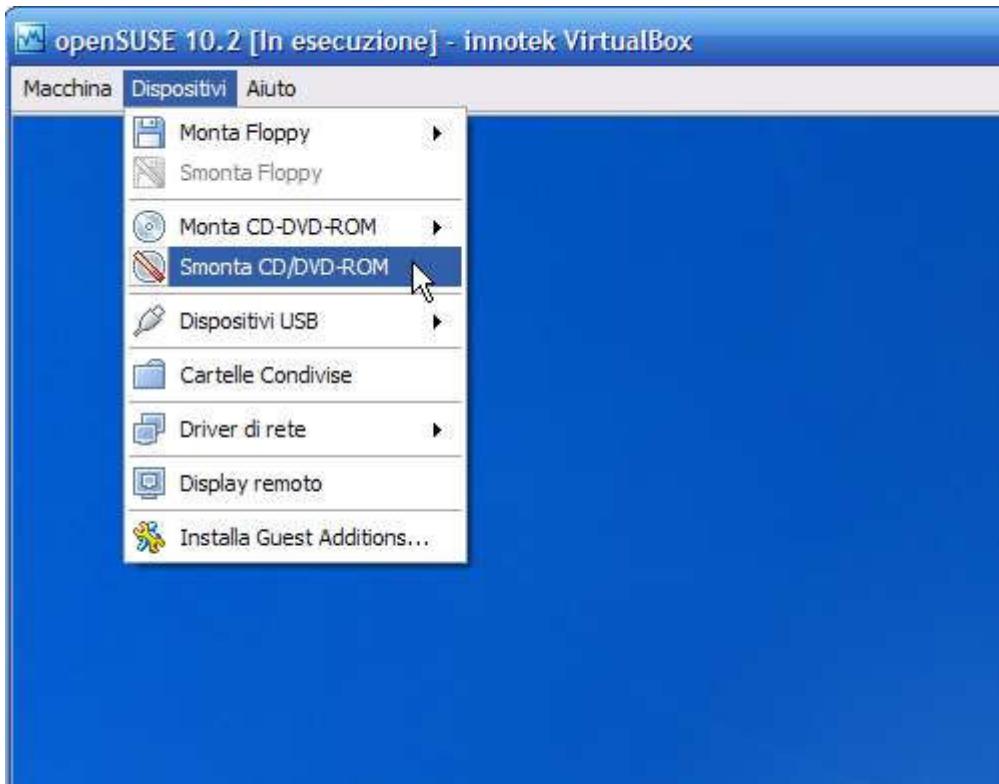


fig. 3

Ora che il lettore è vuoto e smontato, occorre rimontarlo caricandovi dentro l'immagine ISO delle Guest Additions. Cliccate su: *Dispositivi* --> *Monta CD/DVD-ROM* --> *Immagine CD/DVD-ROM* (fig. 4).

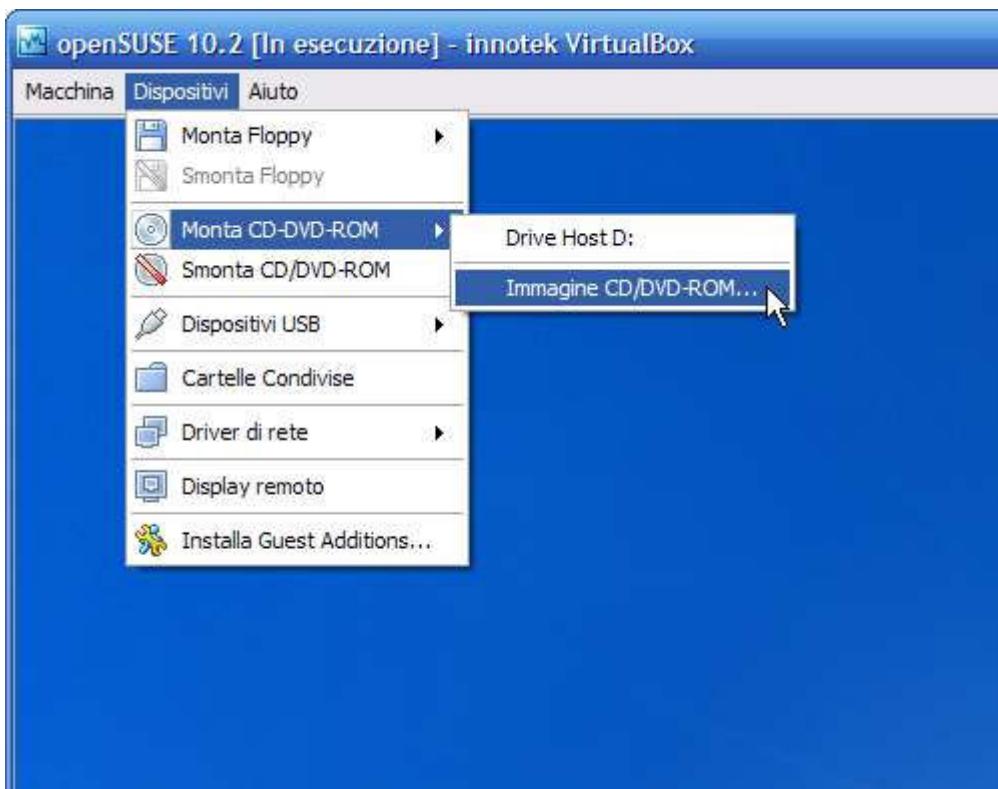


fig. 4

Si aprirà il Gestore dei dischi virtuali sul tab dedicato alla gestione delle immagini CD/DVD, date un'occhiata: è comparso un nuovo file ISO, "VBoxGuestAdditions.iso", che è stato reso disponibile nel Gestore dei dischi virtuali proprio quando avete cliccato su *Installa Guest Additions* e sembrava non fosse accaduto nulla. Cliccate su questo file e poi su *Seleziona* (fig. 5).

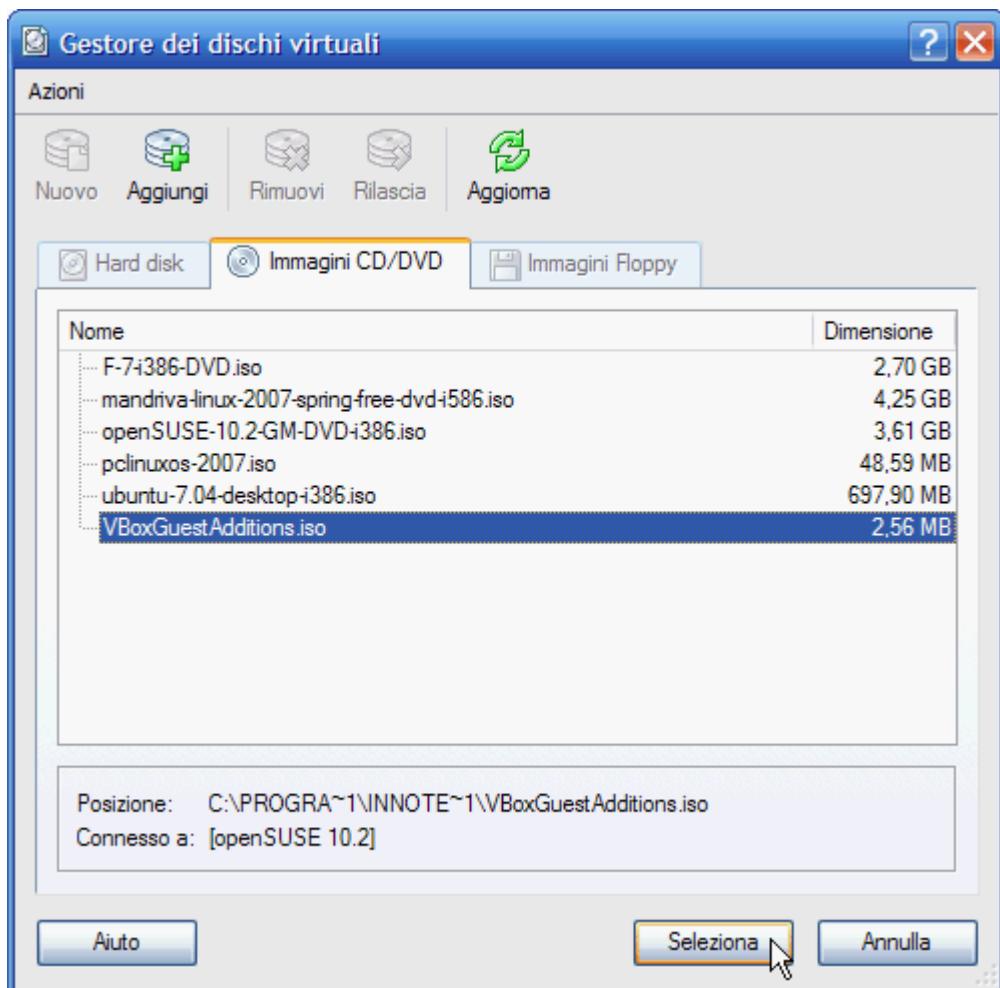


fig. 5

Pochi secondi, ed eccole: le VirtualBox Guest Additions sono state caricate nel lettore CD/DVD (fig. 6).

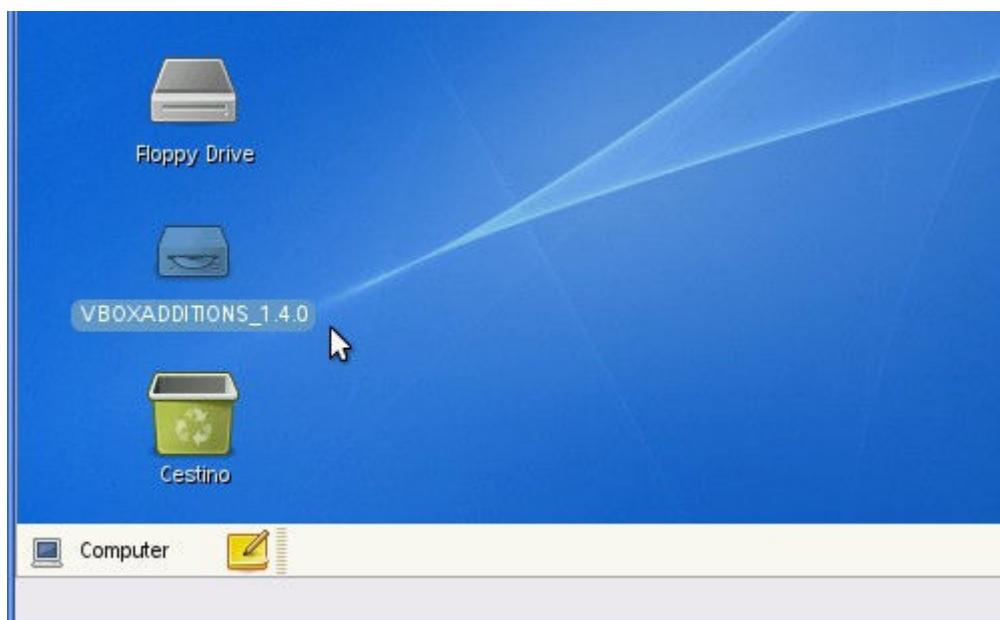


fig.6

Bene, anche la fase 2 è finita. Vi resta solo la fase 3, compilare le Guest Additions, e avrete terminato la loro installazione. Seguitemi nel prossimo post, "VirtualBox: installare le Guest Additions in SUSE, fase 3" per sapere come

fare.

@:>

Pubblicato alle 14.28 Argomenti: [guida](#), [Linux](#), [software](#), [tutorial](#), [Virtualbox](#), [VirtualBox Guest Additions](#), [virtualizzazione](#)

martedì 26 giugno 2007

VirtualBox: installare le Guest Additions in SUSE, fase 3

Nel post "VirtualBox: installare le Guest Additions in SUSE, fase 2" abbiamo visto comparire le Guest Additions sul desktop della vostra macchina virtuale. Non vi resta che compilarle e finalmente potrete godervene tutti i loro benefici.

Riprendiamo da dove eravamo: **fig. 1**.



fig. 1

Date un'occhiata al contenuto del file ISO: fate un doppio click sull'icona delle Guest Additions, vedrete qualcosa di simile a **fig. 2**; quello che a noi interessa è il file "VBoxLinuxAdditions.run".

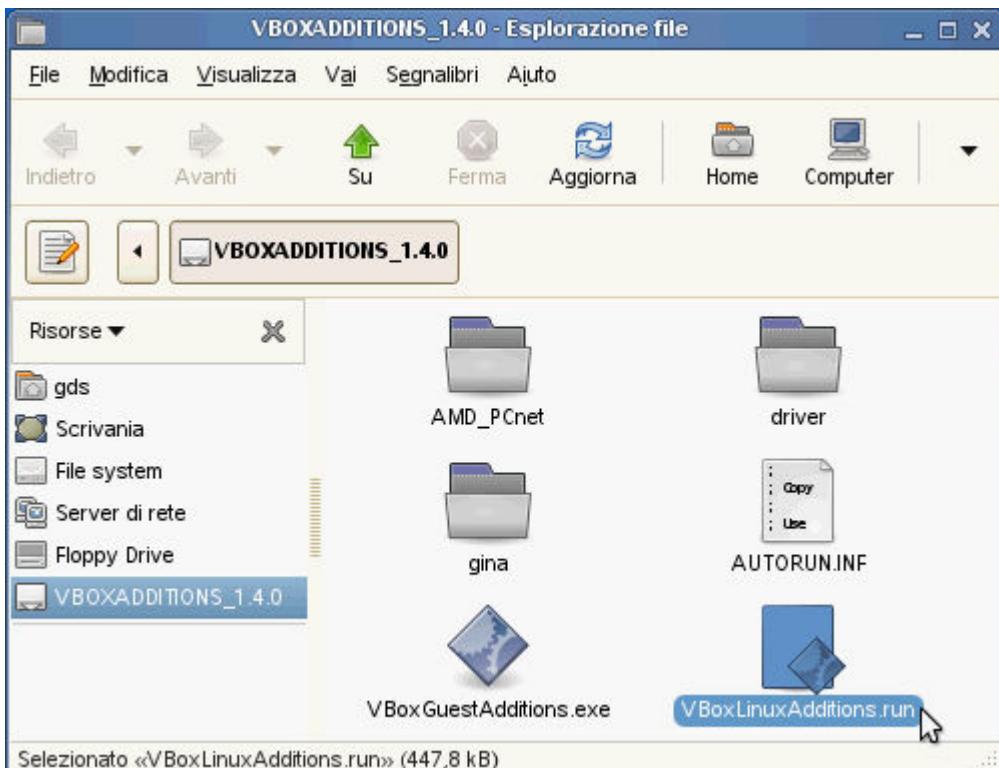


fig. 2

Cliccate sul menù *File* --> *Apri nel terminale* (**fig. 3**).

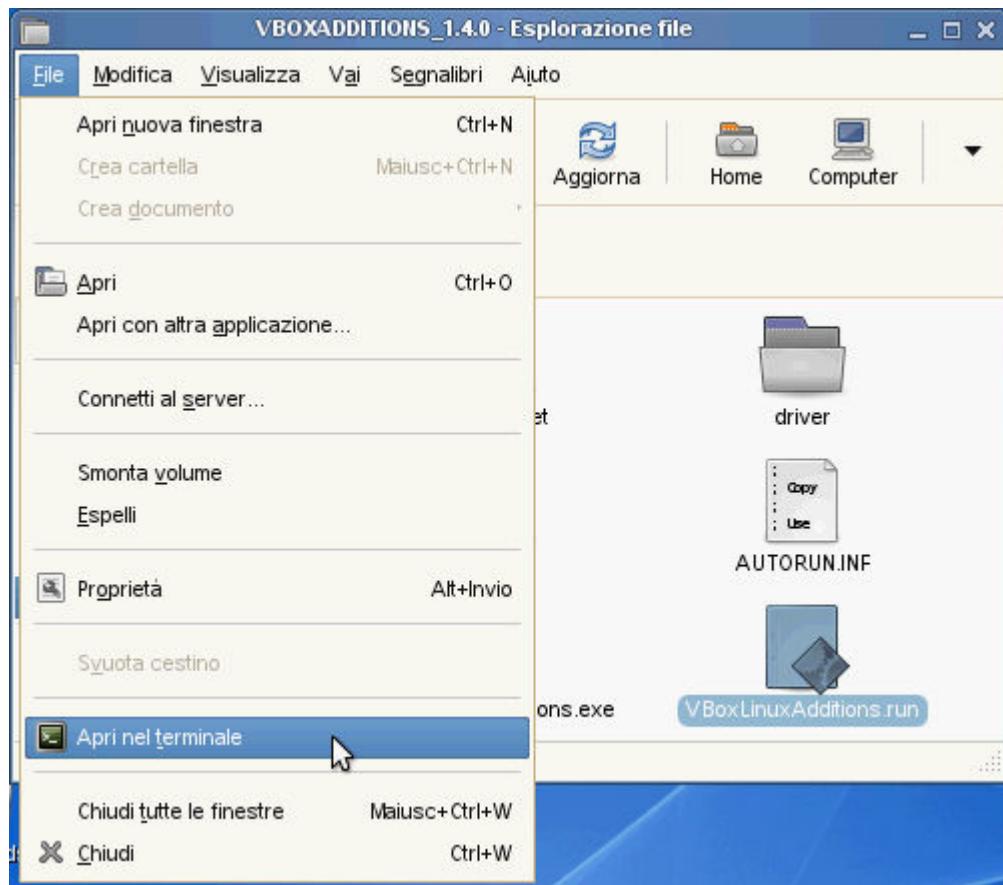


fig. 3

Si aprirà l'emulatore di terminale (**fig. 4**), scrivete il comando:

```
sudo sh ./VBoxLinuxAdditions.run
```

Mi raccomando le maiuscole/minuscole, Linux è case-sensitive.

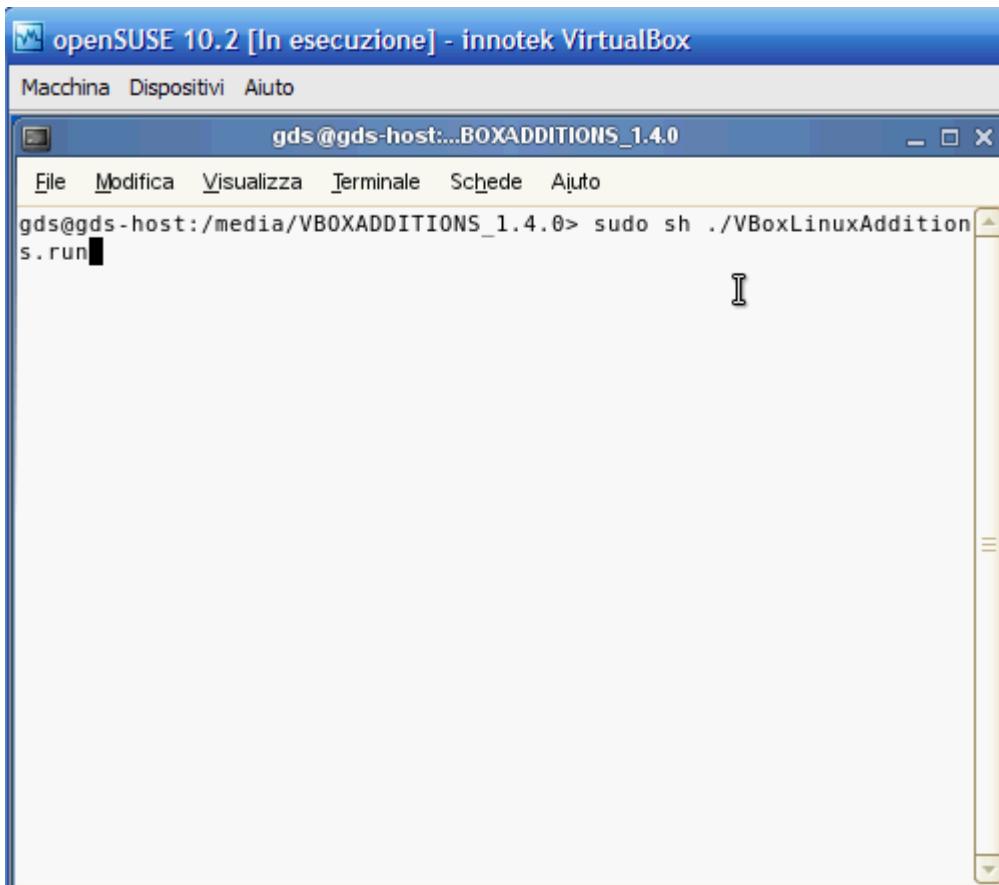


fig 4

Dopo avergli dato la password di root, se tutto è a posto partirà la compilazione della VirtualBox Guest Additions (fig. 5).

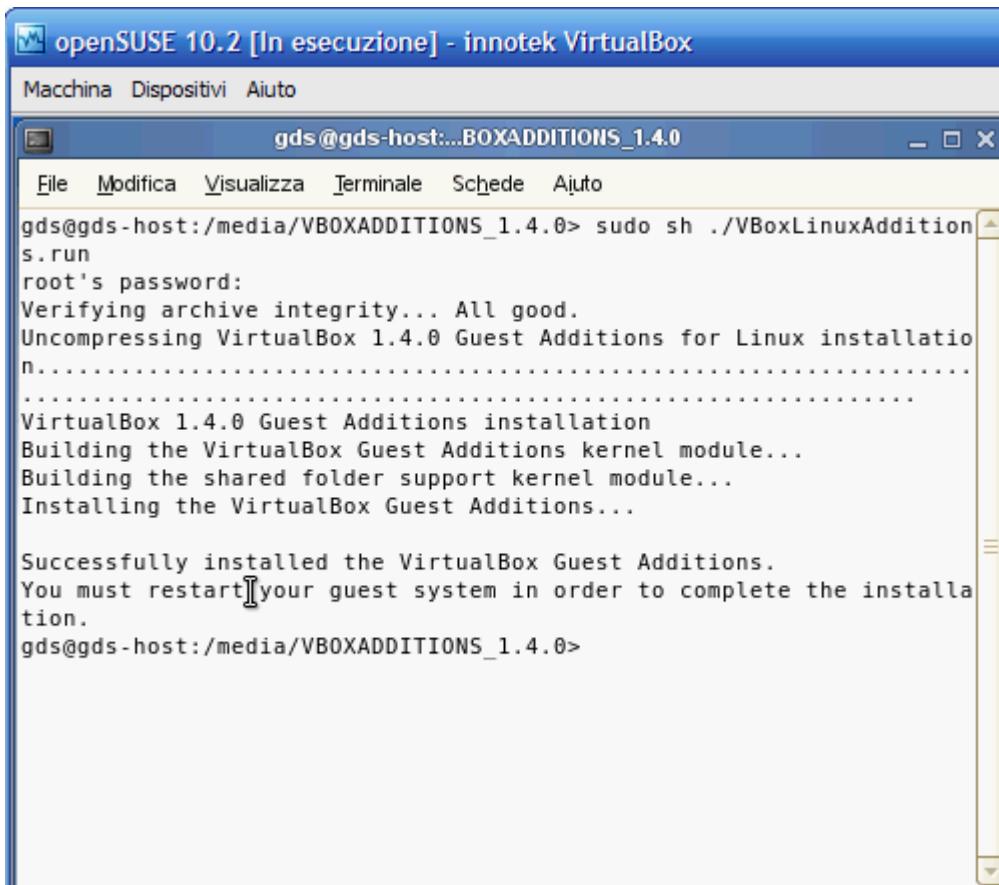


fig. 5

Complimenti!

Siete riusciti ad installare le VirtualBox Guest Additions sul vostro sistema SUSE.

Non vi resta che riavviare il sistema. Una volta fatto, potrete beneficiare fin da subito di maggiore velocità della scheda video, risoluzioni superiori, ancora più colori; il mouse non resterà più prigioniero nella finestra, gli Appunti condivisi (il copia-incolla fra guest ed host) saranno attivi e così via; fate delle prove e verificate che tutto funzioni.

Di una cosa però non potete ancora beneficiare: le Cartelle condivise.

Lo so, per alcuni è stata un pò una delusione, ma la configurazione base della macchina virtuale così come l'abbiamo vista nel post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 1)" ed i successivi 3 della serie, non è sufficiente. Occorre una diversa modalità di rete che vedremo a breve, e qualche altro piccolo aggiustamento nella macchina virtuale: solo allora potrete condividere in tutta comodità anche le cartelle dell'host.

@:\>

Pubblicato alle 17.29 Argomenti: guida, Linux, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione

mercoledì 27 giugno 2007

VirtualBox: Guest Additions in Windows XP

Occupiamoci adesso dell'installazione delle VirtualBox Guest Additions in Windows XP. Ricordatevi che le Guest Additions vanno installate sul sistema guest, non quello host; premesso questo, iniziamo!

Avviate VirtualBox, poi la macchina virtuale di Windows XP (**fig. 1**). Terminato il caricamento di Windows all'interno della finestra, sulla finestra cliccate *Dispositivi* --> *Installa Guest Additions*.



fig. 1

All'interno del sistema guest parte il wizard di installazione delle Guest Additions. Cliccate su *Next* per proseguire (**fig. 2**).

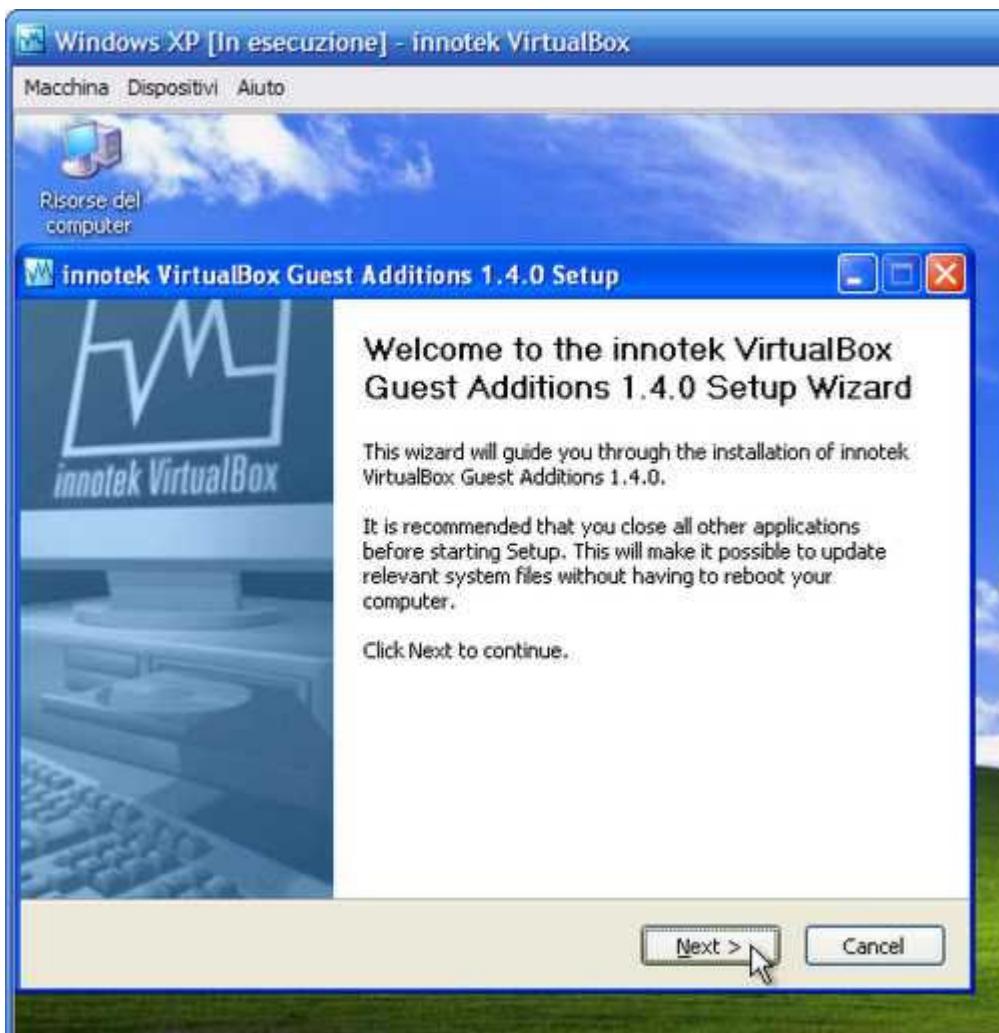


fig. 2

Vi verrà chiesto il percorso della cartella in cui installare le Guest Additions, se non avete esigenze particolare lasciate la scelta di default e proseguite cliccando su *Install* (fig. 3). Tenete a mente che la cartella programmi in cui vengono installate è quella all'interno del PC virtuale, non quella sull'host (nell'ipotesi che host e guest siano entrambi sistemi Windows).

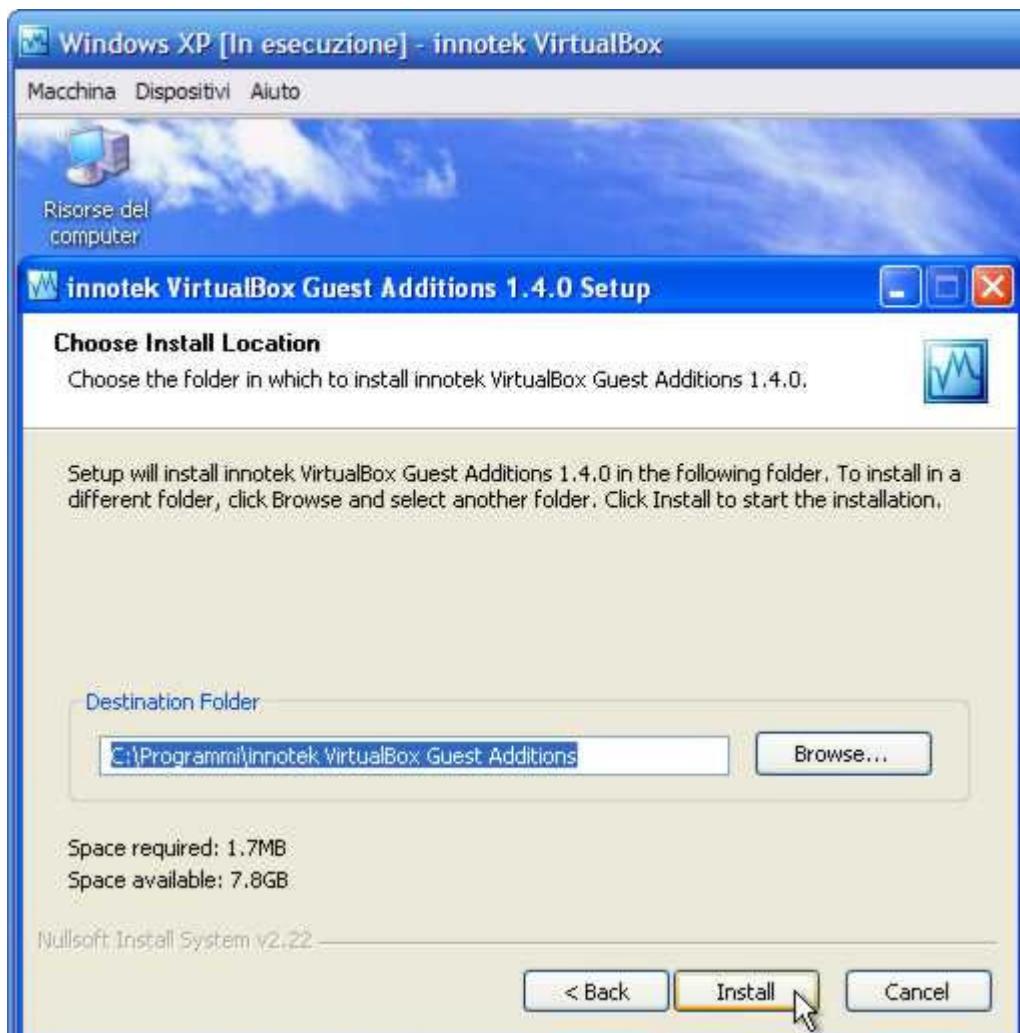


fig. 3

Partita l'installazione (**fig. 4**), questa copierà tutti i file nella cartella di destinazione e apporterà al sistema guest le modifiche opportune.

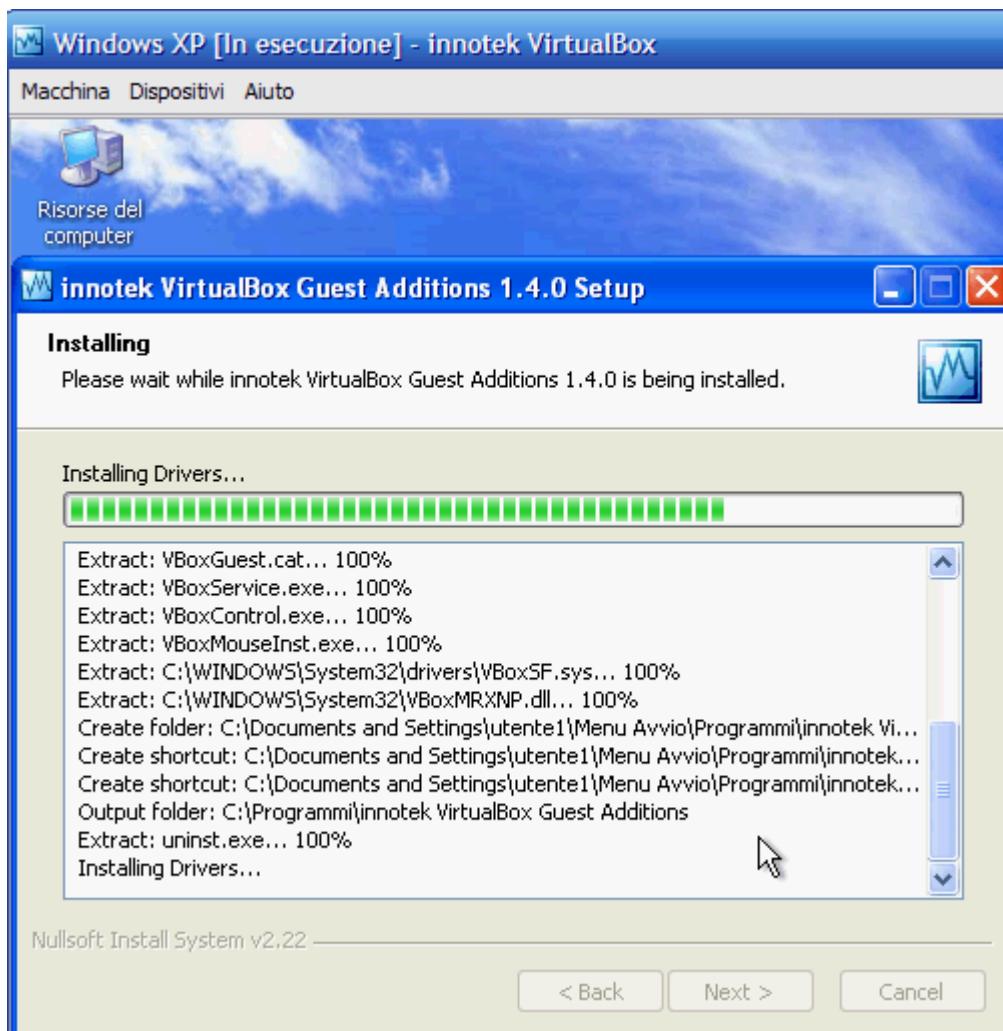


fig. 4

Terminata l'installazione e riavviato il sistema se richiesto, vedrete un'icona nuova nella System Tray a ricordarvi che ora le Guest Additions sono attive (fig. 5).



fig. 5

Vediamole adesso in azione. Cliccate sul desktop con il tasto destro del mouse e poi con il sinistro su *Proprietà*. Nella finestra che si apre cliccate sul tab *Impostazioni* (fig. 6). Come potete notare, la risoluzione massima è passata da 1024x768 a ben 6400x1200 con colore a 32 bit. Scegliete quella a voi più consona e cliccate su *Applica* per verificare che tutto funzioni correttamente.

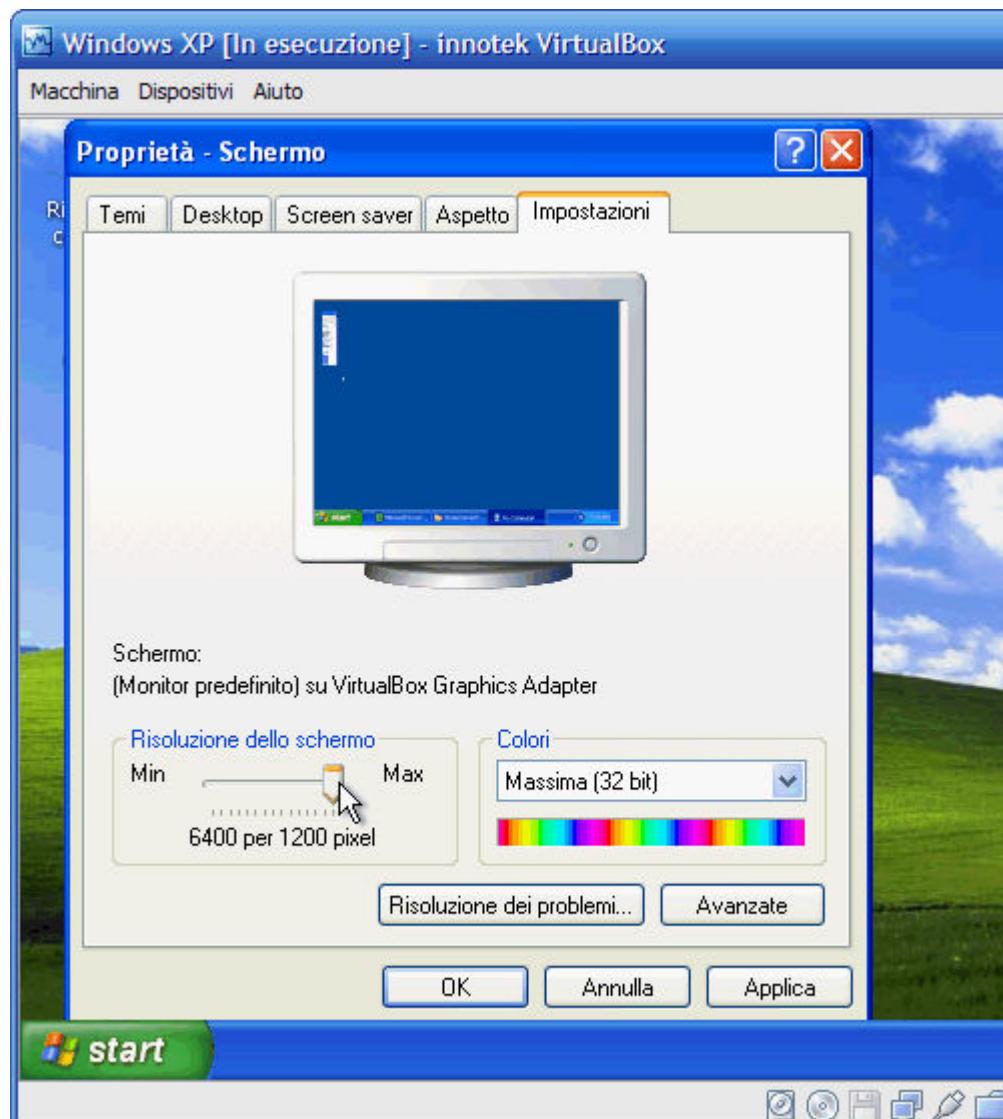


fig. 6

Le Guest Additions in un sistema Windows sono effettivamente molto elastiche e performanti. Per esempio: se cliccate sul menù *Macchina* --> *Ridimensiona automaticamente la finestra Guest* (fig. 7),

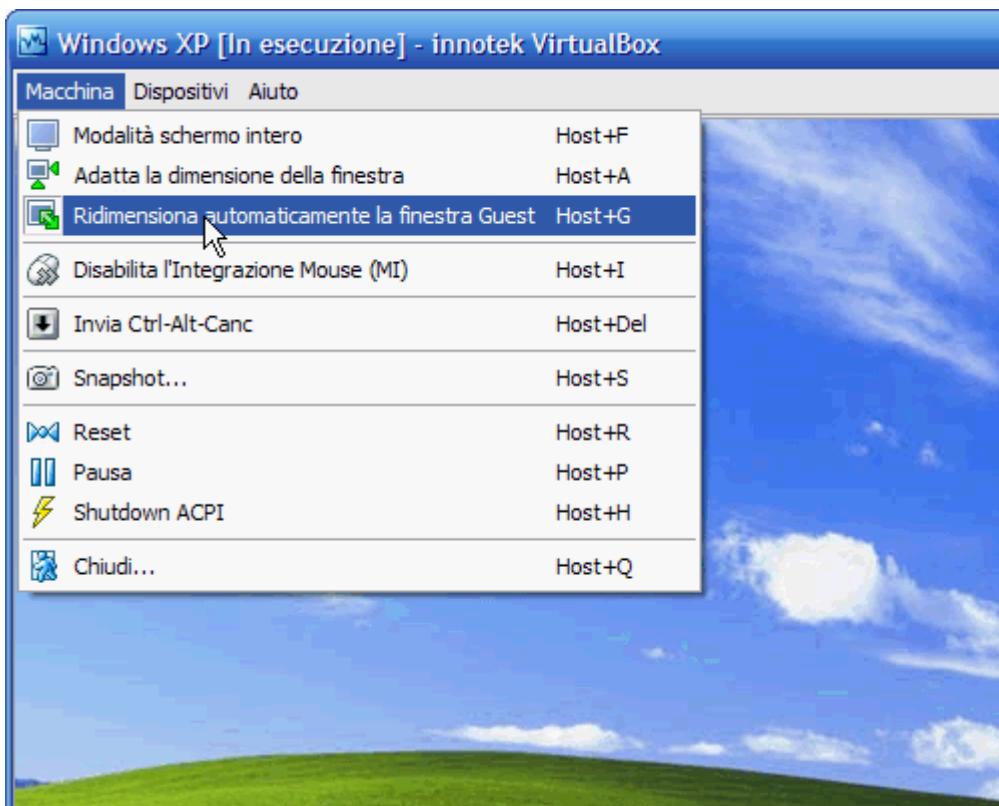


fig. 7

da quel momento in poi, cliccando sul pulsante di ridimensionamento della finestra del guest (**fig. 8**), vedrete la risoluzione all'interno di essa adeguarsi automaticamente. Se cliccate con il tasto destro sul desktop e visualizzate nuovamente *Proprietà*, potrete verificare che la risoluzione impostata è effettivamente cambiata e continua a cambiare.

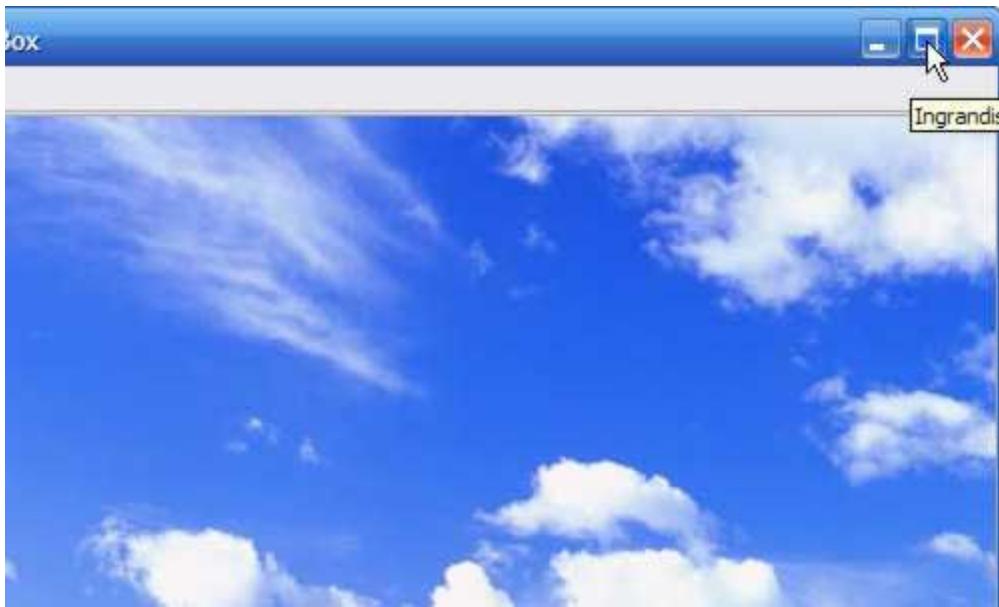


fig. 8

Anche gli appunti condivisi adesso funzionano, comunque fate un giro delle varie funzioni a disposizione e verificate che tutto funzioni a dovere.

@:>

Pubblicato alle 15.28 Argomenti: guida, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione, Windows

2 commenti:

Michele Benvegnù ha detto...

Bene, sei passato ad illustrare i guest Windows; per ora XP, forse passerai ad illustrare l'installazione di Vista come guest.

Di recente avevo postato un commento rimarcando come con Windows Vista non sia riconosciuta la scheda di rete, ed era pertanto necessario scaricare i driver software dal produttore (AMD) ed installarli manualmente.

Correggo quanto affermato: nel manuale di VirtualBox tale carenza di Vista è nota e ben documentata, al punto che i driver della scheda di rete sono presenti nel CD virtuale (file ISO) delle additions: basta nontare il file per trovarsi i driver senza doverli scaricare.

27 giugno 2007 23.07

Giornale di Sistema ha detto...

Ciao Michele.

Grazie come sempre per le segnalazioni; in effetti avevo letto la cosa e poi però me ne ero completamente dimenticato (misteri della memoria umana).

@:)>

28 giugno 2007 10.38

lunedì 23 luglio 2007

VirtualBox per OS X: versione beta 2 disponibile

Un breve post per segnalarvi novità dal mondo VirtualBox.

Il lavoro di sviluppo e miglioramento dei ragazzi della innotek prosegue, e proprio in questi giorni hanno rilasciato la versione beta 2 di VirtualBox per OS X in versione per processori Intel; da segnalare la presenza del supporto USB.

Nel frattempo, si allunga anche la lista dei sistemi operativi guest supportati.

@:\>

Pubblicato alle 17.20 Argomenti: news, software, Virtualbox, virtualizzazione

lunedì 6 agosto 2007

VirtualBox: Guest Additions in Windows Vista

Passata la bufera nella mia vita, anche il blog pian piano riprende a vivere con più intensità. Riprendiamo perciò laddove ci eravamo fermati, dalle VirtualBox Guest Additions, analizzandone l'installazione su un sistema guest Windows Vista; vi ricordo infatti che la loro installazione avviene sempre all'interno della macchina guest.

Avviate la macchina virtuale in cui avete installato Vista e cliccate sul menù *Dispositivi*, quindi su *Installa Guest Additions* (fig. 1).

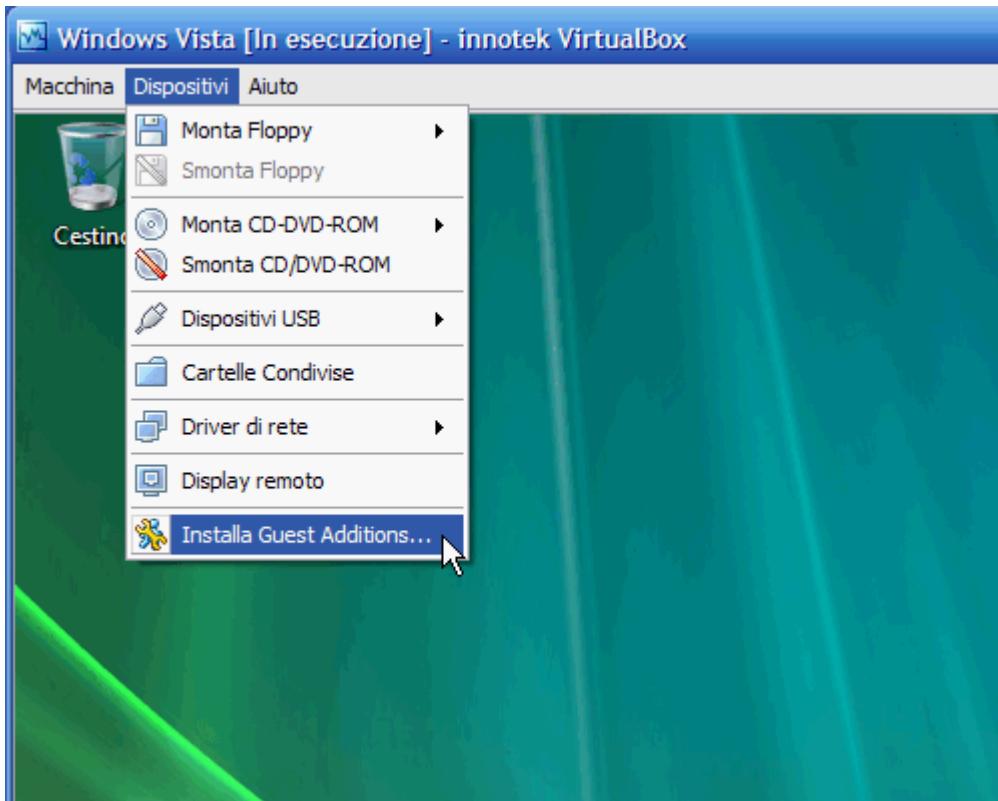


fig. 1

Le Guest Additions verranno caricate all'interno del CD-ROM della vostra macchina virtuale ed il resto lo farà l'autoplay (fig. 2). Cliccate su *Esecuzione di VboxGuestAdditions.exe* per avviare il programma di installazione.

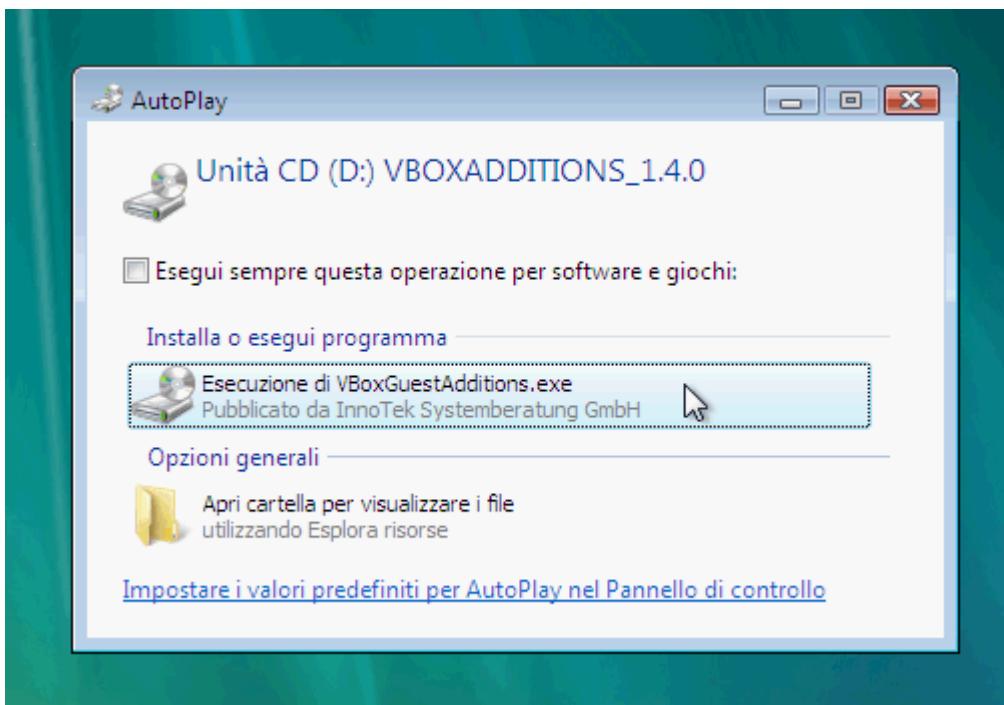


fig. 2

Vedrete partire il Wizard che vi guiderà nell'installazione delle Guest Additions all'interno della macchina virtuale, cliccate su *Next* per proseguire (fig. 3).



fig. 3

Dopo averla letta, accettate i termini della licenza (fig. 4) e cliccate su *Next* per proseguire.

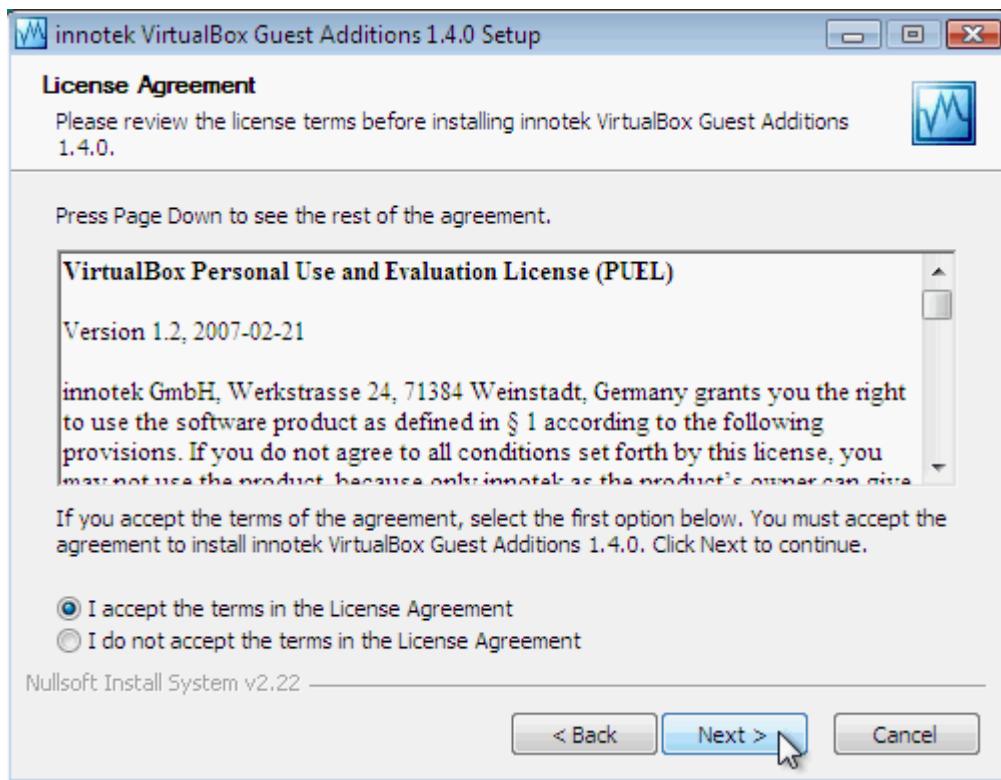


fig. 4

Vi verrà chiesta la cartella dove installare le Guest Additions, crearene una apposita se quella proposta di default non vi piace. Quando avete finito, cliccate su *Install* (fig. 5) per iniziare l'installazione vera e propria.

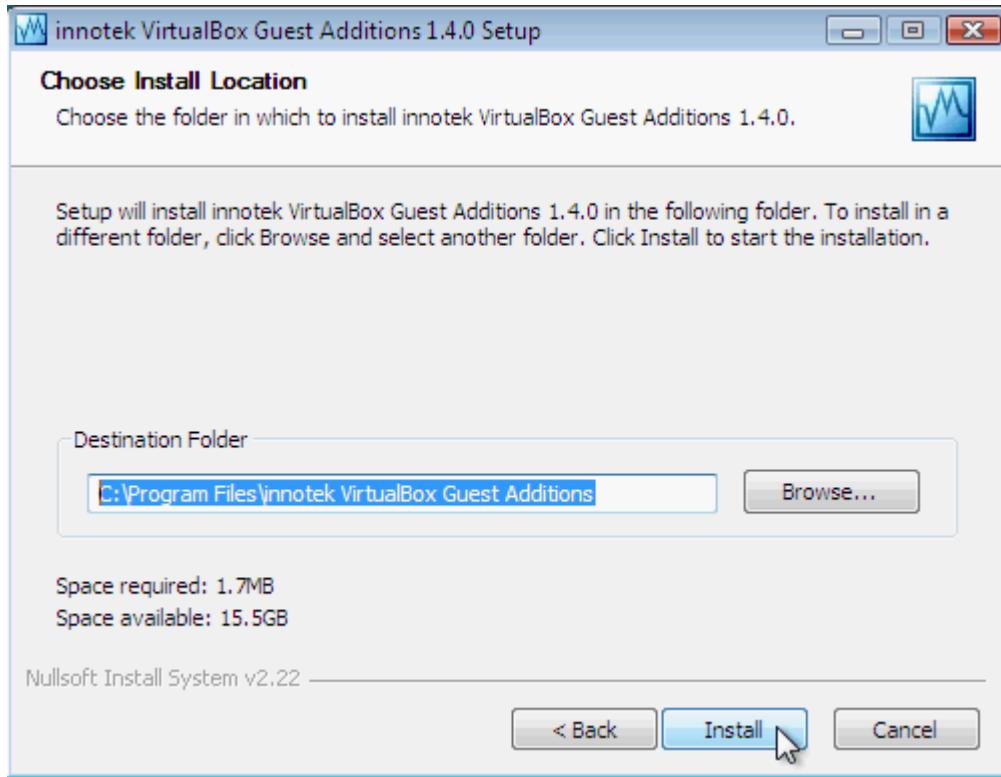


fig. 5

L'installazione andrà avanti per un po', fino a quando non si fermerà per l'intervento dei meccanismi di protezione di Vista (fig. 6). Si tratta dell'installazione del driver della scheda video, cliccate su *Install* se volete proseguire.



fig. 6

Stessa cosa succederà nel momento in cui le Guest Additions cercheranno di installare altri driver di sistema (fig. 7), cliccate ancora una volta su *Installa* se volete proseguire.

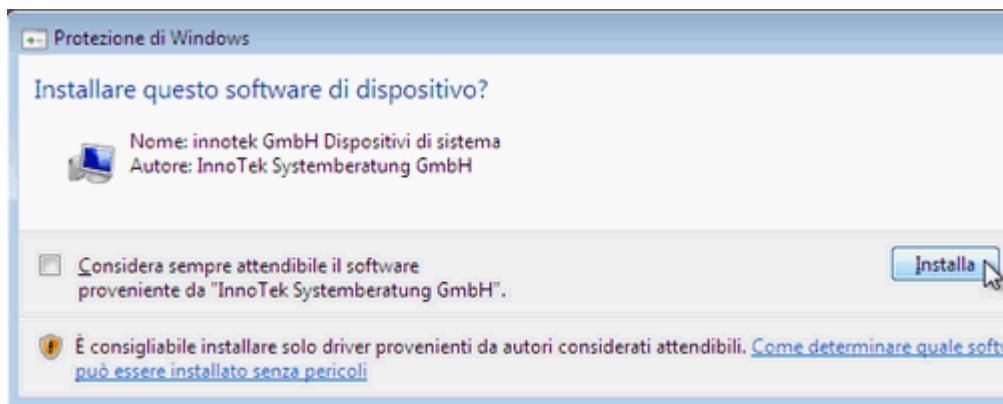


fig. 7

Terminata l'installazione, vi verrà chiesto di riavviare il sistema (fig. 8), selezionate *Reboot now* se già non lo fosse, e cliccate su *Finish* per riavviare.



Fig. 8

Terminato il riavvio, se tutto è andato bene noterete immediatamente le prime differenze:

- il mouse non "casca" più nella finestra restandone intrappolato,
- il desktop di Vista adegua automaticamente la sua risoluzione a quella della finestra di VirtualBox in cui è contenuto.

Provate infatti ad allargare a tutto schermo la finestra di VirtualBox in cui gira Vista: nel giro di pochi istanti vedrete il desktop di Vista allargarsi a sua volta fino a riempire per intero la finestra. Come esempio, in **fig. 9** vi riporto la risoluzione video che leggo sul mio sistema guest; tenete presente che l'host è un portatile con risoluzione 1440x900 pixel.

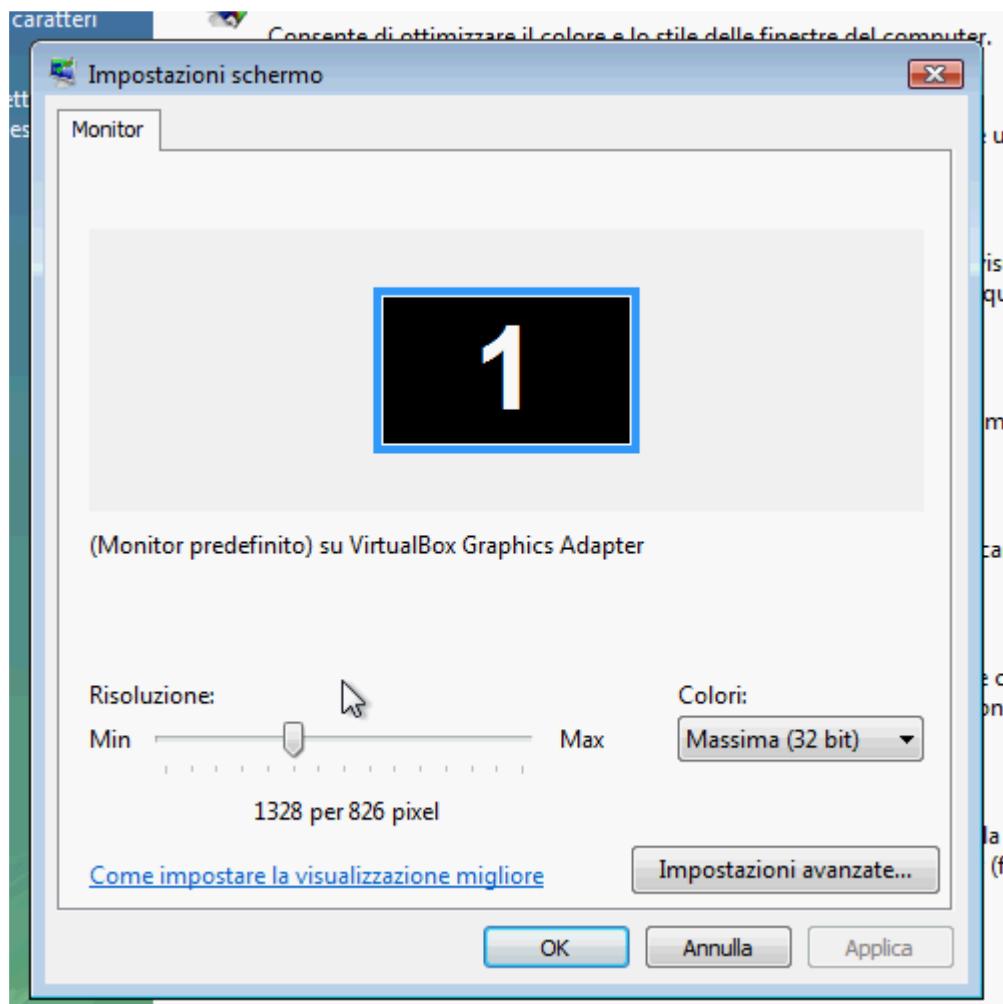


fig. 9

Bene, questo è tutto per quanto riguarda l'installazione delle VirtualBox Guest Additions in Windows Vista.

Nei prossimi post, chiuderò il discorso dell'installazione delle VirtualBox Guest Additions, illustrandovi a grandi linee, come farla nei sistemi Windows e Linux non considerati esplicitamente finora.

@:\>

Pubblicato alle 9.20 Argomenti: guida, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione, Windows

2 commenti:

paolo ha detto...

Complimenti, descrizione interessante e molto ben fatta!.

Ciao,
Paolo

6 agosto 2007 15.16
Giornale di Sistema ha detto...

Ciao Paolo, grazie! :-)

@:\>

7 agosto 2007 10.08

giovedì 9 agosto 2007

VirtualBox: Guest Additions in Windows NT-2000-2003 Server

Post breve questo, più che altro una indicazione.

L'installazione delle Guest Additions in:

- Windows NT,
- Windows 2000,
- Windows 2003 Server,

non si discosta da quanto già visto nell'installazione per Windows XP e Vista che ho usato come esempio; fate pertanto tranquillamente riferimento ai suddetti post.

@:\>

Pubblicato alle 18.18 Argomenti: guida, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione, Windows

mercoledì 15 agosto 2007

VirtualBox: Guest Additions in Ubuntu Linux

Questo post sarebbe dovuto essere l'ultimo della serie sull'installazione delle VirtualBox Guest Additions, ma potevamo davvero non dare un'occhiata all'installazione delle Guest Additions in almeno un'altra distribuzione Linux, la Ubuntu?

In genere questa è la prima distribuzione cui approdano tanti utenti Windows che vogliono provare Linux, perciò saranno in molti alle prime armi che hanno bisogno di qualche dritta su come procedere: vediamolo in concreto.

L'installazione di riferimento è una Ubuntu 7.04 in configurazione di default, ossia, lasciando inalterate tutte le impostazioni proposte dall'installazione stessa; dovreste comunque riuscire ad utilizzare agevolmente questa guida anche con versione meno recenti, ad esempio, la precedente 6.10, e per estensione, anche con distribuzioni simili.

Avviate la macchina virtuale su cui avete installato Ubuntu (per i particolari della creazione di una macchina virtuale su cui installare Ubuntu, fate riferimento al post "Virtualizzazione: creiamo un PC virtuale con VirtualBox 1.4 (parte 1)" e successivi).

Terminato il caricamento, vi troverete davanti al desktop vuoto, tranne forse per la ISO caricata nel CD-ROM virtuale (**fig. 1**).

Se non vi ricordate quali sono in generale i passi necessari per l'installazione delle Guest Additions in Linux, è il momento buono per rileggere il post "VirtualBox: Guest Additions in Linux, procedura generale": quello che occorre, è l'attivazione del supporto ai moduli di kernel esterni, poi il mount, la compilazione e l'installazione delle Guest Additions. Per fare tutto ciò, occorre anche la presenza di GCC, il compilatore per il linguaggio C disponibile nelle varie distribuzioni Linux.

Verifichiamo se il vostro Ubuntu tutto è pronto per l'installazione delle Guest Additions.

Controlliamo per prima cosa se è già installato il compilatore C. Cliccate su *System --> Amministrazione --> Gestore pacchetti Synaptic* (**fig. 1**).



fig. 1

Vi verrà chiesta la password da amministratore, dopo averla inserita correttamente, si aprirà la finestra del Synaptic Package Manager (fig. 2): cliccate su *Section*, (dove indicato dal mouse), poi su *Development* (in alto sopra il mouse), e controllate che la voce "gcc", a destra, sia con il box di selezione annerito, vuol dire che è installato. Se così non fosse, procedete alla sua installazione selezionandolo e cliccando sulla voce "Mark for Installation"che comparirà a video.

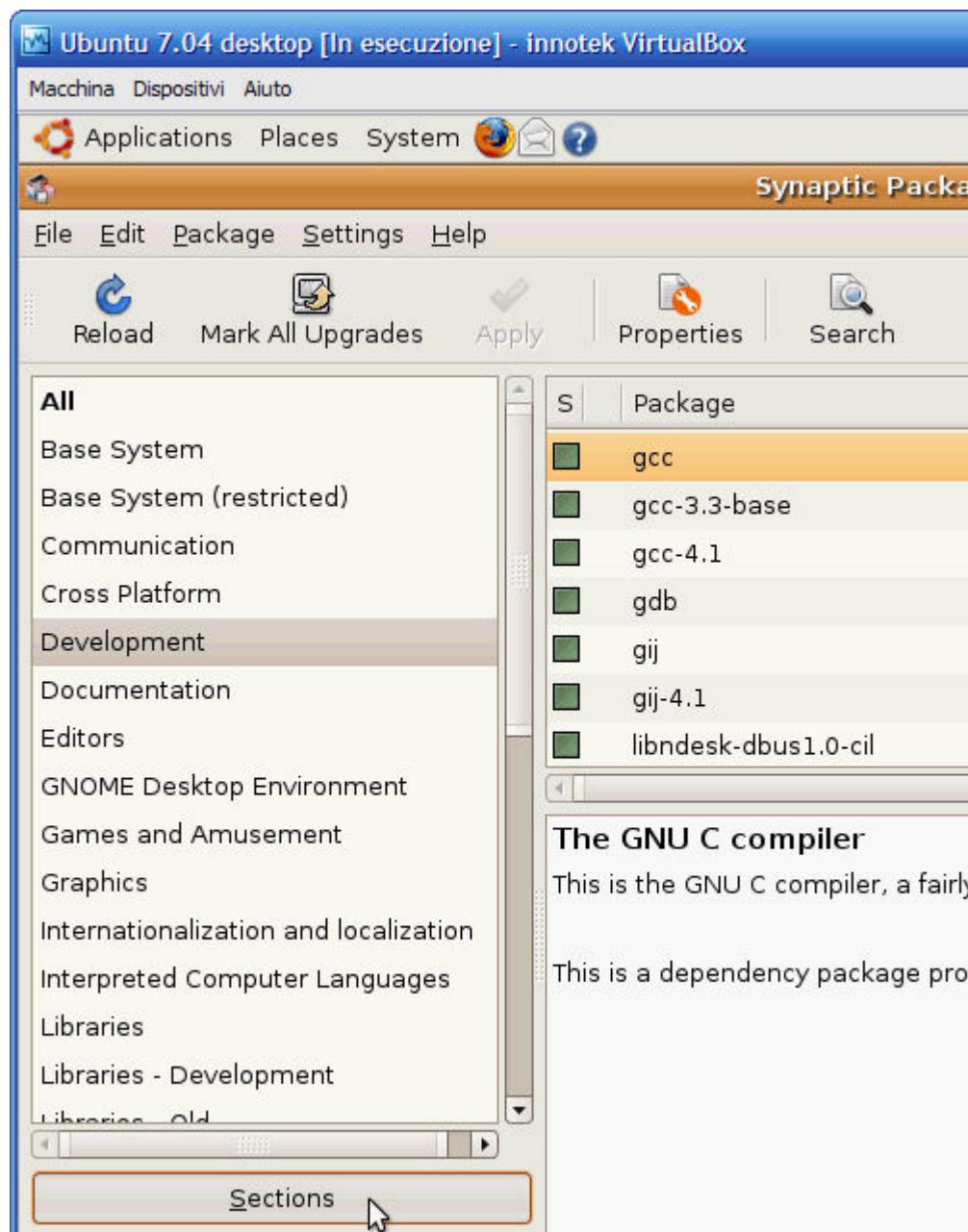


fig. 2

Andata a buon fine la verifica della presenza del compilatore C, controlliamo adesso la presenza del supporto ai moduli di kernel esterni. Dal post "VirtualBox: Guest Additions in Linux, procedura generale" potete verificare il nome dei moduli occorrenti: "linux-headers", e se esiste "linux-kbuild".

Cliccate sul pulsante *Search* (fig. 3), in alto a destra dove è posizionato il mouse, e nel box di ricerca che si apre, alla voce *Search* scrivete "linux", alla voce *Look in* selezionate "Name", infine cliccate sul pulsante *Search* immediatamente sotto.

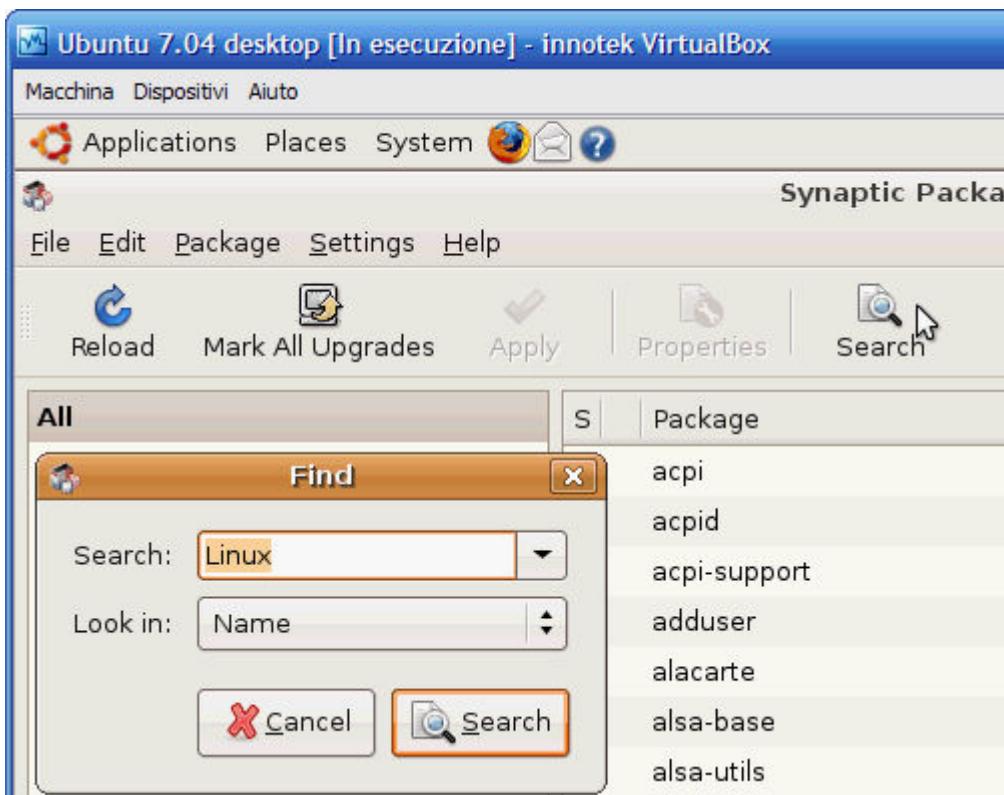


fig. 3

Vi verrà presentata la lista di tutti i package che contengono la parola "linux" **fig. 4**), verificate che le voci "linux-header e qualcosa" siano già installate (box annerito), altrimenti, installatele come già visto.

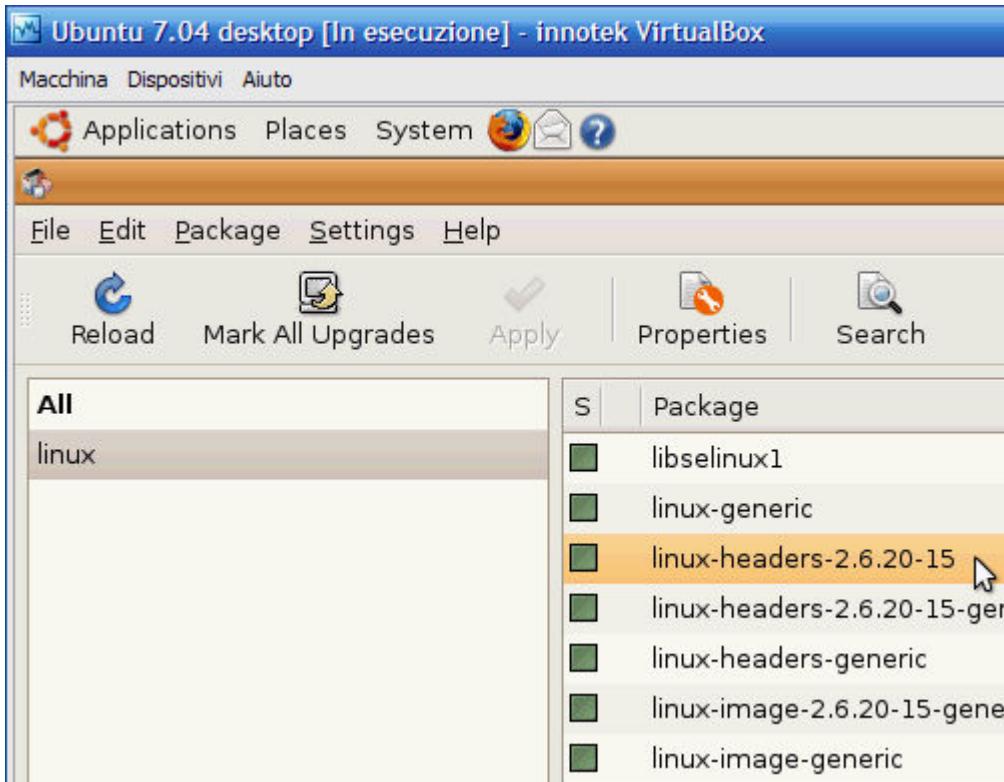


fig. 4

Verificate la presenza o meno della voce "linux-kbuild" (**fig. 5**); in questo caso non c'è, se invece c'è, se invece c'è, allora come prima, verificate che sia installata, oppure installatela.

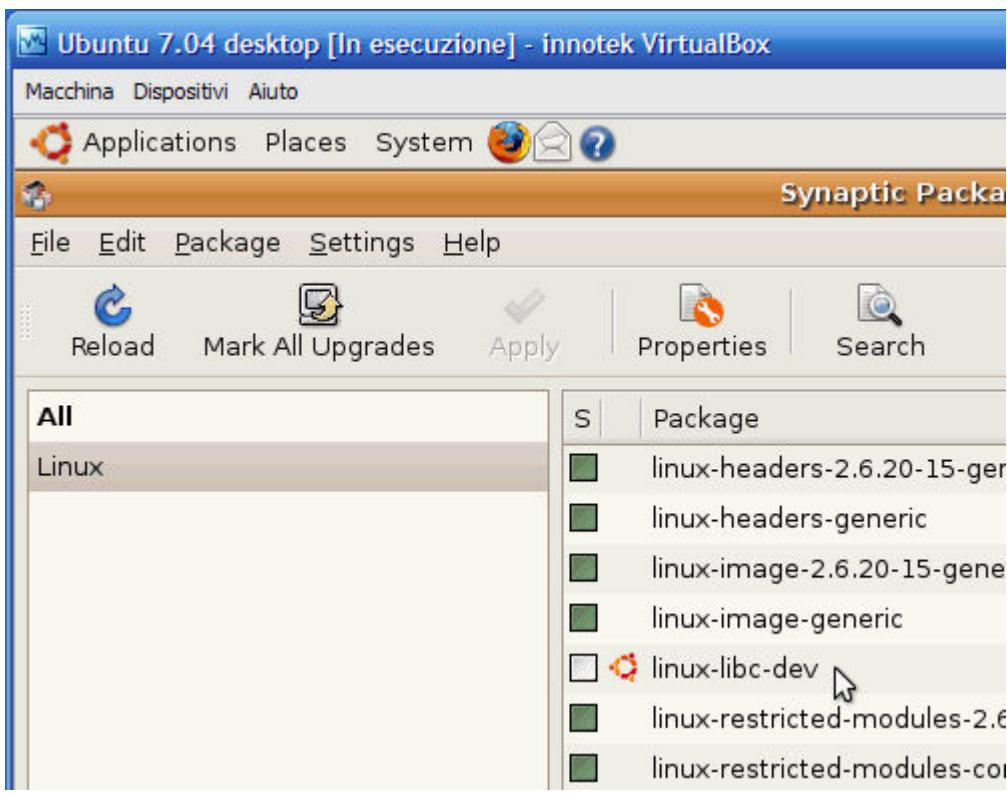


fig. 5

Bene, abbiamo completato tutte le verifiche, dalla presenza del compilatore C a quella dei moduli necessari al supporto dei moduli di kernel esterni, siamo pronti con l'installazione delle Guest Additions.

Se sul desktop è ancora presente il CD o la ISO da cui abbiamo installato Ubuntu, procedete a tirarlo fuori dal CD-ROM virtuale (**fig. 6**).

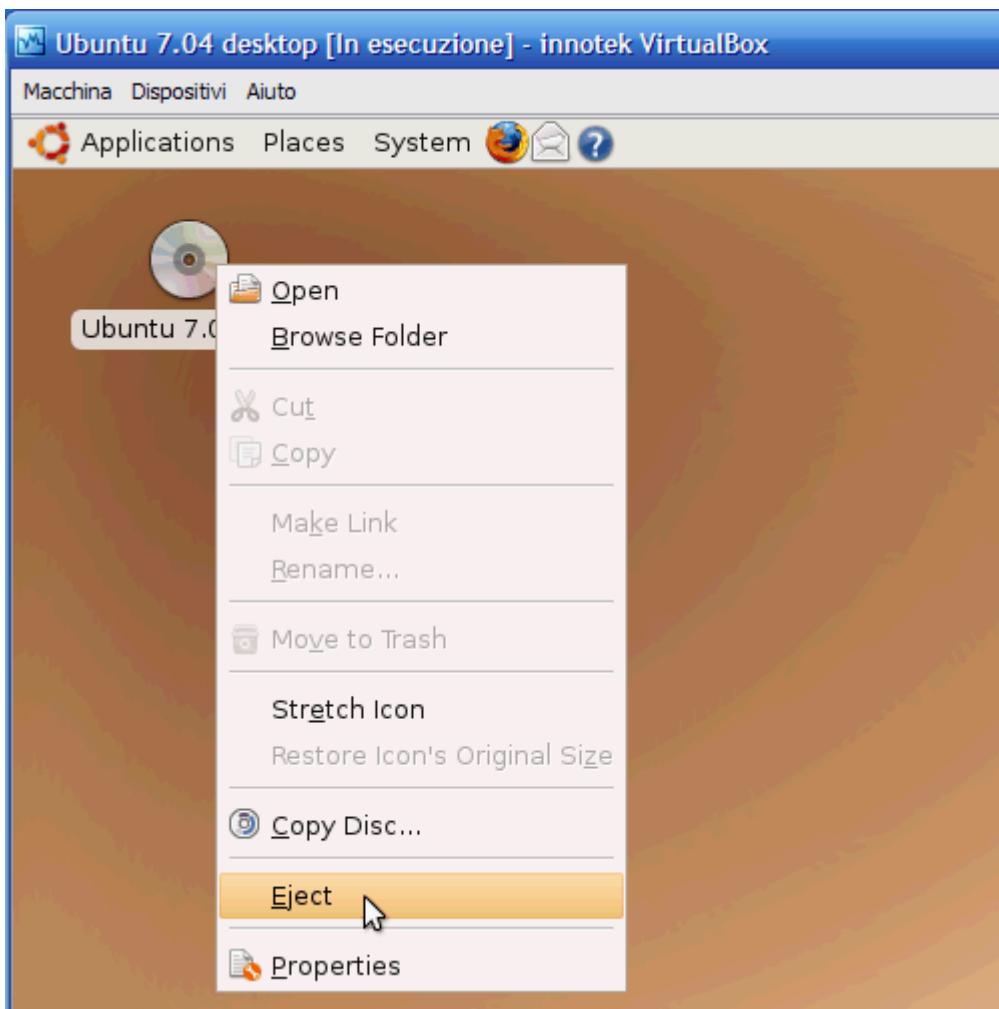


fig. 6

Scomparsa l'icona dal desktop, cliccate sul menù *Dispositivi* --> *Installa Guest Additions* della finestra di VirtualBox (fig. 7).

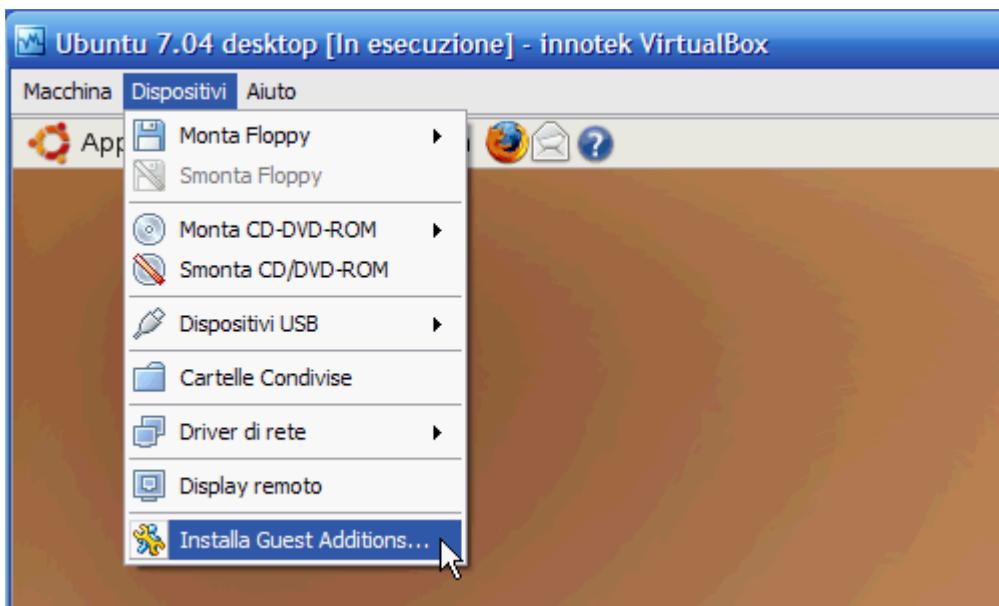


fig. 7

Comparirà sul desktop la ISO delle VirtualBox Guest Additions e automaticamente verranno aperte nel file manager (fig. 8).

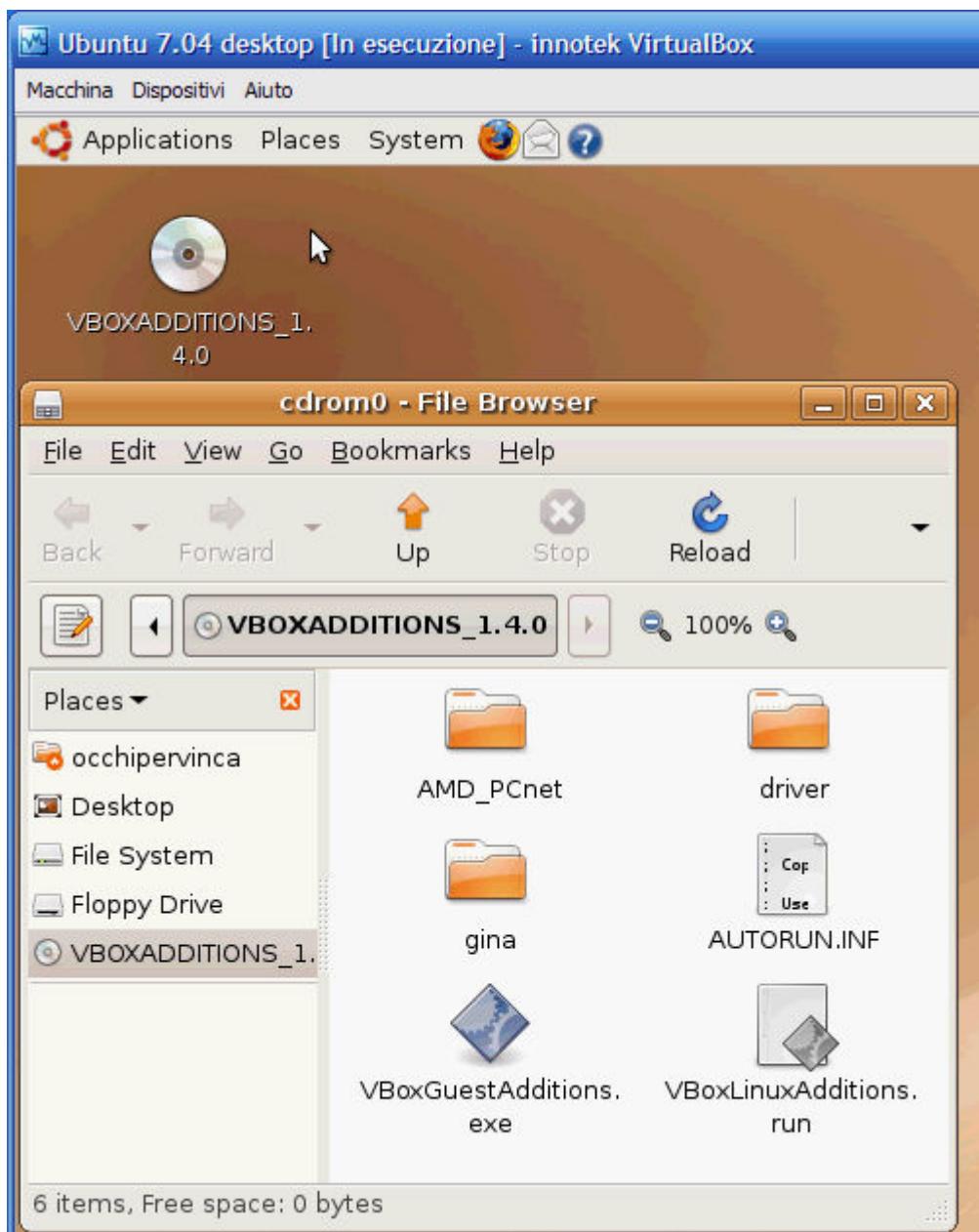


fig. 8

Il file che permette di avviare la compilazione delle Guest Additions è "VBoxLinuxAddition.run", dovete avviarlo dalla shell (l'interfaccia testuale da cui dare comandi al sistema) con i permessi da amministratore.

Cliccate su *Applications* --> *Accessori* --> *Terminale* (fig. 9).



fig. 9

Si aprirà la finestra della shell, tenete d'occhio **fig. 10** per avere un riscontro mentre date questi comandi:

- cd / [invio] # Vi porta nella directory (se non siete pratici della terminologia Linux, directory sta per cartella)
- ls [invio] # Vi da la lista dei file e delle directory presenti, a noi interessa la directory cdrom
- cd cdrom [invio] ## Vi porta nella directory cdrom
- ls [invio] # Vi da la lista dei file e delle directory presenti nella cartella cdrom in cui vi trovate adesso
- sudo sh ./VBoxLinuxAdditions.run [invio] # Avvia il file VBoxLinuxAdditions.run con i permessi da amministratore

Inserite la password da amministratore per proseguire.

Ubuntu 7.04 desktop [In esecuzione] - innotek VirtualBox

Macchina Dispositivi Aiuto

Applications Places System

occhipervinca@occhipervinca-laptop: /cdrom

File Edit View Terminal Tabs Help

```
occhipervinca@occhipervinca-laptop:~$ cd /
occhipervinca@occhipervinca-laptop:/$ ls
bin etc lib opt srv var
boot home lost+found proc sys vmlinuz
cdrom initrd media root tmp
dev initrd.img mnt sbin usr
occhipervinca@occhipervinca-laptop:/$ cd cdrom
occhipervinca@occhipervinca-laptop:/cdrom$ ls
AMD_PConet driver VBoxGuestAdditions.exe
AUTORUN.INF gina VBoxLinuxAdditions.run
occhipervinca@occhipervinca-laptop:/cdrom$ sudo sh ./VBox
LinuxAdditions.run
Password: 
```

cdrom

22 items, Free space: 15,9 GB

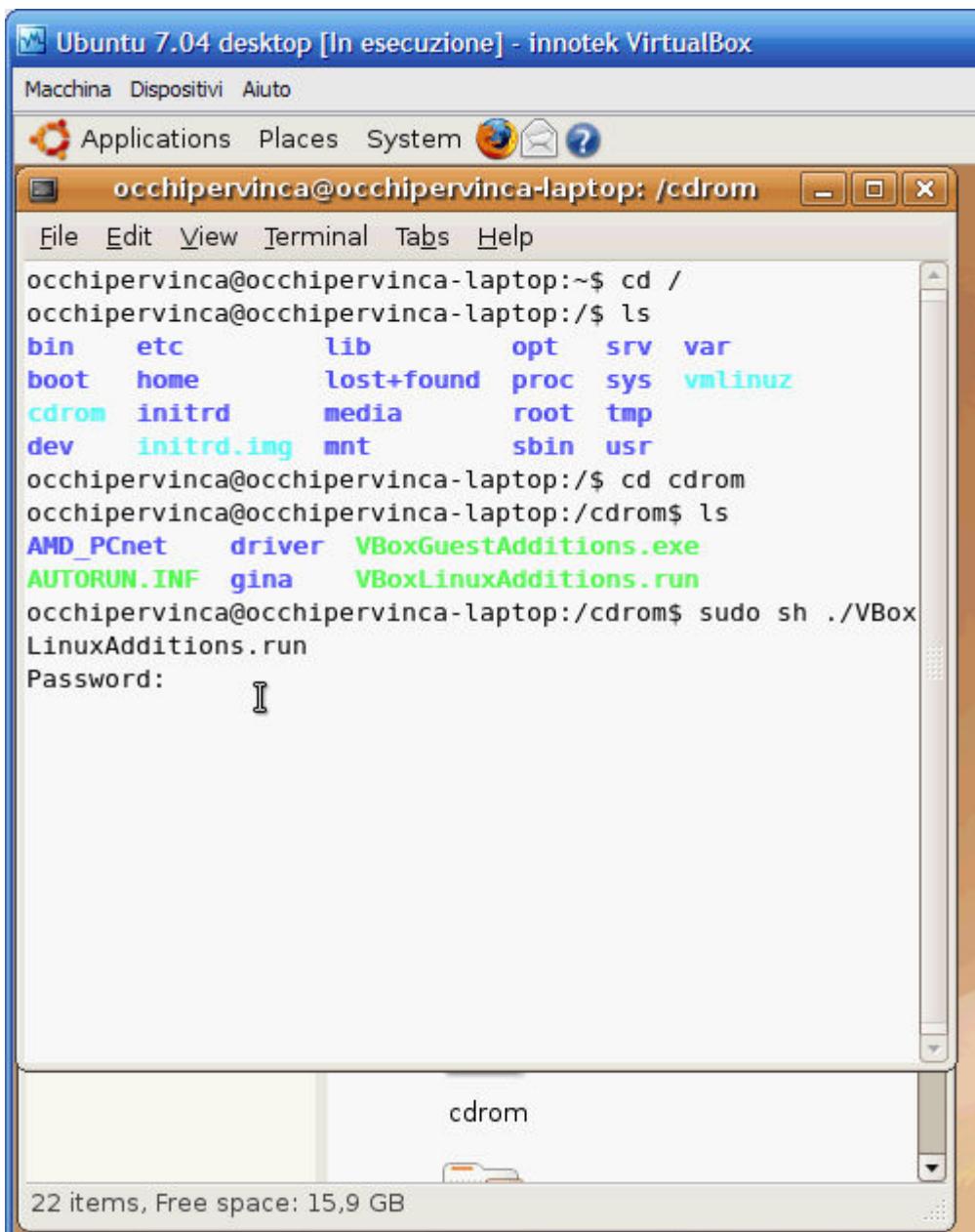
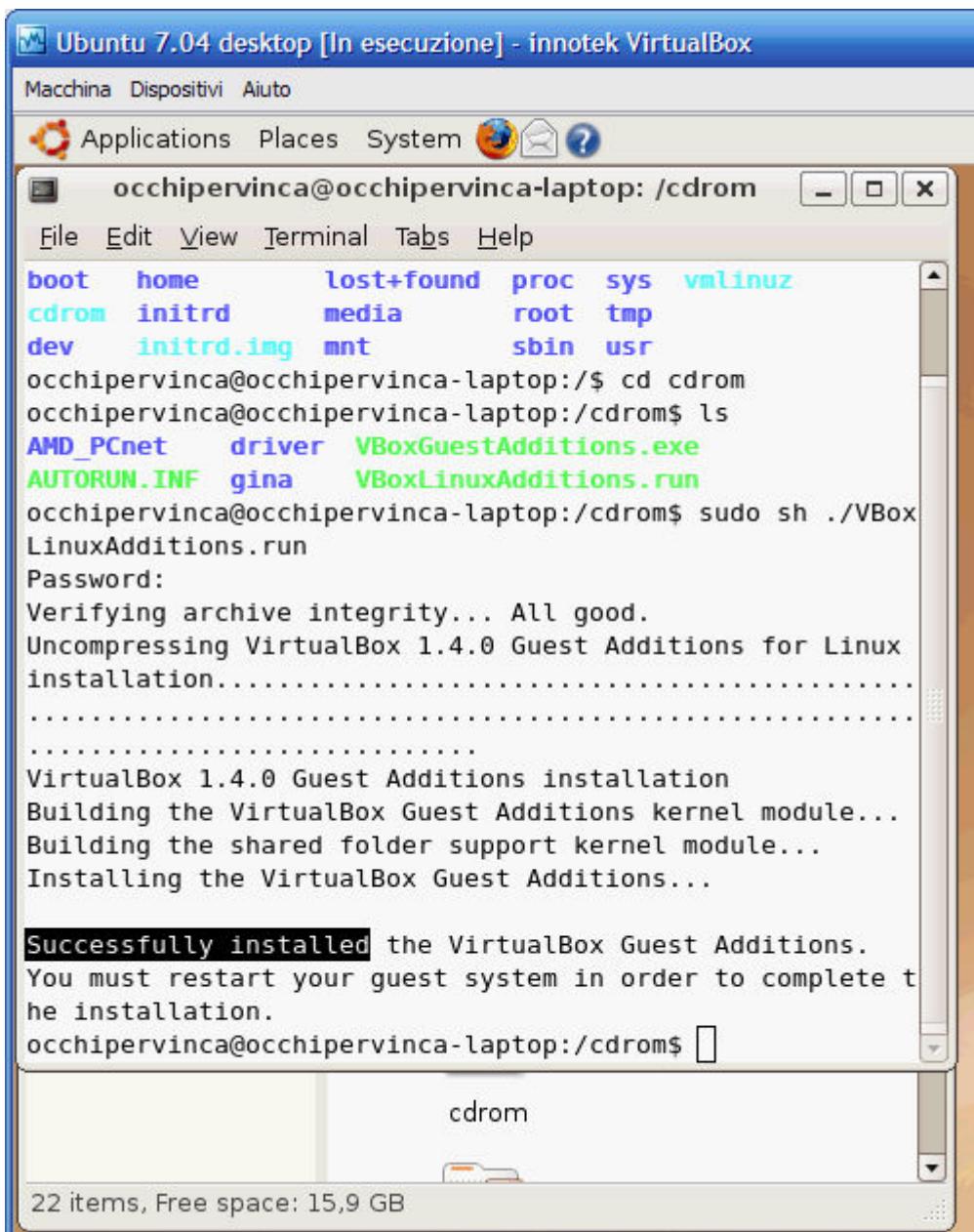


fig. 10

Terminata la compilazione e l'installazione le VirtualBox Guest Additions (fig. 11), non vi resta che riavviare.



```
Ubuntu 7.04 desktop [In esecuzione] - innotek VirtualBox
Macchina Dispositivi Aiuto
Applications Places System
occhipervinca@occhipervinca-laptop: /cdrom
File Edit View Terminal Tabs Help
boot  home      lost+found  proc  sys  vmlinuz
cdrom  initrd    media      root  tmp
dev    initrd.img  mnt      sbin  usr
occhipervinca@occhipervinca-laptop:/$ cd cdrom
occhipervinca@occhipervinca-laptop:/cdrom$ ls
AMD_PNet    driver  VBoxGuestAdditions.exe
AUTORUN.INF  gina    VBoxLinuxAdditions.run
occhipervinca@occhipervinca-laptop:/cdrom$ sudo sh ./VBox
LinuxAdditions.run
Password:
Verifying archive integrity... All good.
Uncompressing VirtualBox 1.4.0 Guest Additions for Linux
installation.....
.
.
.
VirtualBox 1.4.0 Guest Additions installation
Building the VirtualBox Guest Additions kernel module...
Building the shared folder support kernel module...
Installing the VirtualBox Guest Additions...

Successfully installed the VirtualBox Guest Additions.
You must restart your guest system in order to complete t
he installation.
occhipervinca@occhipervinca-laptop:/cdrom$ cdrom
22 items, Free space: 15,9 GB
```

fig. 11

Riavviata la macchina noterete subito la prima differenza, il mouse non “casca” più nella finestra di VirtualBox in cui gira Ubuntu restandone intrappolato.

Brutte notizie invece dal fronte risoluzioni video.

Con mia sorpresa e a differenza di quanto accaduto con SUSE, ove le nuove risoluzioni erano subito disponibili, in Ubuntu ciò non accade (**fig. 12**).

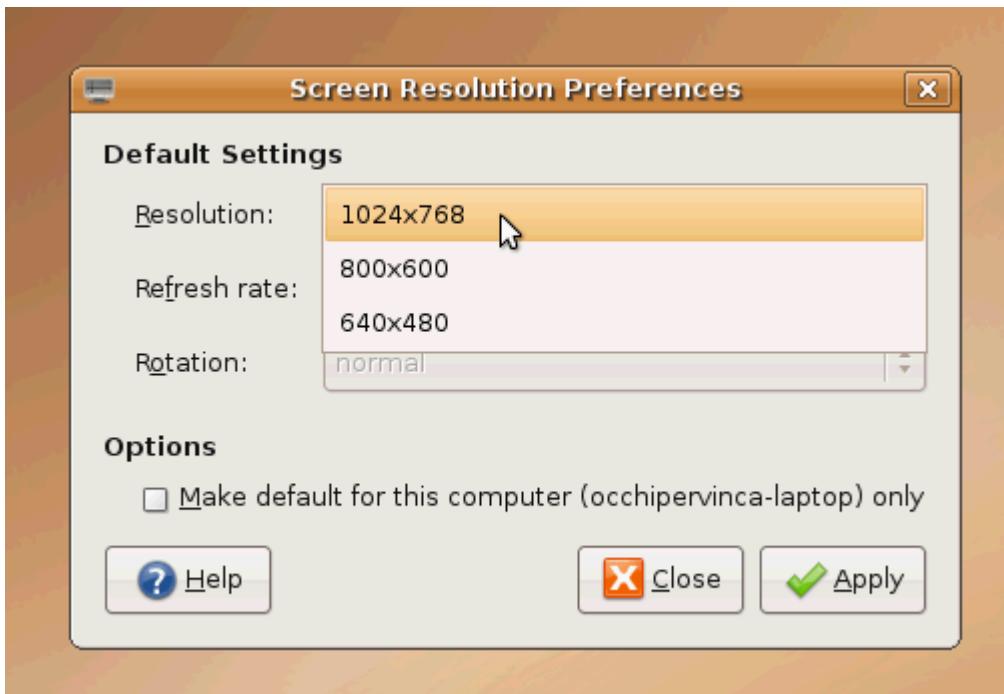


fig. 12

Ma niente paura: nel prossimo post, "VirtualBox: Guest Additions e risoluzioni video in Ubuntu Linux", vedremo come avere risoluzioni video maggiori anche in Ubuntu grazie alle VirtualBox Guest Additions e qualche piccolo smanettamento con i file di configurazione di Linux.

Buon Ferragosto a tutti! :-D

@:)>

Pubblicato alle 8.53 Argomenti: guida, Linux, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione

sabato 18 agosto 2007

VirtualBox: Guest Additions e risoluzioni video in Ubuntu Linux

Nel post precedente, "VirtualBox: Guest Additions in Ubuntu Linux", abbiamo scoperto che Ubuntu non si avvantaggia automaticamente di tutti i benefici che l'installazione delle VirtualBox Guest Additions potrebbe portare: le risoluzioni video a disposizione, nonostante l'installazione delle Guest Additions, non cambiano, restando ferme a 1024x768 pixel (fig. 1).

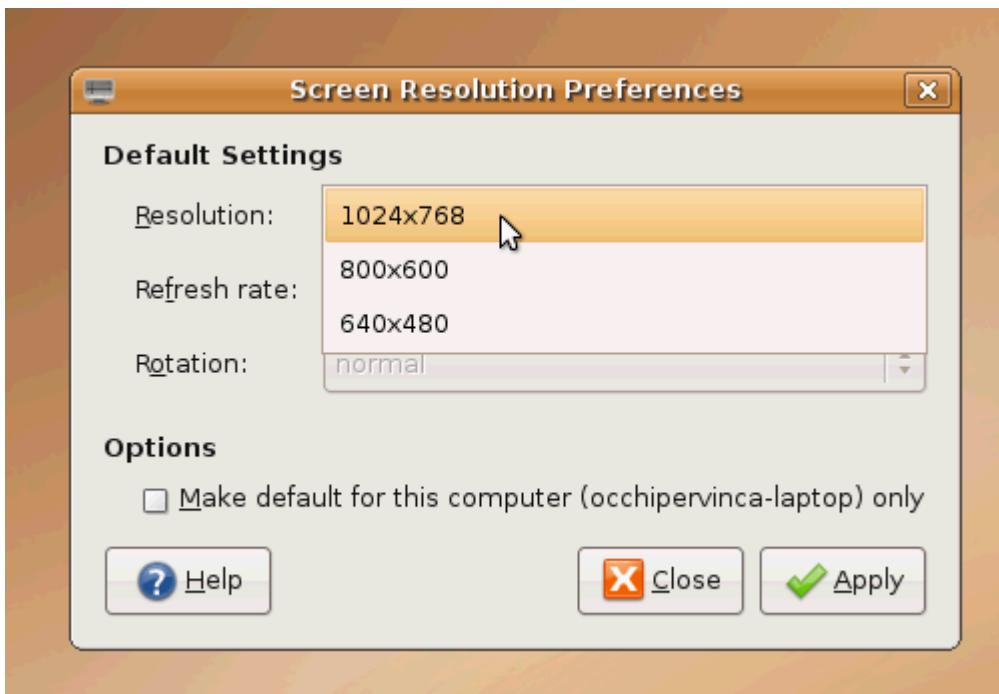


fig. 1

C'è però la possibilità di forzare il sistema grafico X usato da Linux, a vedere e utilizzare le nuove modalità grafiche a disposizione; vediamo come.

In breve, si tratta di andare a modificare un file di configurazione di Linux, *xorg.conf* (X.org è la particolare implementazione del sistema grafico X attualmente usata, dopo che la precedente, Xfree86, ha avuto problemi di licenza), in modo da avvertire esplicitamente il sistema quali sono le modalità video a disposizione. Per fare ciò useremo un editor di testo, *gedit*, avviato con i diritti da amministratore, così da poter salvare le modifiche apportate al file di configurazione.

Vediamo tutto ciò in concreto.

Cliccate su *Applications* --> *Accessori* --> *Terminale* per aprire la shell di Ubuntu (l'interfaccia testuale da cui dare comandi al sistema, fig. 2).

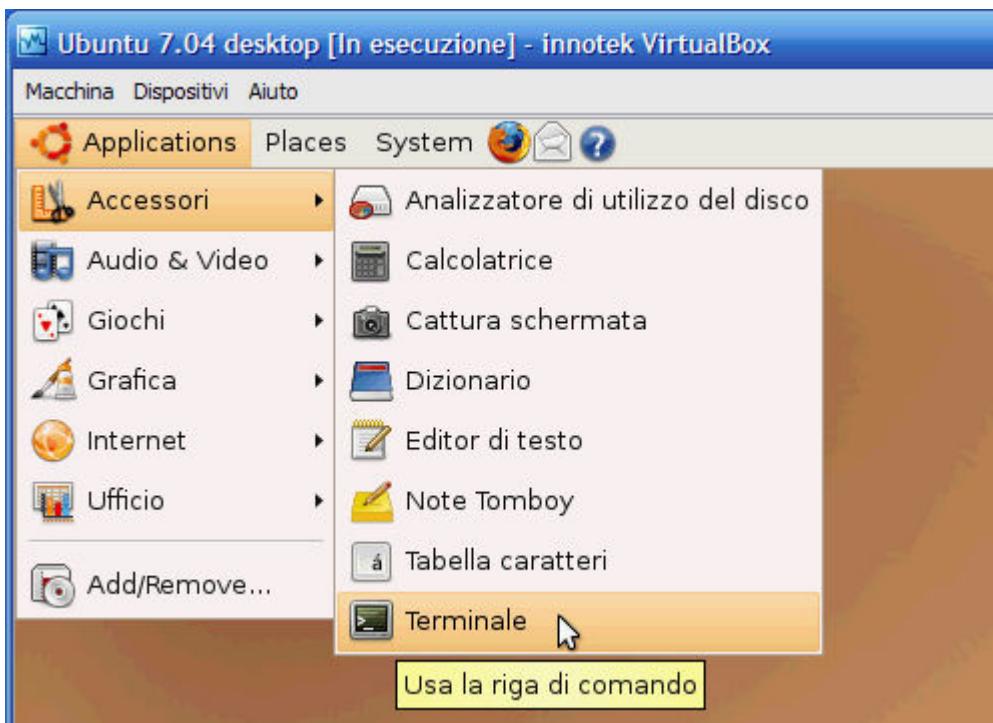


fig. 2

Ora date il comando (fig. 3)

sudo gedit

per lanciare gedit con i privilegi amministrativi; inserite la vostra password da amministratore e date invio per proseguire.

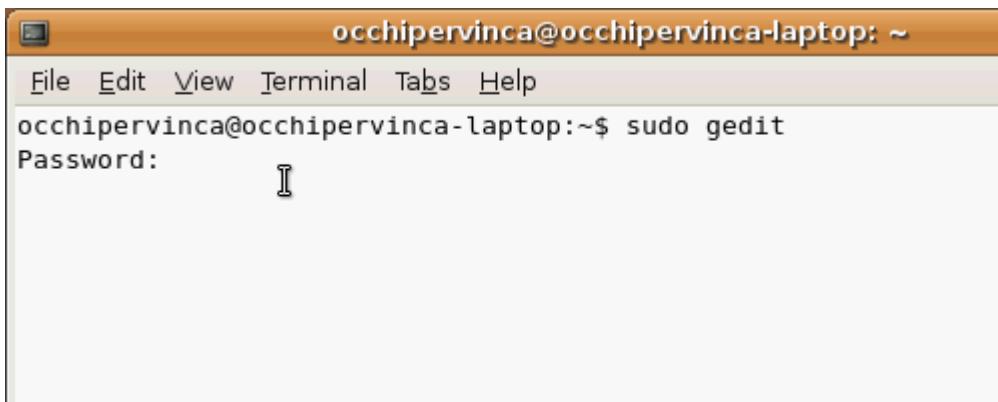


fig. 3

Quando gedit si sarà aperto, cliccate sul pulsante *Open* nella barra degli strumenti (fig. 4).

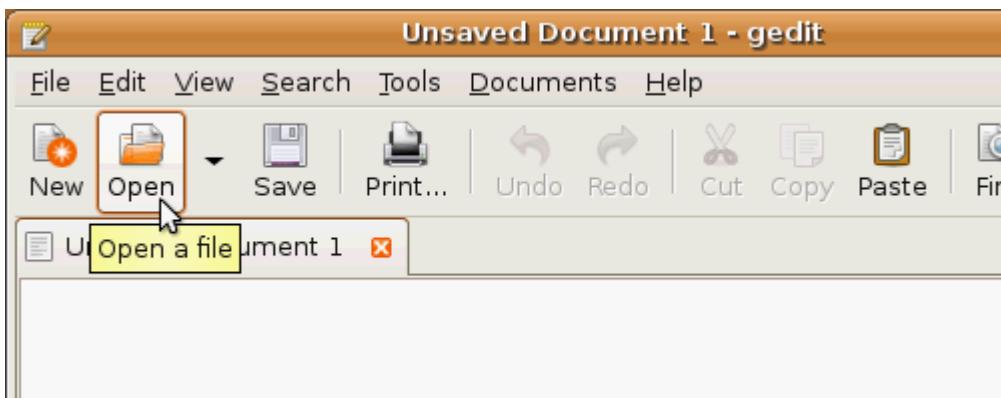


fig. 4

Ora cliccate nel punto indicato dal mouse (fig. 5),

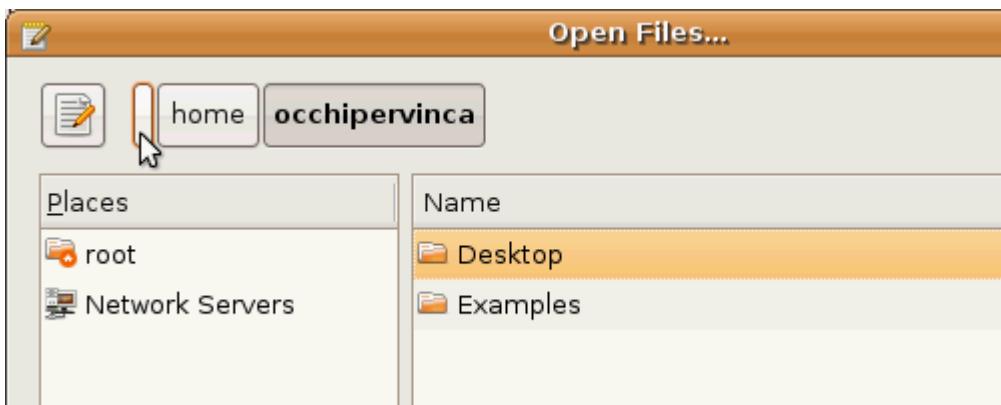


fig. 5

e quando verra visualizzata la directory root (cioè quella principale, indicata anche con il simbolo slash "/"), aprete con un doppio click la directory *etc* (vi ricordo che il termine directory in Linux corrisponde pressapoco al termine cartella in gergo Windows) come in fig. 6,

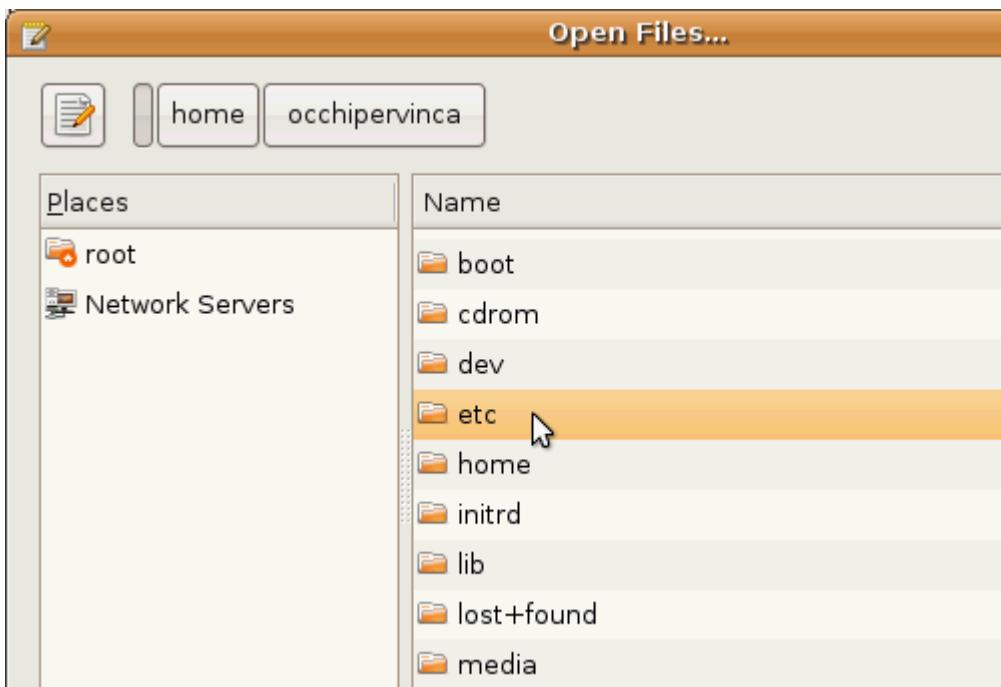
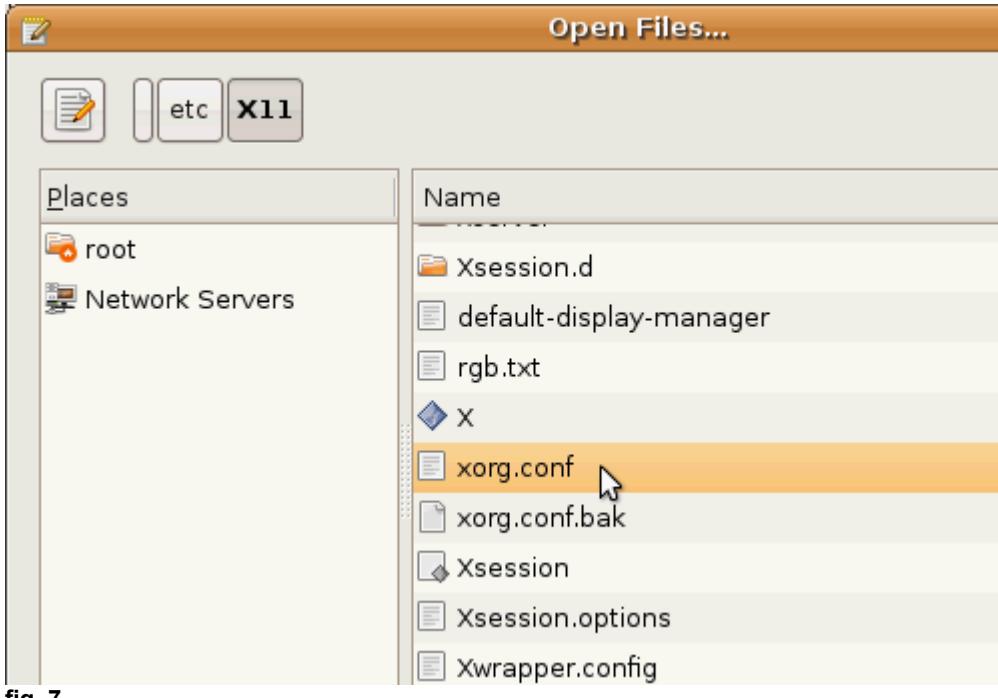


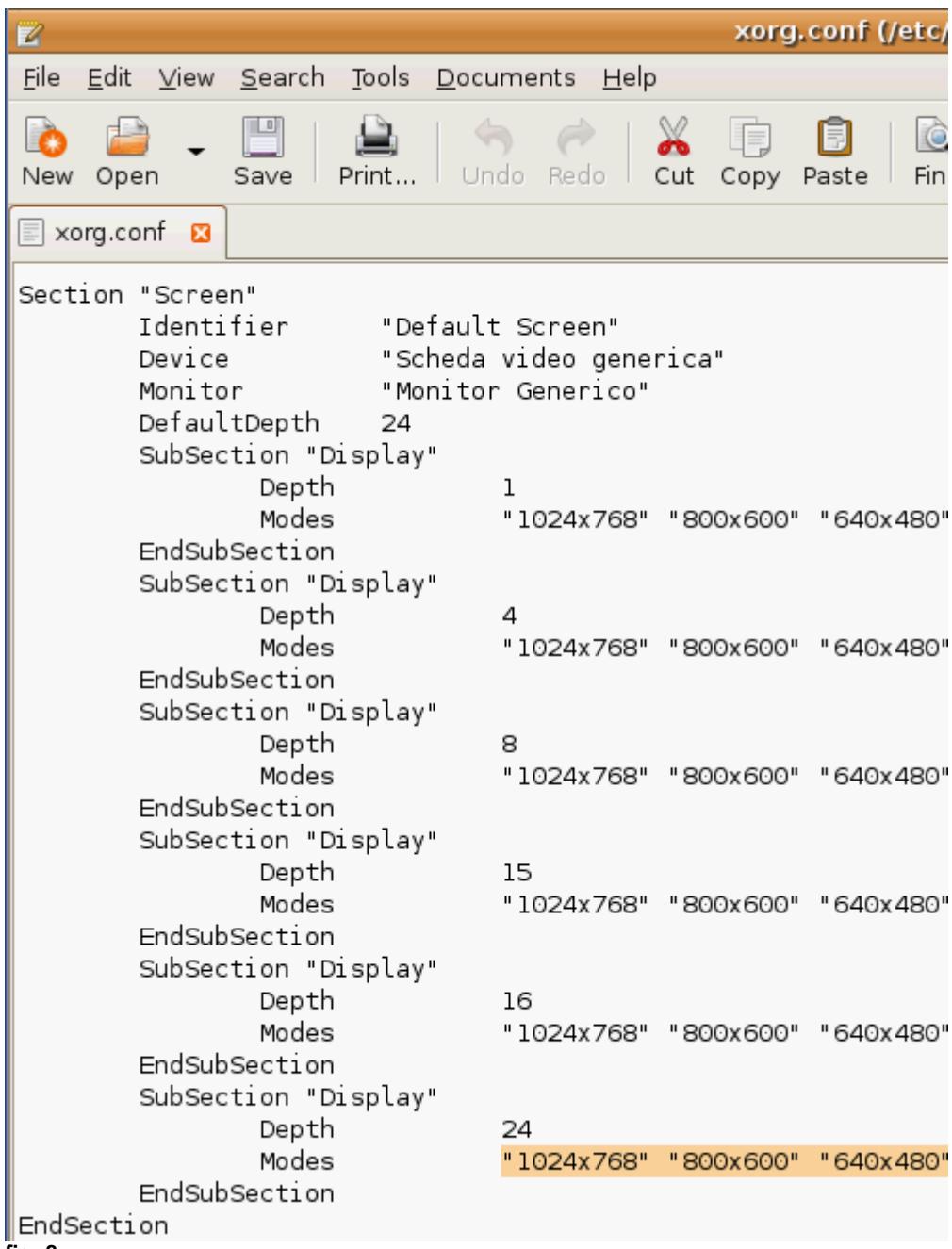
fig. 6

poi cercate e aprite con un doppio click la directory **X11** e al suo interno fate un doppio click sul file **xorg.conf** per aprirlo nell'editor (**fig. 7**).



Fate scorrere il contenuto del file fino a trovare la *Section "screen"* come in **fig. 8**.

La parte evidenziata mostra le attuali risoluzioni disponibili con 24 bit di colore (le abbiamo già viste in **fig. 1**), con 16 bit di colore (ossia i classici 65536 colori) e così via. Useremo la quantità di colori più alta (24 bit, pari a 16.777.216 colori) con una risoluzione tale da permettervi di occupare tutto lo schermo disponibile.



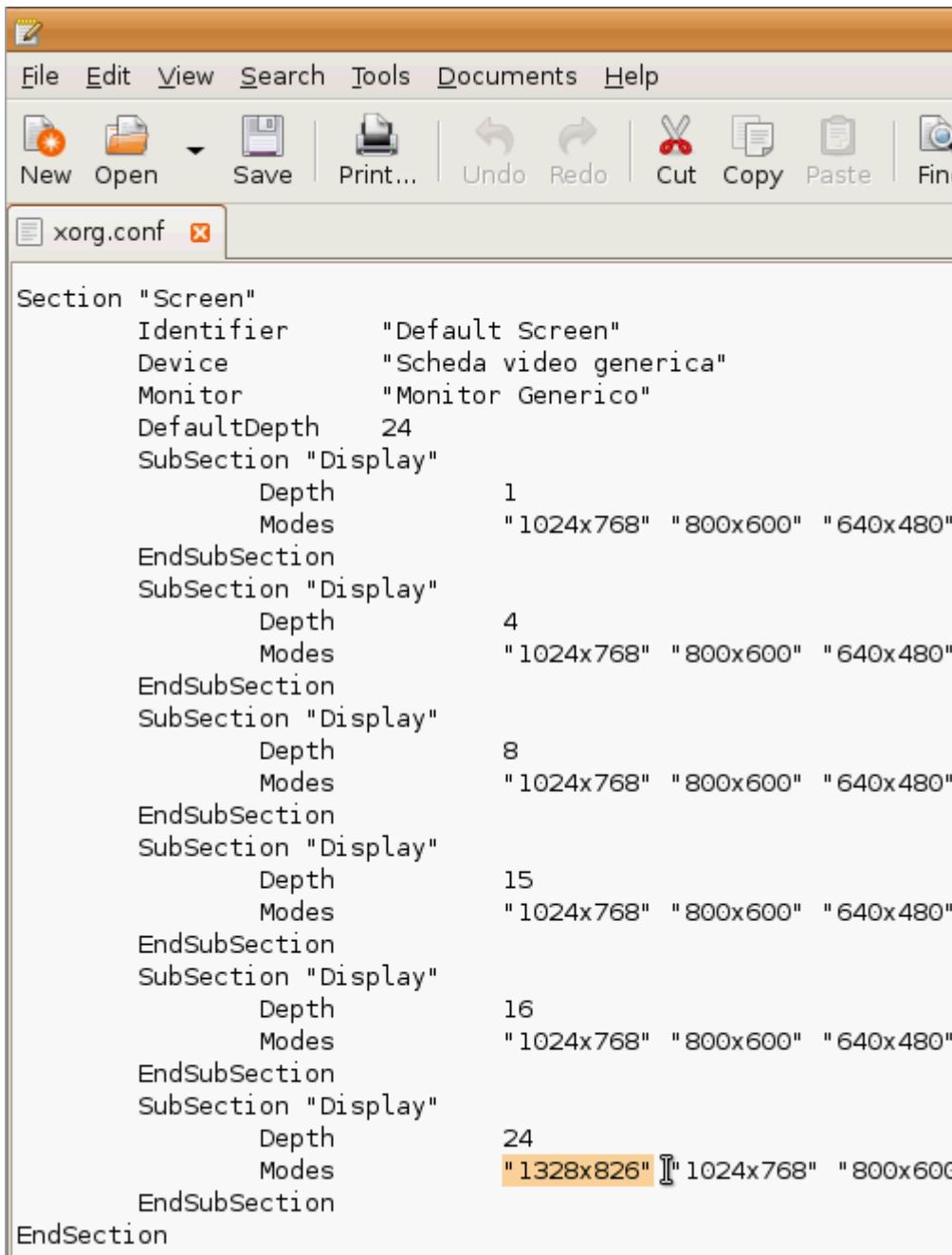
```
Section "Screen"
    Identifier      "Default Screen"
    Device         "Scheda video generica"
    Monitor        "Monitor Generico"
    DefaultDepth   24
    SubSection "Display"
        Depth          1
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          4
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          8
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          15
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          16
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          24
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
EndSection
```

fig. 8

Iniziate con l'aggiungere le istruzioni evidenziate in **fig. 9**, sono la risoluzione video che ho trovato più adatta per il mio portatile.

Ha uno schermo di 1440x900 pixel: visto il grande spazio in larghezza, ho messo la barra delle applicazioni in verticale su uno dei due lati invece che sotto come al solito. Facendo un po' di prove, ho trovato come dimensione ottimale quella che vedete, 1326x828 pixel, un po' meno di quella massima del monitor.

Dovrete fare anche voi un pò di prove tenendo presente la risoluzione del vostro schermo: se ad esempio avete il classico 1280x1024, provate inizialmente con 1200x950, un pò in meno della massima disponibile. Questo perché bisogna tener conto dello spazio sottratto dalla barra delle applicazioni di Windows e dalla barra di stato della finestra di VirtualBox in cui gira Ubuntu.



```
Section "Screen"
    Identifier      "Default Screen"
    Device         "Scheda video generica"
    Monitor        "Monitor Generico"
    DefaultDepth   24
    SubSection "Display"
        Depth          1
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          4
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          8
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          15
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          16
        Modes          "1024x768" "800x600" "640x480"
    EndSubSection
    SubSection "Display"
        Depth          24
        Modes          "1328x826" "1024x768" "800x600"
    EndSubSection
EndSection
```

fig. 9

Fatte le vostre modifiche, uscite da gedit salvando tutto e riavviate Ubuntu per renderle effettive.

Terminato il riavvio, vi accorgerete subito che la dimensione della finestra di VirtualBox è cambiata in accordo con le nuove impostazioni di Ubuntu. Per averne conferma, cliccate sul menu *System --> Preferenze --> Risoluzione dello schermo* (fig. 10).



fig. 10

Guardate **fig. 11**: ora è presente anche la vostra nuova risoluzione.

Se avete usato una risoluzione troppo grande, vedrete comparire delle barre di navigazione nella finestra in cui gira Ubuntu: diminuite la risoluzione fino a quando scompariranno (certo, se poi non vi danno fastidio, allora lasciate tutto così). Continuate a provare risoluzioni video fino a quando non troverete quella per voi più comoda; ricordatevi di riavviare dopo ogni modifica per renderla effettiva.

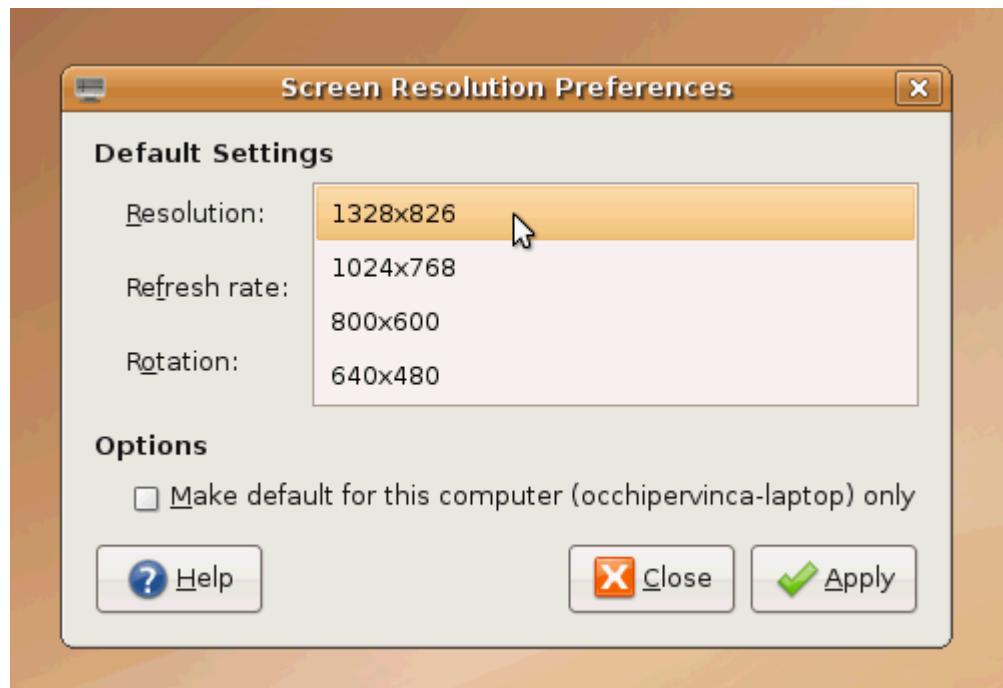


fig. 11

La configurazione della risoluzione video con le VirtualBox Guest Additions in Ubuntu, è terminata; non vi resta che

usarlo. Buon lavoro! :-)

@:\>

Pubblicato alle 10.16 Argomenti: [guida](#), [Linux](#), [software](#), [tutorial](#), [Virtualbox](#), [VirtualBox Guest Additions](#), [virtualizzazione](#)

lunedì 27 agosto 2007

VirtualBox: Guest Additions in una generica distribuzione Linux

Con questo, chiudiamo la lunga serie di post dedicati all'installazione delle VirtualBox Guest Additions. In questa occasione non ci saranno indicazioni particolareggiate, vi darò invece delle indicazioni di massima per agevolarvi nel vostro compito, quando il sistema guest è una qualunque distribuzione Linux.

Iniziate leggendo il post "VirtualBox: Guest Additions in Linux, procedura generale", troverete indicata nei particolari la procedura da seguire. Per procedere poi concretamente, usate come tutorial i post che si riferiscono all'installazione delle VirtualBox Guest Additions:

- in una distribuzione SUSE Linux;
- in una distribuzione Ubuntu Linux.

E con questo, è davvero tutto.

Dal prossimo post cominceremo finalmente a goderci i benefici della loro installazione, non mancate. :-)

@:)>

Pubblicato alle 8.54 Argomenti: guida, Linux, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione

2 commenti:

Riva11 ha detto...

Ottimo lavoro, a presto.

27 agosto 2007 11.35

Giornale di Sistema ha detto...

Grazie. :-)

@:)>

27 agosto 2007 17.35

lunedì 3 settembre 2007

Rilasciato Virtualbox 1.5.0

Così, a sensazione, me lo aspettavo. Appena tornati dalle ferie, i ragazzi della innotek (che volete da me, me li immagini giovanotti di 30 anni che si divertono un mondo a smanettare con le tecnologie di virtualizzazione) hanno reso disponibile una nuova versione di VirtualBox: la 1.5.0.

Mentre scrivo, sta andando il download del programma e del suo manuale utente, appena ho ulteriori informazioni e qualche screenshot, vi aggiorno. Nel frattempo, se siete curiosi e non ce la fate ad aspettare, date un'occhiata al Changelog (in inglese), è davvero molto nutrito ed interessante.

@:\>

Pubblicato alle 10.58 Argomenti: [Linux](#), [news](#), [software](#), [Virtualbox](#), [virtualizzazione](#), [Windows](#)

martedì 4 settembre 2007

VirtualBox: versione 1.5.0, tutte le novità

Nel post "Rilasciato Virtualbox 1.5.0" vi segnalavo ieri, l'uscita fresca fresca della versione 1.5.0 di VirtualBox. Eccoci oggi a guardare più da vicino alcune delle novità, che, se avete già dato un'occhiata al Changelog segnalato sempre ieri, avrete notato sono davvero corpose.

Solo alcune, in ordine sparso:

- 64 bit anche su sistemi host Windows;
- 2 GB di memoria assegnabile ad ogni macchina guest (contro i precedenti 1,2 GB);
- miglioramenti nel supporto alle estensioni per la virtualizzazione implementate nei processori Intel e AMD e relativo settaggio direttamente dall'interfaccia grafica di VirtualBox;
- introduzione delle porte seriali sulle macchine virtuali;
- supporto alle DirectDraw su guest Windows per migliori performance video;
- supporto Intel PXE 2.1 per il boot da rete;
- Guest Additions anche per IBM OS/2 Warp;
- supporto per più monitor su sistemi guest Windows;
- possibilità di scaricare le VirtualBox Guest Additions direttamente da Internet se non sono presenti in locale;
- supporto a Xorg 1.3 in Linux (per cui, ad esempio, ora Fedora 7 è supportato);
- indicatore di attività sulla barra di stato anche per i dispositivi USB e le cartelle condivise.

Miglioramenti vari, tesi soprattutto a migliorarne stabilità e prestazioni, si registrano poi in quasi ogni area di VirtualBox. Guardavo tra l'altro le dimensioni delle Guest Additions: da poco meno di 3 MB, sono passate a quasi 5 MB. Così pure la dimensione complessiva del file d'installazione di VirtualBox: da poco più di 15 a quasi 17 MB.

Una delle feature introdotte nelle nuove Guest Additions è presente solo su sistemi guest Windows. Si tratta di una nuova modalità di visualizzazione della finestra guest chiamata Seamless windows (letteralmente, senza cucitura), nella traduzione italiana, *Modalità Trasparente*. Ve ne parlo in [fig. 9](#), ora addentriamo pian piano nel nuovo VirtualBox; venite con me.

Selezionate una macchina virtuale (ho preso a caso una SUSE Linux) e invocate le sue impostazioni dall'apposito menu (fig. 1).

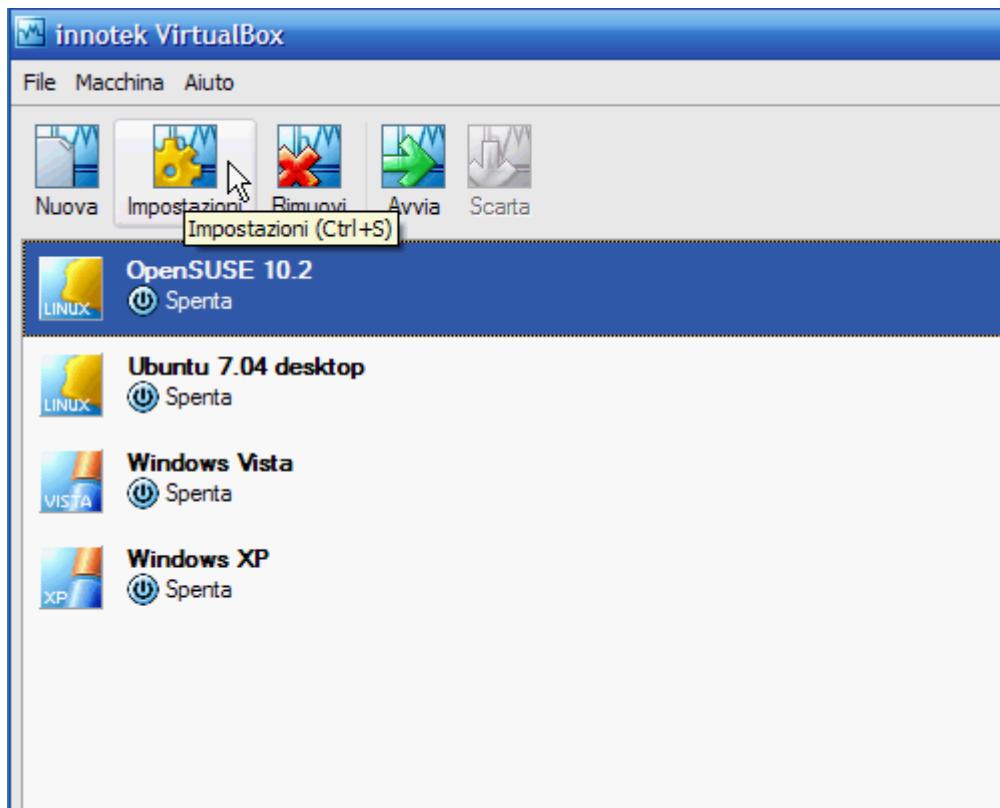


fig. 1

Vi accorgerete subito della prima differenza: ora la memoria base a disposizione di ogni macchina virtuale è stata

incrementata a 2 GB dai precedenti 1,2 (fig. 2).

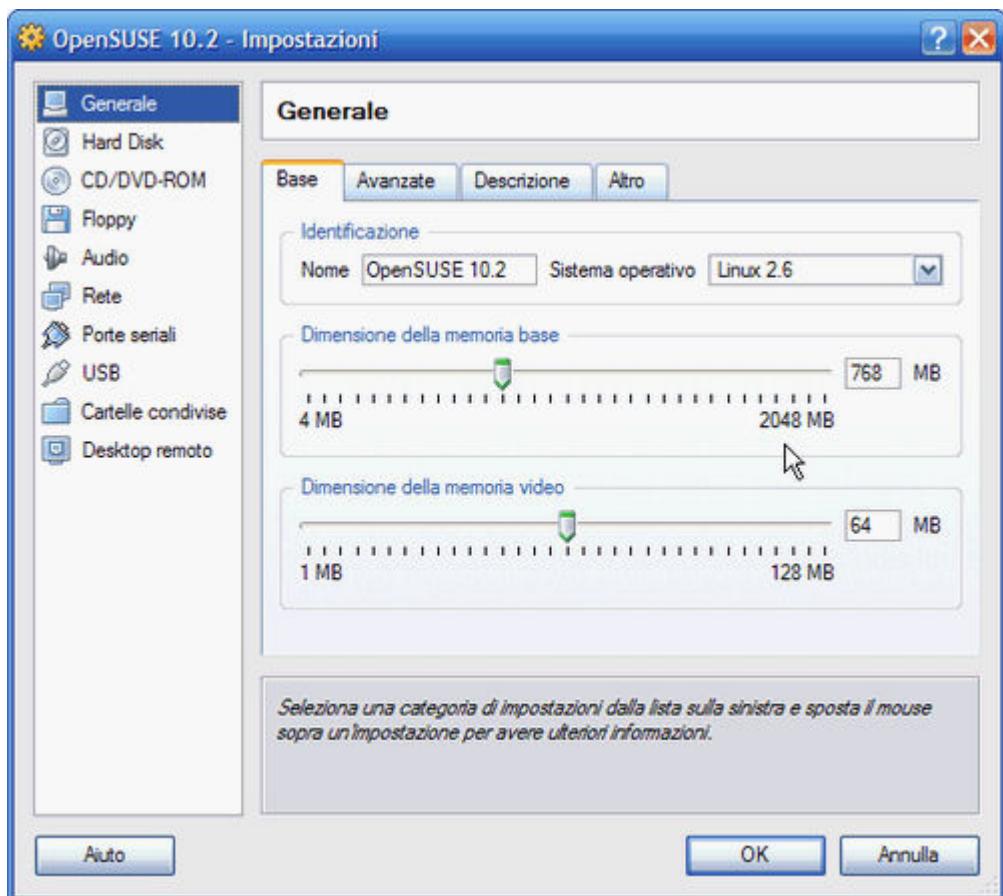


fig. 2

E non è ancora finita: sempre nelle impostazioni *Generale*, alla scheda *Avanzate* (fig. 3), è stata aggiunta la nuova voce *Abilita VT-x/AMV-V* che permette di attivare, direttamente dall'interfaccia grafica di VirtualBox, l'uso delle estensioni per la virtualizzazione contenute in tutti i più recenti processori Intel e AMD del vostro sistema host.

2 annotazioni:

1) fate attenzione al primo click: non compare il classico segno di spunta (come nella voce *Abilita ACPI* per intenderci, quella immediatamente sopra), ma un quadratino. In questo caso l'effettiva attivazione delle estensioni per la virtualizzazione dipende dalle impostazioni globali.

Per verificare, cliccate sul menù *File* --> *Impostazioni...* di fig. 1: se in *Generale* la voce *Abilita VT-x/AMV-V* è selezionata, allora la macchina virtuale userà le estensioni, altrimenti no.

Torniamo su fig. 3: se date un secondo click sulla voce *Abilita VT-x/AMV-V*, comparirà il classico segno di spunta: in questo caso le estensioni vengono usate sempre, indipendentemente dalle impostazioni globali appena viste.

2) le estensioni di default sono disattivate. Leggendo il manuale, questi suggerisce di lasciarle disattivate in quanto VirtualBox "usa sofisticate tecniche software che normalmente assicurano prestazioni migliori comparate alla virtualizzazione hardware". Se ne consiglia l'abilitazione solo per sistemi operativi guest "esotici", come esempio viene indicato IBM OS/2 Warp.

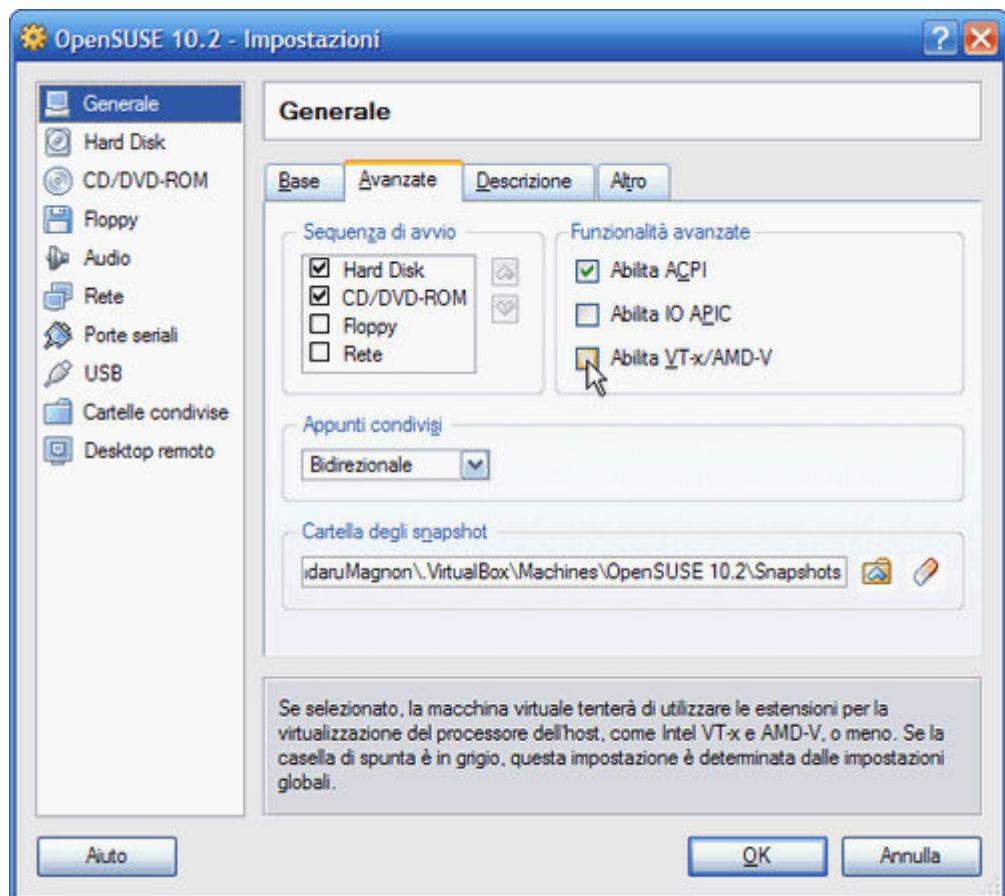


fig. 3

Un'altra nuova funzionalità la vediamo in **fig. 4**. Se *Ricorda media montati a runtime* è abilitata, VirtualBox salva nei suoi file di impostazioni lo stato di un CD/DVD o Floppy montato a runtime preservandolo così ad ogni riavvio della macchina virtuale.

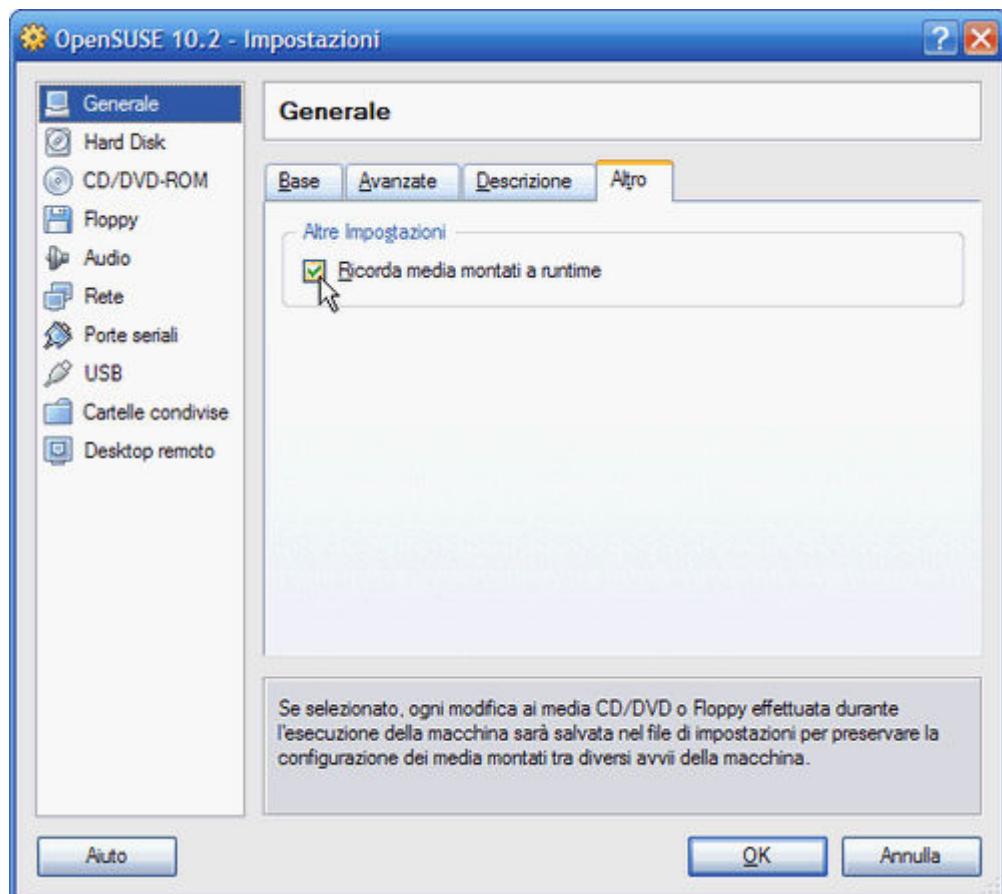


fig. 4

In **fig. 5** vedete invece la nuova possibilità di usare il masterizzatore CD/DVD dell'host anche nel sistema guest. Fate attenzione però: come esplicitamente riportato nella didascalia che compare quando con il mouse passate su una qualunque opzione di VirtualBox (in basso, proprio sopra il pulsante *OK*), non è ancora supportata la scrittura di CD audio direttamente dall'interno del guest.

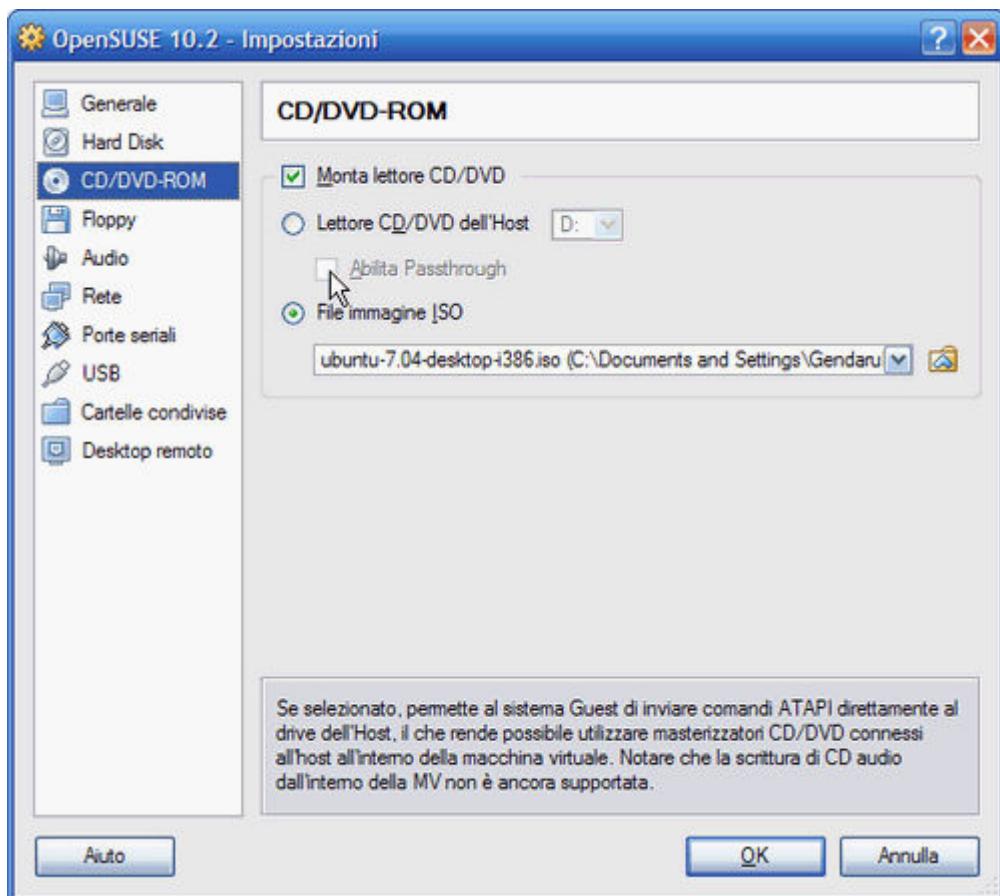


fig. 5

In **fig. 6** potete vedere una delle novità più grosse. Ora il PC virtuale simulato da VirtualBox non ha più solo dispositivi USB, Floppy, CD-DVD ROM, ma anche le porte seriali. Ormai non sono più tantissimi i dispositivi che ancora usano la porta seriale come interfaccia di collegamento al computer, ma modem esterni 56k, vecchie stampanti (la mia HP Deskjet 690C) possono forse trovare nuova vita anche all'interno della vostra macchina virtuale.

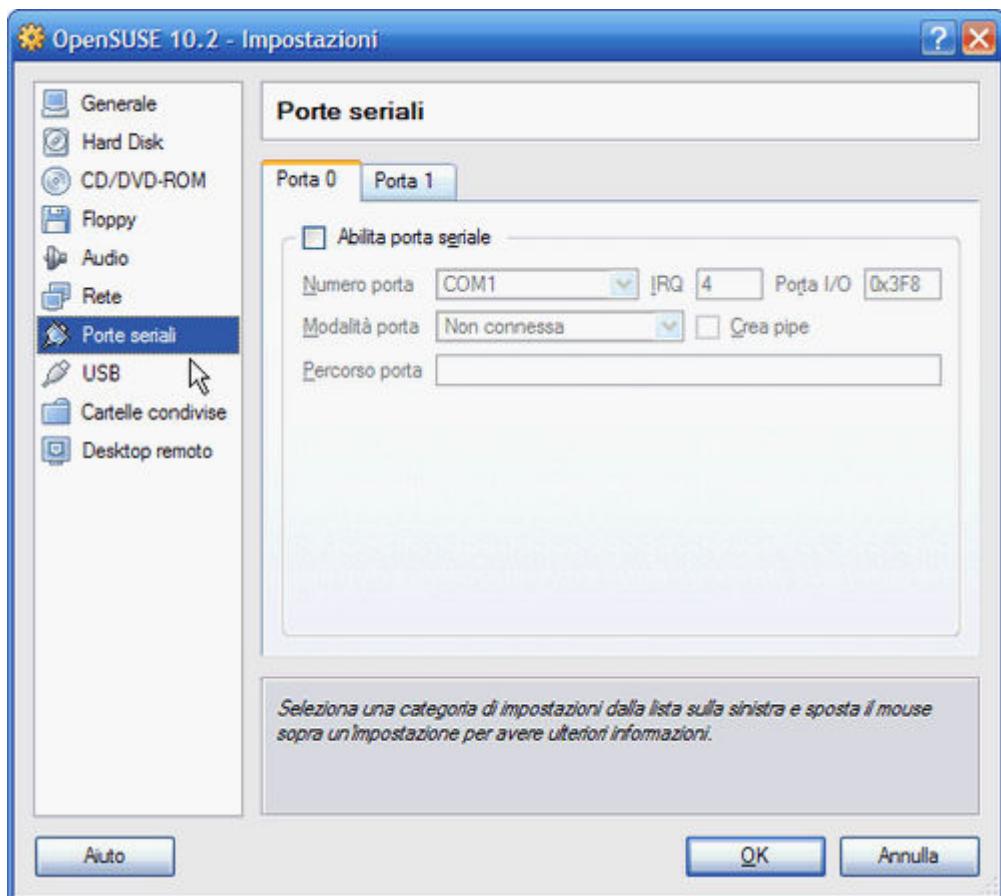


fig. 6

Ho fatto qualche giro di prova con il VirtualBox nuovo fiammante, e devo dire che sono soddisfatto. In particolare, nessun problema con le macchine virtuali create con la precedente versione 1.4, tranne che per un semplice box informativo relativo alla risoluzione video (fig. 7, un sistema Ubuntu, ma compare in tutti i guest, Linux o Windows che siano). Nel sistema guest era impostato con colore a 16 bit (65.536 colori) mentre la finestra VirtualBox che ospitava il guest è ottimizzata per i 32 bit; se volete eliminare la segnalazione, potete: o adeguare il numero di colori nel sistema guest passando a 32 bit, oppure spuntare la voce *Non mostrare ancora questo messaggio* che vedete in fig. 7.

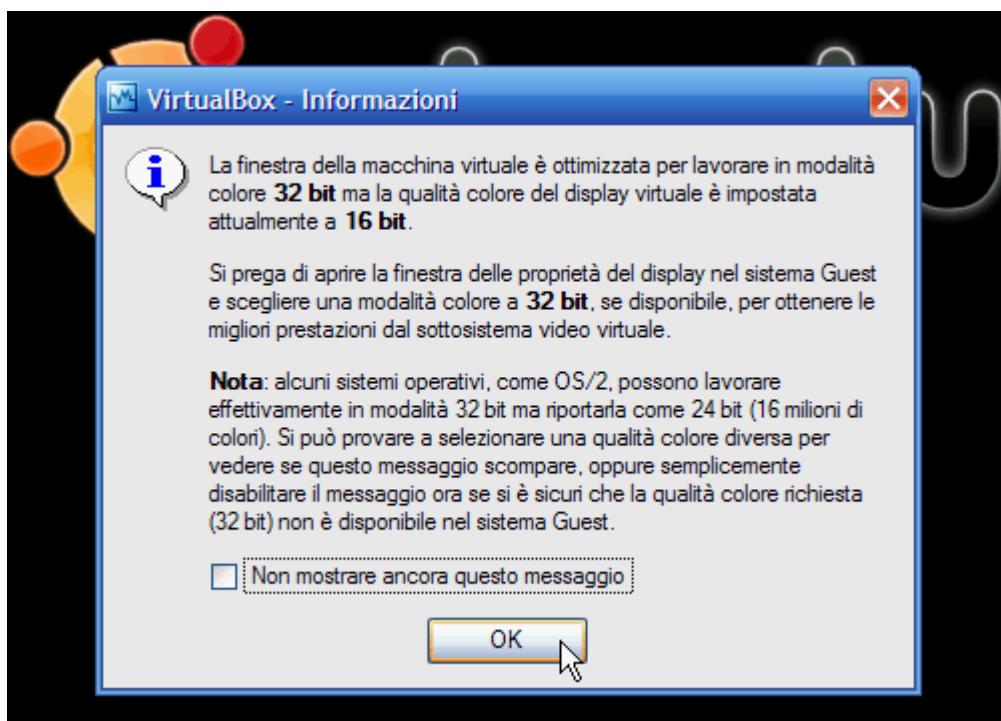


fig. 7

L'ultima novità a cui diamo un'occhiata è la modalità Seamless windows (letteralmente, senza cucitura), tradotta come **Modalità Trasparente** (fig. 8).

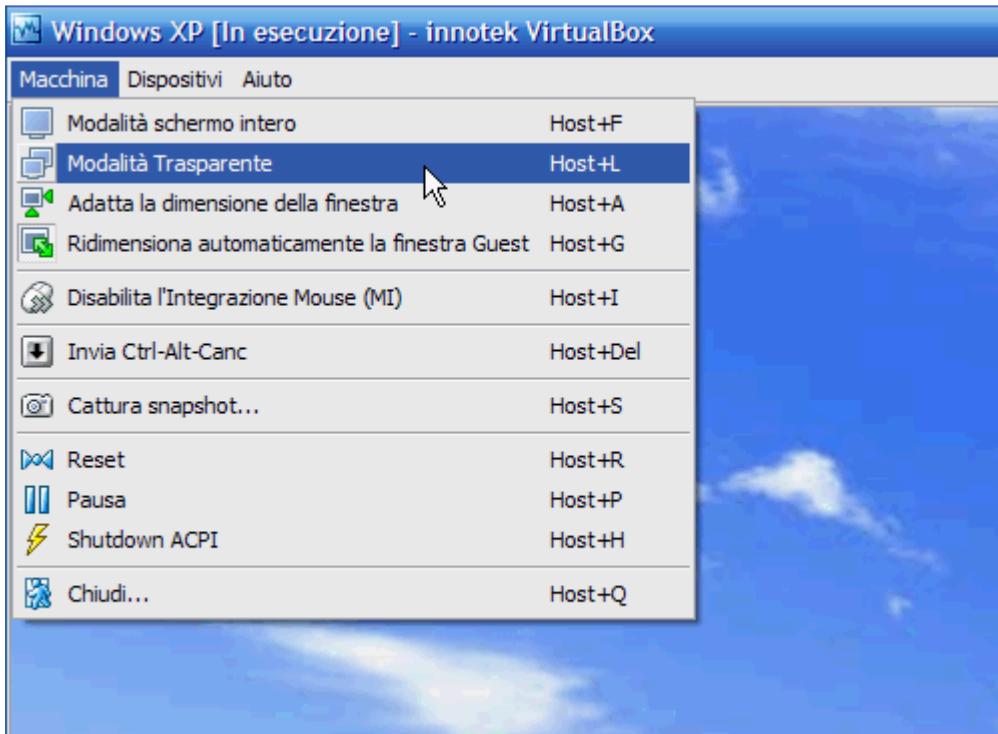


fig. 8

Essa permette di far scomparire la finestra del guest in modo che le sue applicazioni, una volta aperte, sembrano aperte sul desktop dell'host. In fig. 9 potete vederne l'effetto, il Blocco note è stato aperto nel guest, poi è stata attivata la Modalità Trasparente: la finestra che ospitava il guest è scomparsa, e sotto, è comparso il desktop dell'host con l'icona di VirtualBox ed il suo manuale pdf.

E potete notare anche un'altra cosa: dei problemi di visualizzazione in questa nuova modalità. Almeno sul mio sistema, si è creato un effetto scia quando ho trascinato in giro per lo schermo la finestra del Blocco note. Cliccando sul desktop dell'host con il tasto destro e poi sulla voce *Aggiorna*, la scia compare, per ricomparire appena la finestra viene nuovamente trascinata.

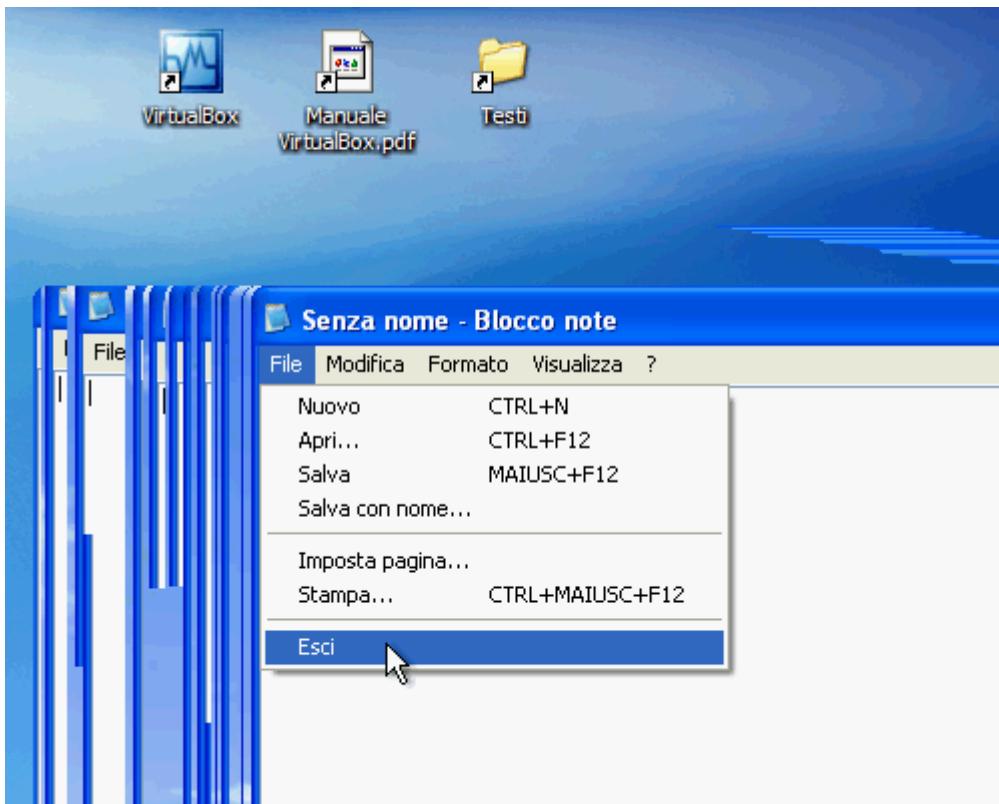


Fig. 9

Bene, la carrellata sulle principali novità di VirtualBox è terminata. Nel prossimo post vedremo come farne l'update dalla vecchia versione salvaguardando l'integrità delle macchine virtuali già create, e come aggiornare le Guest Additions presenti in quest'ultime.

@:>

Pubblicato alle 15.03 Argomenti: guida, Linux, news, software, Virtualbox, virtualizzazione, Windows

mercoledì 5 settembre 2007

VirtualBox: aggiornarlo alla versione 1.5.0

Come sa chi ha già usato le versioni precedenti di VirtualBox, non esiste una procedura di aggiornamento vera e propria del programma: occorre prima disinstallare la vecchia versione, poi installare la nuova. La disinstallazione non va a toccare le macchine virtuali che avete già creato, ma per sicurezza salvatele da qualche parte prima di procedere. Una volta finito tutto, mi raccomando, non dimenticate di aggiornare anche le Guest Additions all'interno di ogni singola macchina virtuale.

Riassumendo, occorre procedere così:

- salvataggio delle macchine virtuali eventualmente esistenti;
- disinstallazione di VirtualBox;
- installazione della nuova versione di VirtualBox;
- aggiornamento delle Guest Additions in ogni macchina virtuale dall'interno della stessa (a breve il link).

E' tutto, buon lavoro! :-)

@:\>

Pubblicato alle 6.52 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione

giovedì 6 settembre 2007

VirtualBox: aggiornare le Guest Additions

L'aggiornamento delle VirtualBox Guest Additions si rende necessario nel momento in cui aggiorniamo VirtualBox e usiamo le macchine virtuali create con la versione precedente. Le Guest Additions infatti, vengono installate nelle macchine virtuali che non vengono toccate dall'aggiornamento di VirtualBox, dovete pertanto provvedere manualmente al loro aggiornamento in ogni singola macchina.

Fortunatamente la procedura è molto semplice sia su guest Windows che Linux (su quest'ultimi un po' meno, ma niente a che vedere con le difficoltà della loro installazione ex-novo): è sufficiente provvedere al loro aggiornamento diretto, senza ulteriori fronzoli come eventuali disinstallazioni; vediamolo in concreto con 2-3 esempi, sia in guest Windows che Linux.

In Windows:

Se proprio volete, perdeteci un attimo di tempo e disinstallate prima la versione precedente delle Guest Additions (**fig.1**), quando è terminata, e sempre per esser fiscali, riavviate la macchina virtuale.

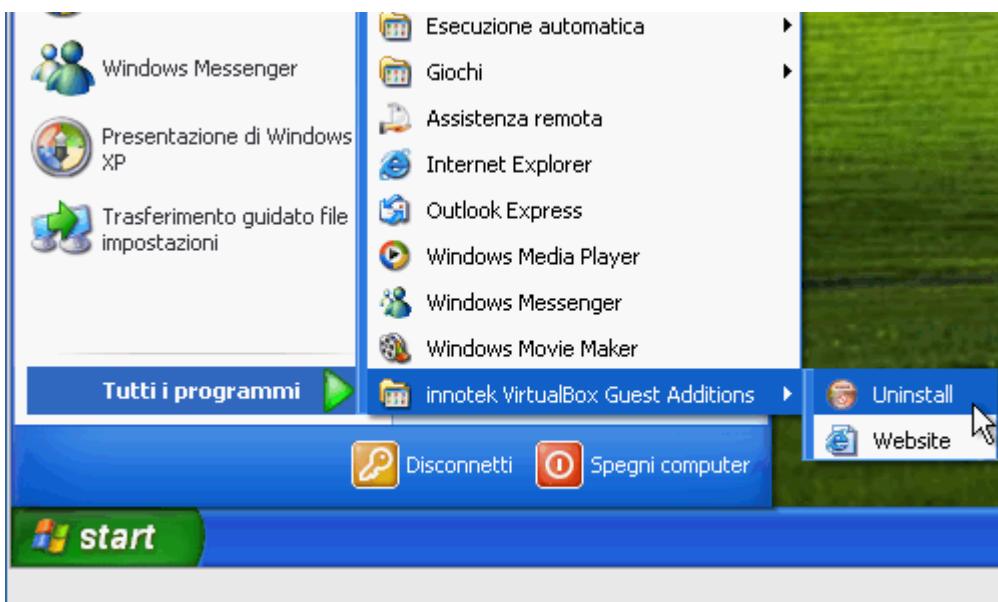


fig. 1

Terminato il riavvio, procedete con l'installazione della nuova versione delle Guest Additions (**fig. 2**), infine riavviate di nuovo; tutto qui.

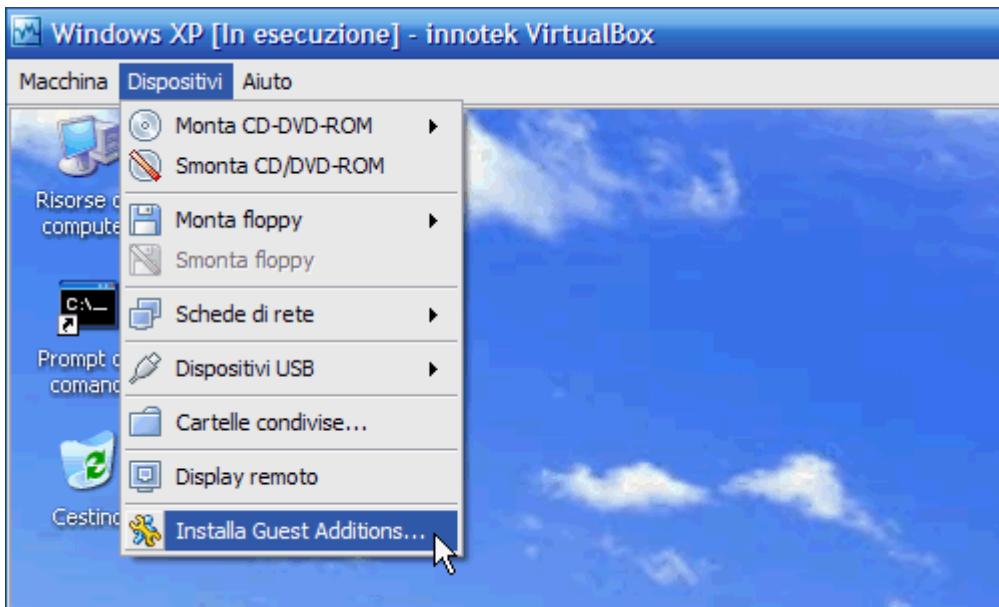


fig. 2

Se avete bisogno di maggiori informazioni sulla loro reinstallazione, consultate i post relativi a Windows XP, Windows Vista, oppure un'altra versione di Windows, sono comunque tutte molto simili fra loro.

In una generica distribuzione Linux:

L'installazione ex-novo delle Guest Additions in una generica distribuzione Linux è in 3 fasi:

- 1) attivazione supporto moduli di kernel esterni,
- 2) installazione delle Guest Additions (in realtà si tratta del mount del file iso nella vostra macchina virtuale,
- 3) compilazione e attivazione delle Guest Additions.

Trattandosi di un aggiornamento, la fase 1 è già stata fatta, occorre solo portare a termine la fase 2 e la fase 3. Fate pertanto riferimento solo a quest'ultime due nel relativo post "VirtualBox: Guest Additions in Linux, procedura generale"

In SUSE Linux:

Anche qui, come per la versione generica, la fase 1 è stata portata a termine già in precedenza, attenetevi solo alle fasi 2 e 3, trattate rispettivamente nei post "VirtualBox: installare le Guest Additions in SUSE, fase 2" e "VirtualBox: installare le Guest Additions in SUSE, fase 3".

In Ubuntu Linux:

Stesso discorso: la fase 1 non serve, è già stata fatta. Pertanto nel relativo post "VirtualBox: Guest Additions in Ubuntu Linux" cominciate a seguirne le indicazioni da **fig. 6**.

Bene, spero di non aver dimenticato nulla; nel caso di dubbi, fatemi sapere.
@:>

Pubblicato alle 8.26 Argomenti: guida, Linux, software, tutorial, Virtualbox, VirtualBox Guest Additions, virtualizzazione, Windows

3 commenti:

Michele Benvegnù ha detto...

Utilizzo sia VirtualBox che Microsoft Virtual PC, su host Vista Business preinstallato su un PC Olidata. Ho configurato alcune macchine virtuali con guest Vista, utilizzando ovviamente il CD di sistema creato dal PC

Oidata, e pertanto originale.

Per alcune settimane le macchine virtuali Vista hanno regolarmente funzionato; da qualche giorno (dopo 3 mesi?) si bloccano e richiedono l'inserimento di un nuovo Product Key; se digito quello del mio Vista originale, ottengo un messaggio di errore, che informa che il Product Key è già in uso (ovvio: è utilizzato dal mio PC "fisico" Oidata).

Si pone pertanto l'annoso problema delle licenze Microsoft su macchine virtuali.

Dal "Contratto di licenza per il software Microsoft Windows Vista Business" (un pdf che si può scaricare da microsoft.com) riporto testualmente al punto 3-f:

"Utilizzo con le Tecnologie di Virtualizzazione. Il licenziatario potrà utilizzare il software installato sul dispositivo con licenza all'interno di un sistema hardware virtuale..."

Al contrario, Vista installato su Pc virtuale si blocca. Da notare che si blocca anche utilizzando Virtual PC, il software di virtualizzazione della stessa Microsoft.

Ma allora a cosa servono i software di virtualizzazione? Non crederà Microsoft che un utente acquisti a caro prezzo una licenza per ciascuna macchina virtuale? Anche perché deve essere chiaro un concetto: una sola licenza per ciascun PC fisico, qualunque sia il numero delle macchine virtuali utilizzate!!! Se non altro, perché un PC fisico non può essere utilizzato contemporaneamente da due utenti.

È sin troppo facile procurarsi un DVD pirata di Vista, che non esiterà ad utilizzare perché mi sento imbrogliato e derubato da Microsoft.

6 settembre 2007 18.58

Giornale di Sistema ha detto...

Problema annoso... che continua ad ingigantirsi.

Da un lato i produttori di software hanno tutto il diritto di vedersi riconosciuto un giusto tributo per i loro sforzi.

Dall'altro l'acquirente del software, che avendolo pagato, si aspetta, alle condizioni della licenza stessa, di poterlo pienamente utilizzare.

Invece, ad esempio e per restare in tema, è notizia dei giorni scorsi, che il Windows Genuine Advantage (WGA), tecnologia di autenticazione della Microsoft che distingue un Windows autentico da uno non autentico, ha avuto problemi. E così, almeno da quanto ho letto in giro per la rete, sono state migliaia le macchine dichiarate con software non autentico quando al precedente controllo lo erano.

La situazione ha trovato rimedio in poche ore, e va benissimo. Ma non va altrettanto bene che la mia macchina da un momento all'altro possa smettere di funzionare per l'ansia da controllo di qualcuno. Con il PC ci lavoro, e ho pagato per poter avere un software che lo faccia funzionare quando dico io: ma in qualunque momento e al di fuori di qualunque mio controllo, qualcuno può impedirmelo a suo insindacabile giudizio.

C'è di che riflettere.

@:>

7 settembre 2007 21.45

Giornale di Sistema ha detto...

Ulteriore notizia di questi giorni è che, senza avvisare nessuno, sono state aggiornate alcune componenti di Windows Update, il servizio che si occupa di tenere Windows sempre aggiornato con le ultime patch. Tutti siamo abituati a vedere un'icona di notifica nella System Tray che ci avvisa della presenza di nuovi aggiornamenti, ma questa volta niente, e soprattutto, la cosa è avvenuta su macchine a servizio disattivato.

Sembra che la posizione Microsoft sia quella che "si tratta ormai di una prassi consolidata e necessaria per tenere Windows Update stesso aggiornato anche sulle macchine client, pena l'impossibilità di determinare successivamente le corrette patch da installare" e che stanno comunque lavorando per rendere più chiaro agli utenti il comportamento di Windows Update...

... magari un banale messaggio di avviso? Continua ad esserci di che riflettere.

@:>

14 settembre 2007 23.23

mercoledì 26 settembre 2007

VirtualBox: cartelle condivise in guest Windows (1a parte)

Le *cartelle condivise* sono un'altra interessante possibilità offerta dall'installazione delle VirtualBox Guest Additions. Permettono infatti di condividere in maniera estremamente semplice ed immediata cartelle fra host e guest senza dover usare la condivisione in rete: è VirtualBox stesso ad occuparsene tramite un apposito servizio che gira sul sistema host e un driver per il file system nel guest.

Non dovete stare più a pasticciare con la configurazione della rete per fare la condivisione e potrete scegliere in qualunque momento, la cartella o le cartelle da condividere: potrete infatti farlo prima di far partire la macchina virtuale, ma anche a macchina virtuale già in funzione.

Ed in quest'ultimo caso avrete a disposizione un'ulteriore modalità di condivisione chiamata *cartella condivisa transiente*: funziona come una normale cartella condivisa ma la condivisione viene rimossa allo spegnimento della macchina virtuale.

Della condivisione cartelle ve ne avevo già parlato in occasione della creazione di una macchina virtuale, e avevo indicato all'interno dei commenti, come superare un problema quando si usano spazi nel *Nome cartella* di VirtualBox, guardate **fig. 7**. Ebbene, dalla versione 1.5 di VirtualBox tutto ciò è stato corretto: ora non solo potete dare un nome qualsiasi alla cartella condivisa in VirtualBox, ma anche tutta la procedura è stata molto semplificata, soprattutto in presenza di guest Windows.

Non vi è più bisogno infatti di dare comandi strani dal DOS, appena la cartella condivisa viene aggiunta, basta premere F5 nell'Esplora risorse del guest per aggiornare la visualizzazione, ed è subito disponibile. E nulla vieta, anche se la funzione si chiama "condivisione cartelle", di **condividere l'intero disco rigido dell'host** come potete vedere in **fig. 1** nel punto indicato dal mouse.

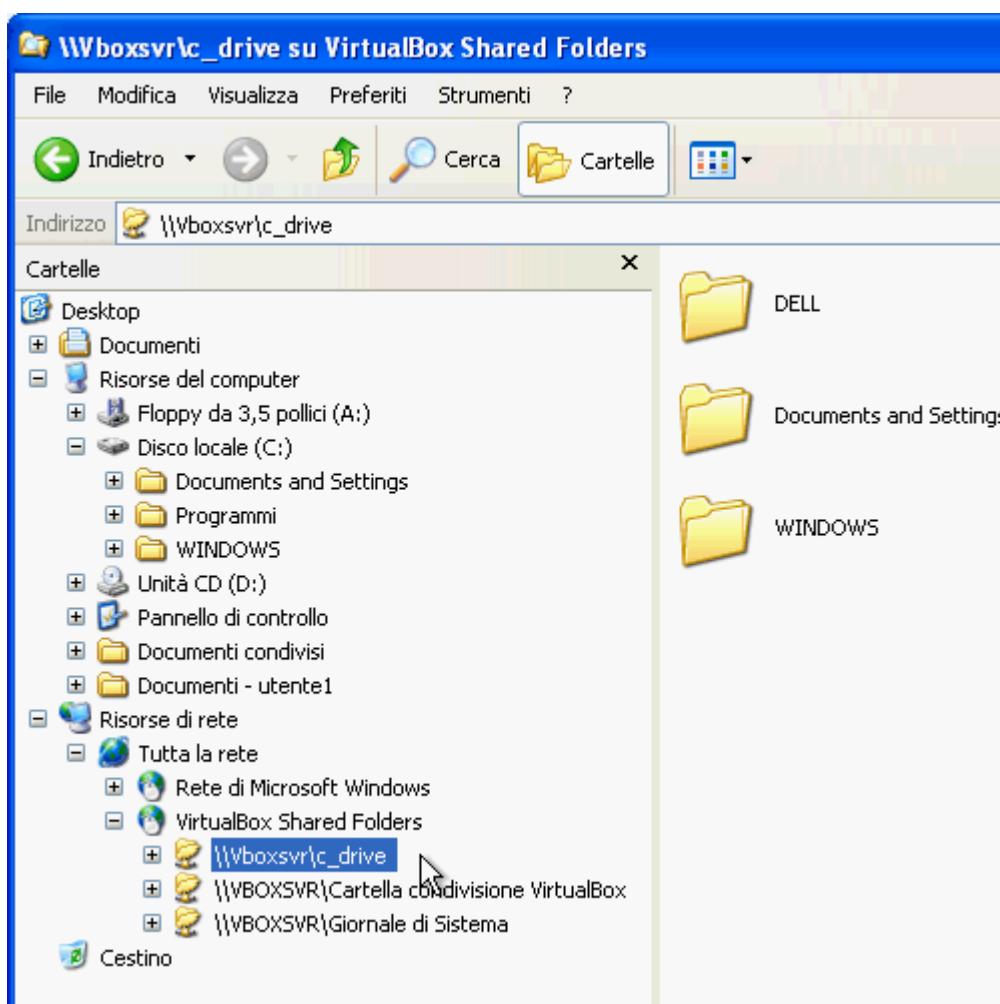


fig. 1

Ok, fatta questa lunga premessa, ora diamoci dentro. La macchina host è un sistema Windows, per questo primo esempio useremo Windows anche per il sistema guest (un guest Linux lo vedremo in post successivi). Vogliamo condividere la cartella sull'host che vedete in **fig. 2** contenente Prevx CSI: l'idea è di fare una scansione antimalware sul sistema guest lanciando il software direttamente dall'host; vediamo se l'esperimento riesce.

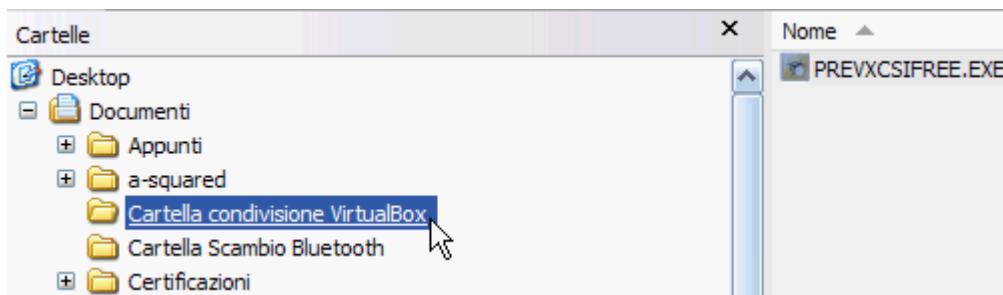


fig. 2

Cominciate con l'avviare VirtualBox e selezionate la macchina guest Windows (**fig. 3**). Notate la voce *Rete* tra le voci sopra quella di *Cartelle condivise*: la rete è disabilitata, proprio per mostrarvi che la funzione condivisione cartelle è un'altra cosa dalla condivisione fatta tramite la rete e funziona a prescindere da questa. Cliccate sulla voce *Cartelle condivise* per passare alle impostazioni delle cartelle condivise.

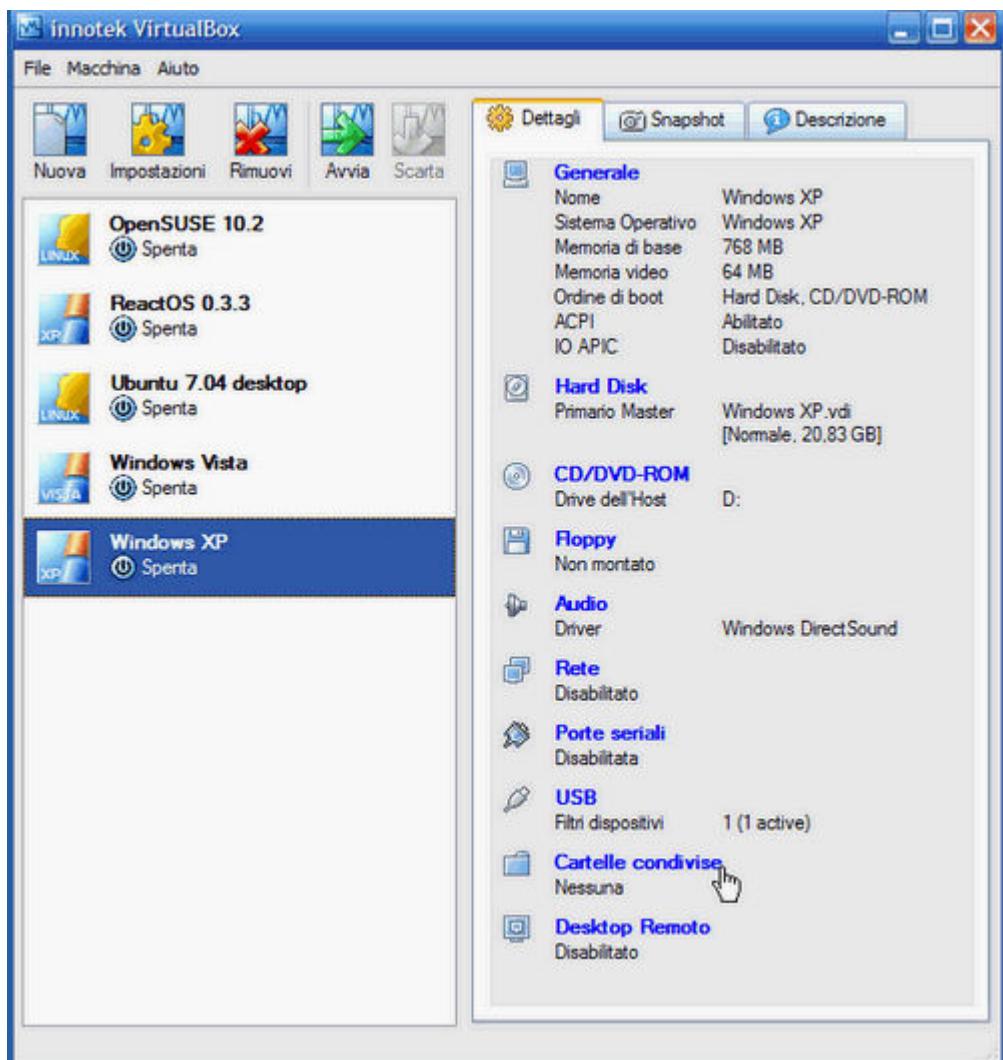


fig. 3

Vi ritroverete nella schermata di aggiunta e modifica delle cartelle condivise (**fig. 4**), come potete notare non ve ne sono ancora. Iniziamo con il condividere la cartella di **fig. 2** contenente Prevx CSI.

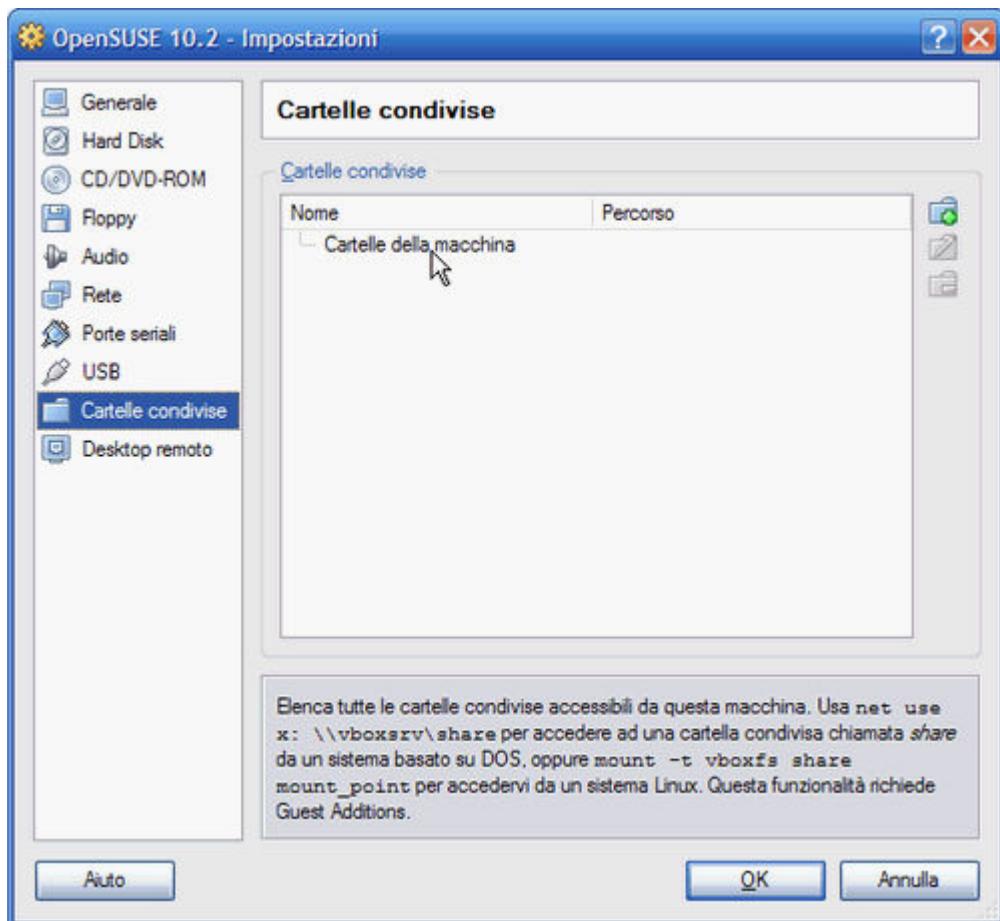


fig. 4

Cliccate sul pulsante di aggiunta di una cartella da condividere indicato dalla posizione del mouse in **fig. 5** (l'indicazione di cosa fa il pulsante compare anche nel box informativo sopra il pulsante OK).

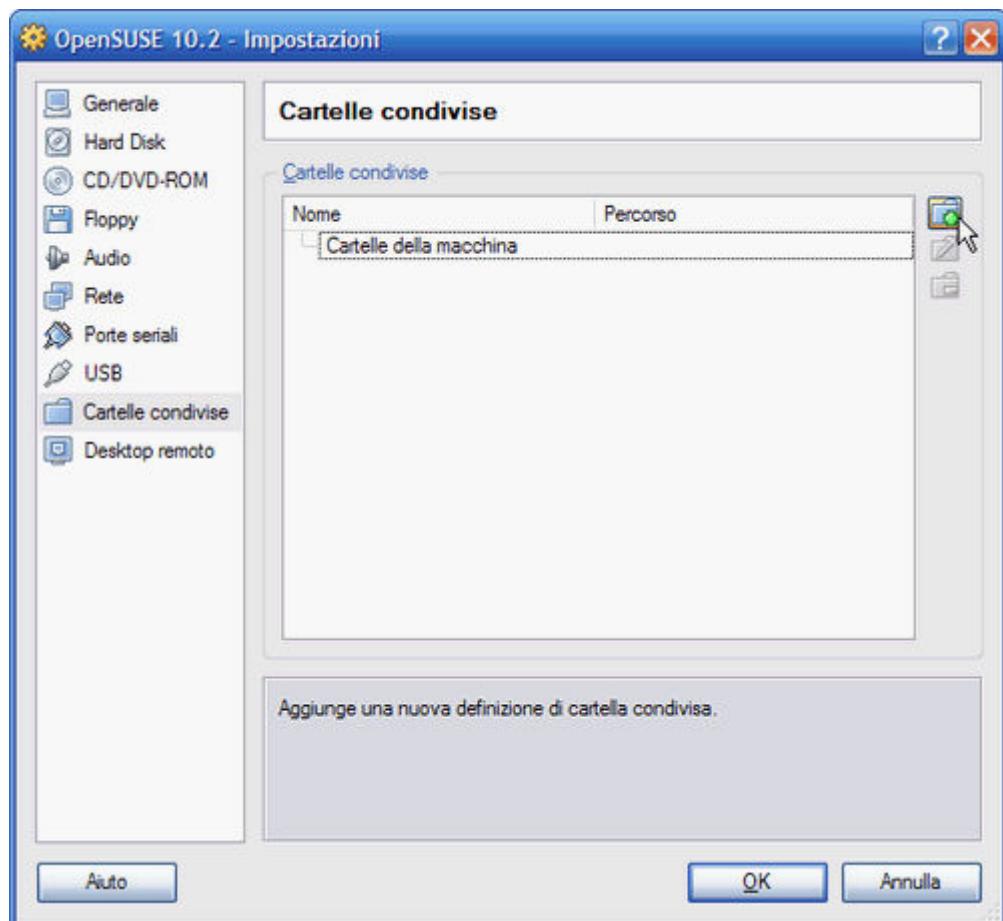


fig. 5

Vi comparirà la piccola finestra di *Aggiungi condivisione* visibile in **fig. 6**, cliccate sul pulsante indicato dalla posizione del mouse per proseguire.

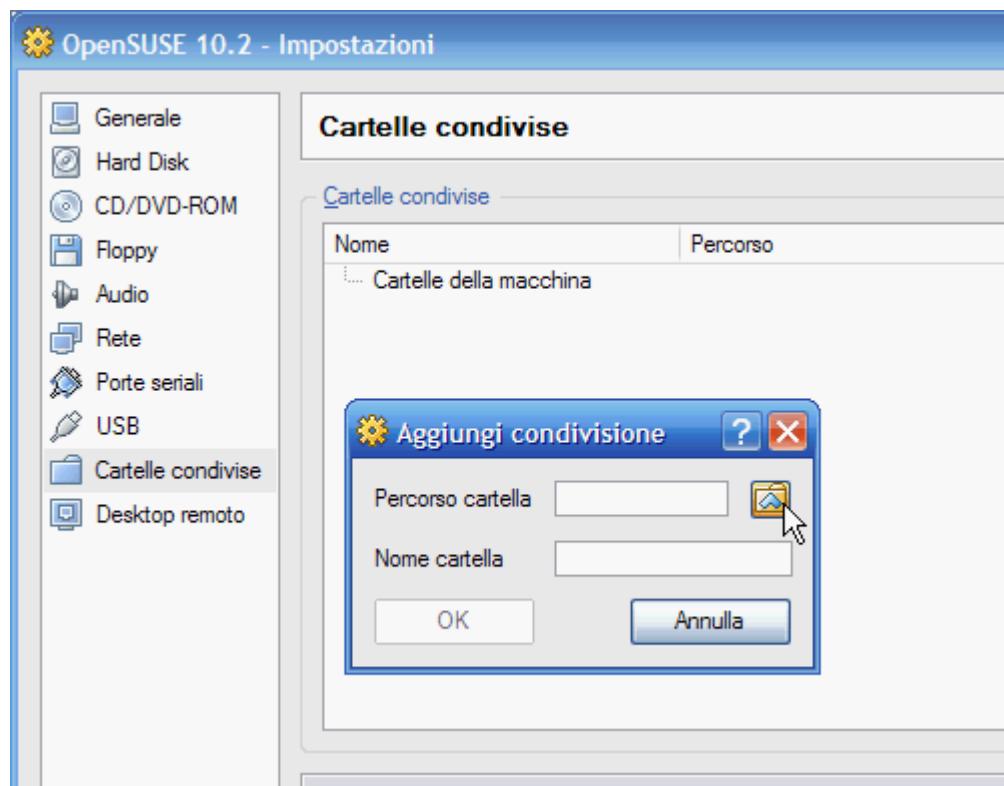


fig. 6

Tramite la finestra di navigazione che vi comparirà, trovate la cartella da condividere sull'host e selezionatela, in *Percorso cartella* comparirà il percorso completo della cartella scelta (fig. 7). In *Nome cartella* assegnamele il nome con cui volete vi appaia nel Guest. Di default VirtualBox le assegna lo stesso nome che ha sull'host; fatta la vostra scelta, cliccate su *OK* per proseguire. Ricordatevi che il problema con gli spazi nel nome è stato risolto per cui va bene se ci sono.

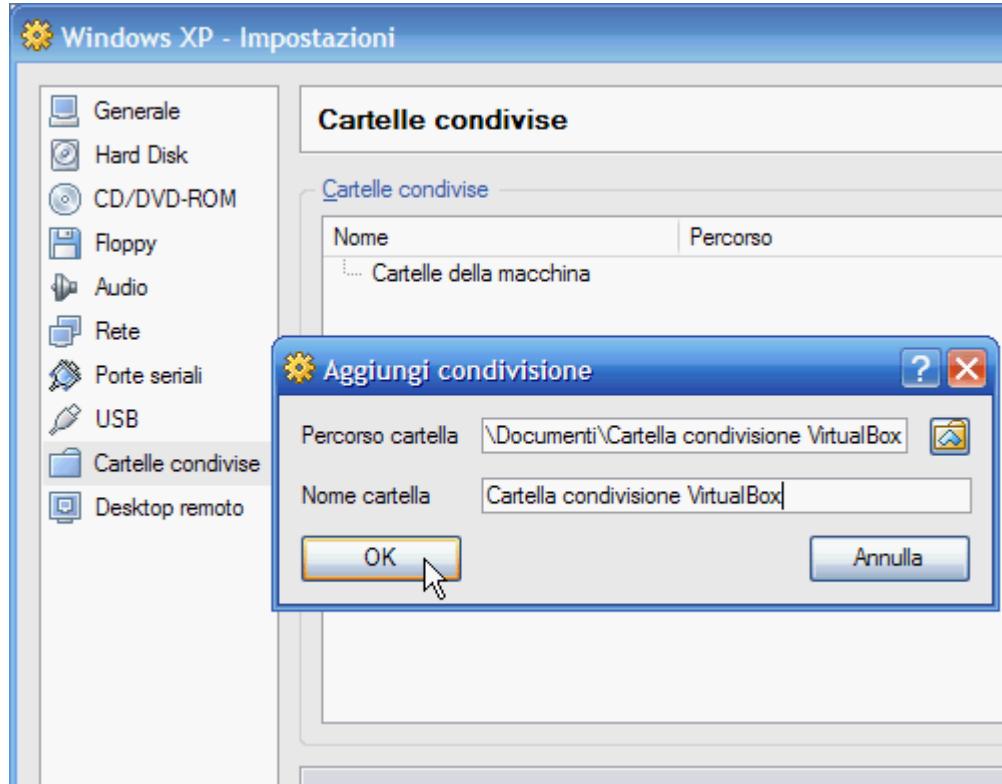


fig. 7

Ed eccola la vostra cartella condivisa, compare finalmente nelle impostazioni della macchina virtuale (fig. 8). Cliccate su *OK* per accettare le modifiche alla macchina virtuale e proseguire.

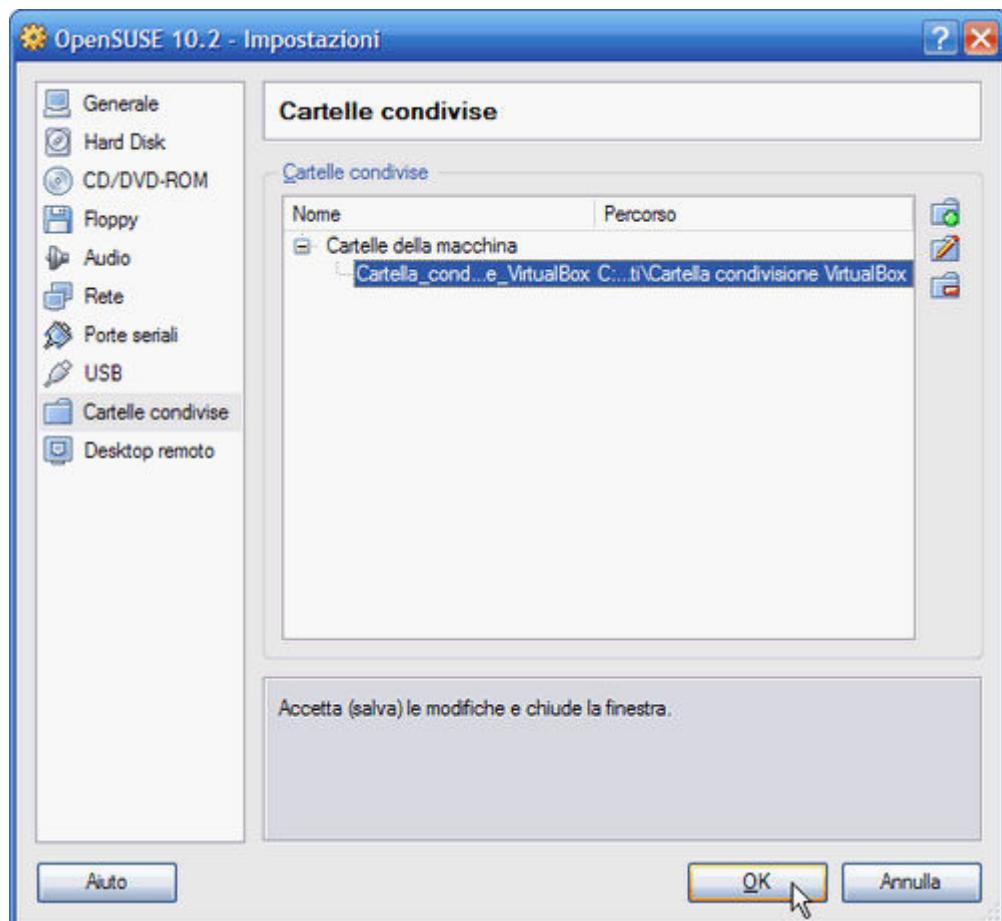


fig. 8

Se tutto è andato bene, ora nei dettagli della macchina virtuale prescelta compare l'indicazione che questa ha una cartella condivisa (fig. 9). Assicuratevi che la macchina virtuale sia selezionata (altrimenti selezionatela) e fatela partire cliccando sul pulsante Avvia.

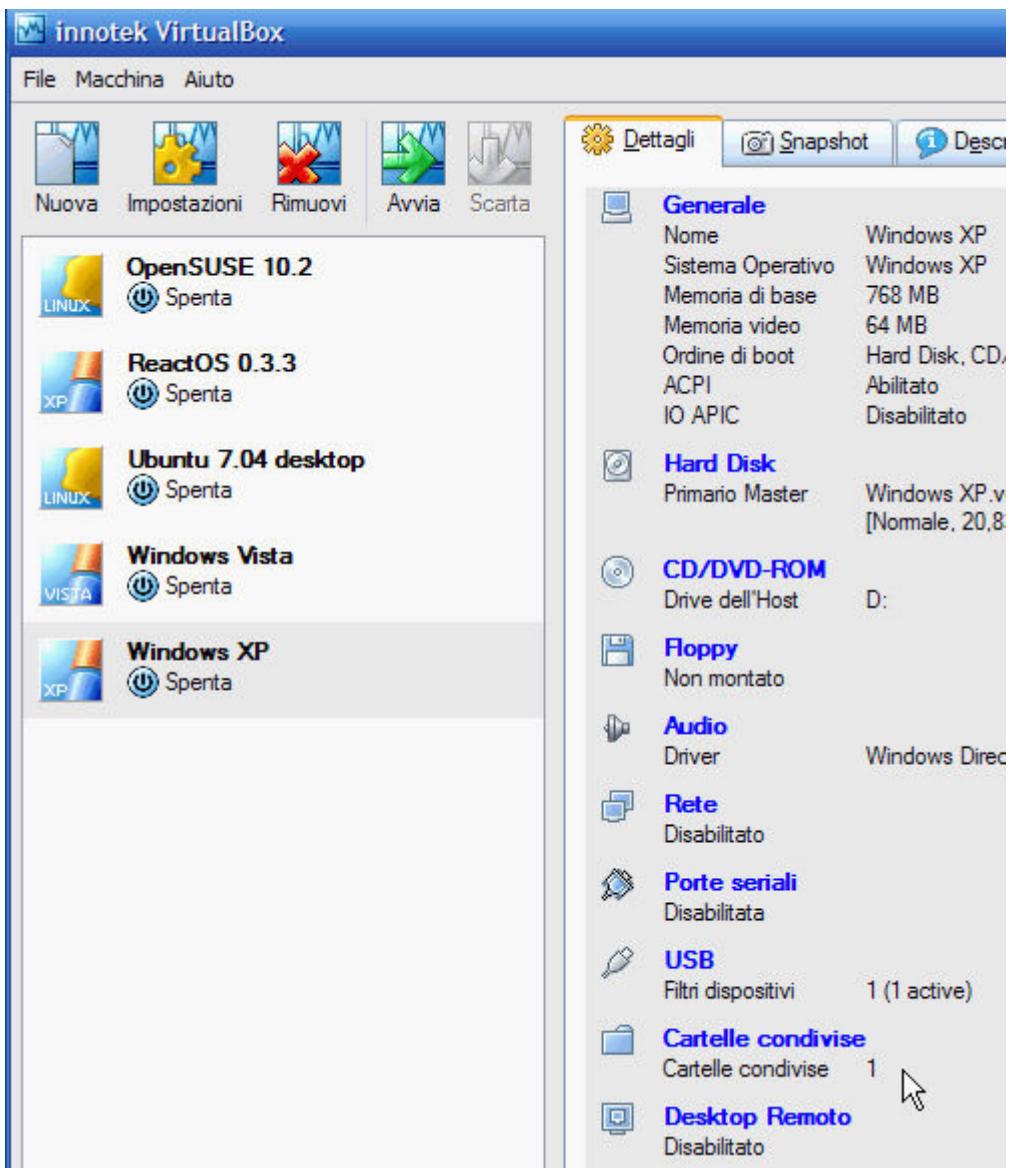


fig. 9

Nel guest avviate l'Esplora risorse e date un'occhiata alla rete (**fig. 10**): eccola lì la nostra cartella condivisa, ha lo stesso nome che le abbiamo assegnato in *Nome cartella* di **fig. 7**; all'interno è visibile Prevx CSI.

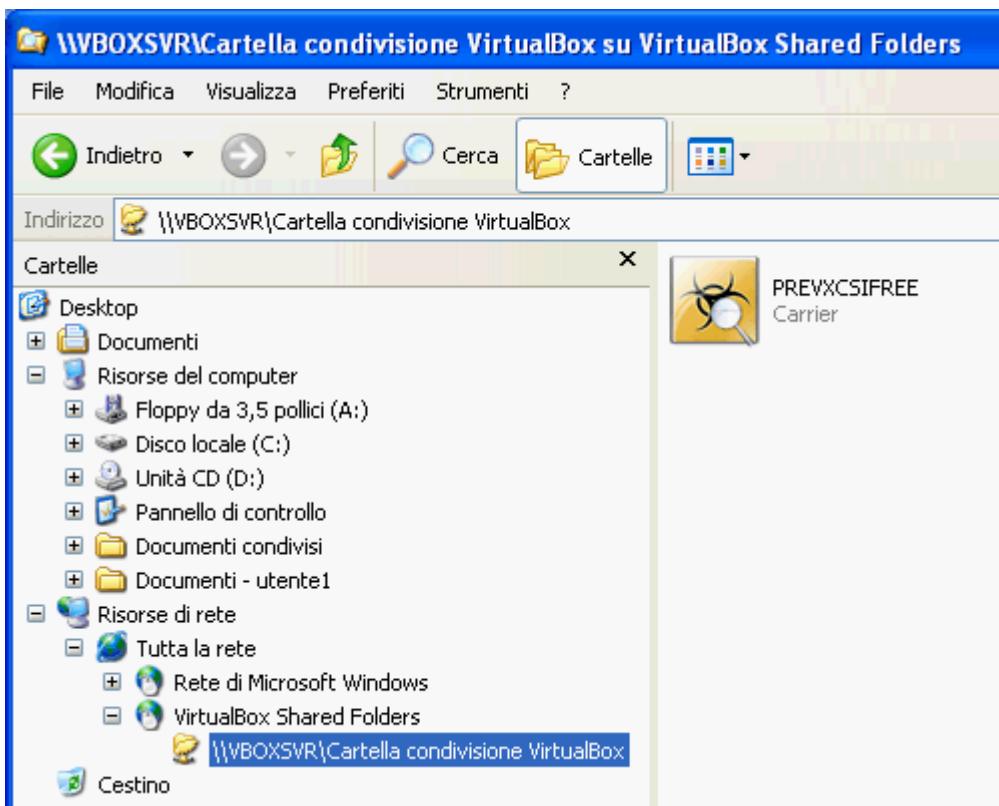


fig. 10

Provate a fare doppio click sul file (**fig. 11**). Il file che, vale la pena ricordarlo, si trova sull'host, è andato in esecuzione come se fosse sul guest: non male, eh?!? ;-)
Esperimento pienamente riuscito! :-D

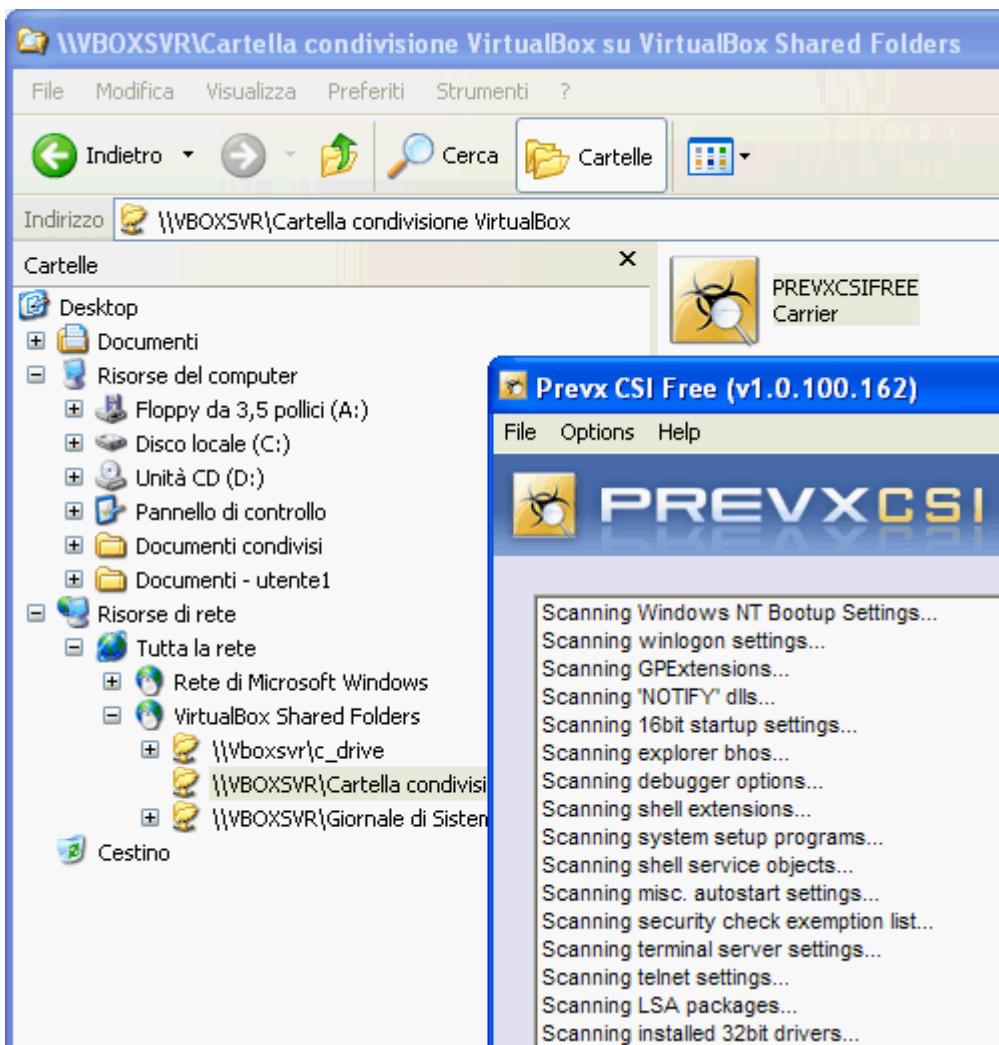


fig. 11

Pensate che sia finita qua? Certo che no! Nel prossimo post vedremo come sia possibile aggiungere cartelle condivise anche a macchina virtuale accesa ed a non limitarci a condividere una singola cartella dell'host: condivideremo infatti tutto l'hard disk! Al prossimo post allora. :-)

@:>

Pubblicato alle 6.12 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione, Windows

lunedì 1 ottobre 2007

VirtualBox: compattare dischi virtuali ad espansione dinamica (1a parte)

Quando create il disco rigido della vostra macchina virtuale, VirtualBox crea un file sull'host con un nome del tipo C:\Documents and Settings\VostroNomeUtente\VirtualBox\Machines\NomeMacchinaVirtuale.vdi

Su di esso viene salvato tutto il contenuto del disco rigido della macchina virtuale; una cosa del genere l'abbiamo già vista nei post relativi all'installazione di Linux direttamente da Windows.

Oltre a supportare con alcune restrizioni il formato aperto VMDK, VirtualBox utilizza soprattutto il formato proprietario VDI, tramite cui può creare 2 tipi di dischi rigidi virtuali:

- a dimensione fissa;
- ad espansione dinamica.

Rivediamo rapidamente cosa sono: supponendo di aver creato un disco rigido virtuale di 10 GB, nel primo caso sul vostro host viene creato subito un file della stessa dimensione, 10 GB. Non è la cosa migliore quando abbiamo poco spazio perché non abbiamo ancora installato nulla e già abbiamo tanto spazio occupato; in compenso le prestazioni sono migliori perché non si perde tempo ad allocare ed eventualmente modificare le dimensioni del file vdi, lo si fa una volta per tutte all'inizio.

Nel secondo caso invece, nel sistema host viene creato un file che inizialmente è di piccole dimensioni, pochi KB, e che cresce man mano serve ulteriore spazio durante l'installazione del sistema operativo, degli applicativi, o comunque durante il normale uso del sistema virtualizzato.

Lo spazio disco sull'host in questo modo non viene sprecato, ne viene sempre allocato esattamente quanto ne serve in quel momento al disco rigido guest, ma le prestazioni sono, sia pur di poco, inferiori rispetto ai dischi rigidi virtuali a dimensione fissa proprio per via di questa ulteriore attività di allocazione al volo ogni volta che serve più spazio al sistema guest.

Questa soluzione però non è tanto dinamica quanto si potrebbe pensare: quando cancelliamo dati dal disco rigido virtuale, il corrispondente file vdi sulla macchina host non si restringe, ma rimane della dimensione massima sino a quel momento raggiunta.

Facciamo un esempio per capirci: dopo avere installato un guest Windows (ma la cosa è analoga con guest Linux) con un Office e gli altri programmi che ci possono servire, dalle proprietà del disco rigido nel sistema guest ci accorgiamo che l'occupazione di spazio è 3 GB: il file vdi sull'host sarà all'incirca della stessa grandezza. Se scaricate 4 GB di file da Internet, o li copiate dall'host tramite la condivisione (un file ad esempio di 4 GB potrebbe essere la iso di una distribuzione Linux), nel disco rigido virtuale vedrete salire l'occupazione dello spazio a 7 GB, e così il file vdi.

Ora viene il bello (meglio dire, il brutto): se cancellate questo file e vuotate il cestino nel guest, l'occupazione di spazio nel guest torna a 3 GB, ma il file vdi sull'host rimane di 7 GB!

Per risolvere questo problema è necessario "compattare" il file vdi tramite il comando modifyvdi, disponibile solamente dalla console a linea di comando attraverso il gestore di comandi VboxManage di VirtualBox stesso.

Con il contributo di Michele Benvegnù, che già qui aveva raccolto un pò di idee e soluzioni, nella prossima parte del post vedremo come affrontare l'argomento.

@:\>

Pubblicato alle 18.14 Argomenti: guida, Linux, tutorial, Virtualbox, virtualizzazione, Windows

mercoledì 3 ottobre 2007

VirtualBox: cartelle condivise in guest Windows (2a parte)

Nel post "VirtualBox: cartelle condivise in guest Windows (1a parte)" abbiamo visto come era possibile aggiungere cartelle condivise alla macchina virtuale spenta. In questo post invece vedremo come farlo quando la macchina virtuale è già accesa, e come nulla ci obblighi a condividere singole cartelle: condivideremo infatti l'intero hard disk dell'host! Cominciamo.

Avviate la macchina virtuale Windows e cliccate sul menù Dispositivi --> Cartelle condivise... della finestra di VirtualBox (fig. 12).

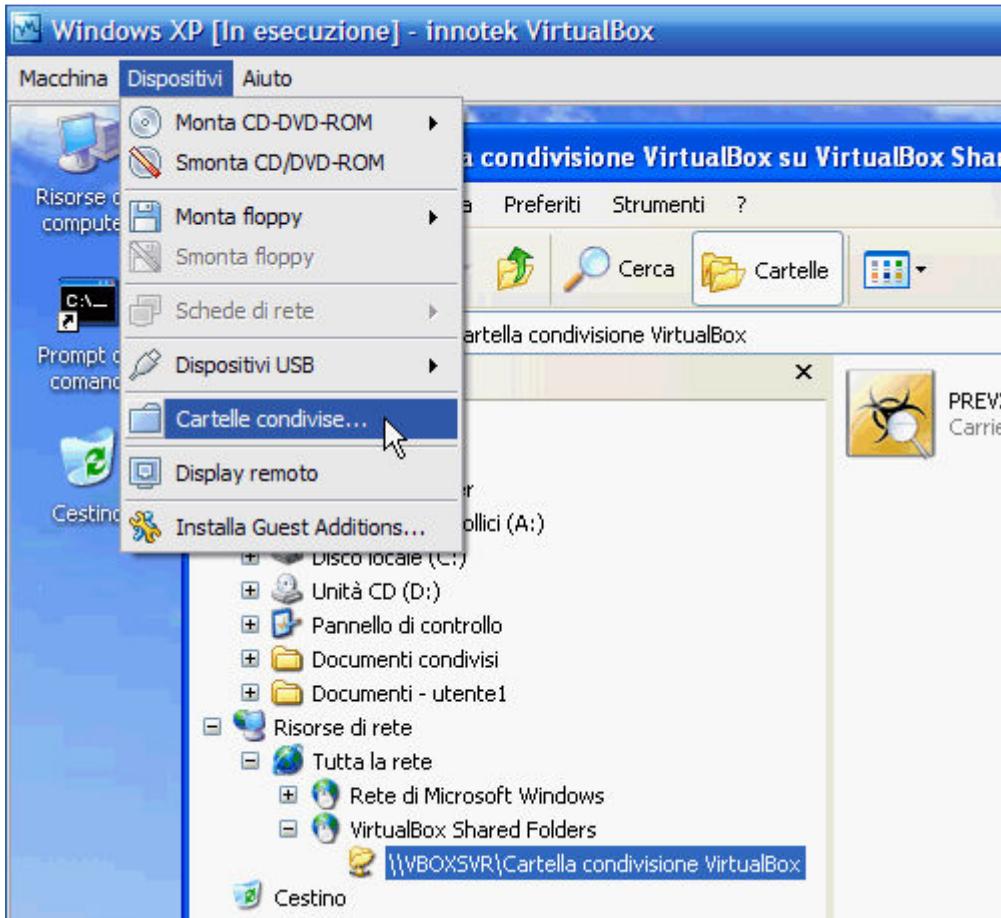


fig. 12

Si aprirà la finestra di dialogo di Cartelle condivise, cliccate sul pulsante indicato dal mouse per aggiungere una nuova cartella (fig. 13).

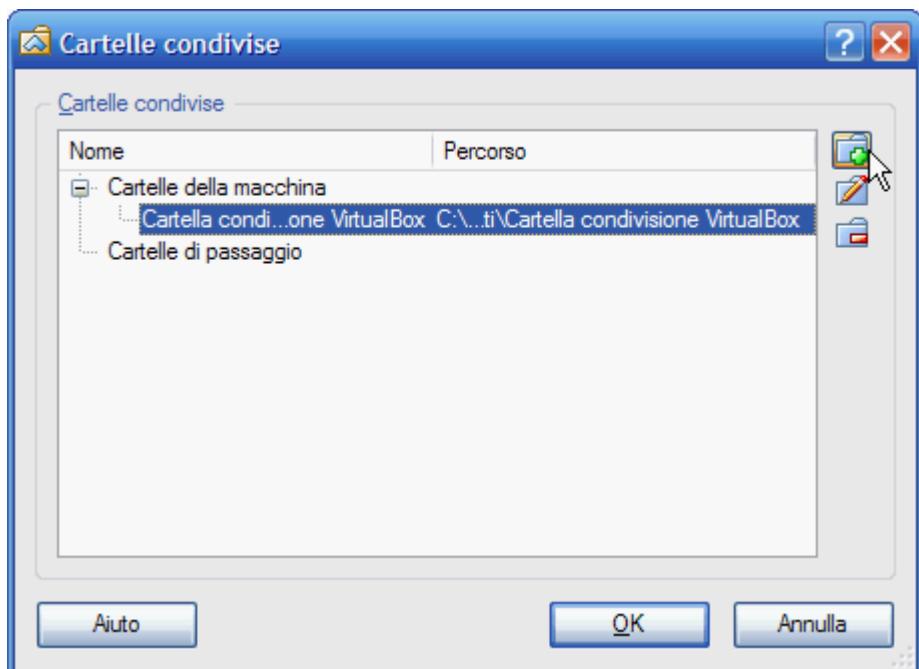


fig. 13

Dalla finestra Aggiungi condivisione che si aprirà, cliccate sul pulsante indicato dal mouse per trovare la posizione sull'host della cartella da condividere (fig. 14).

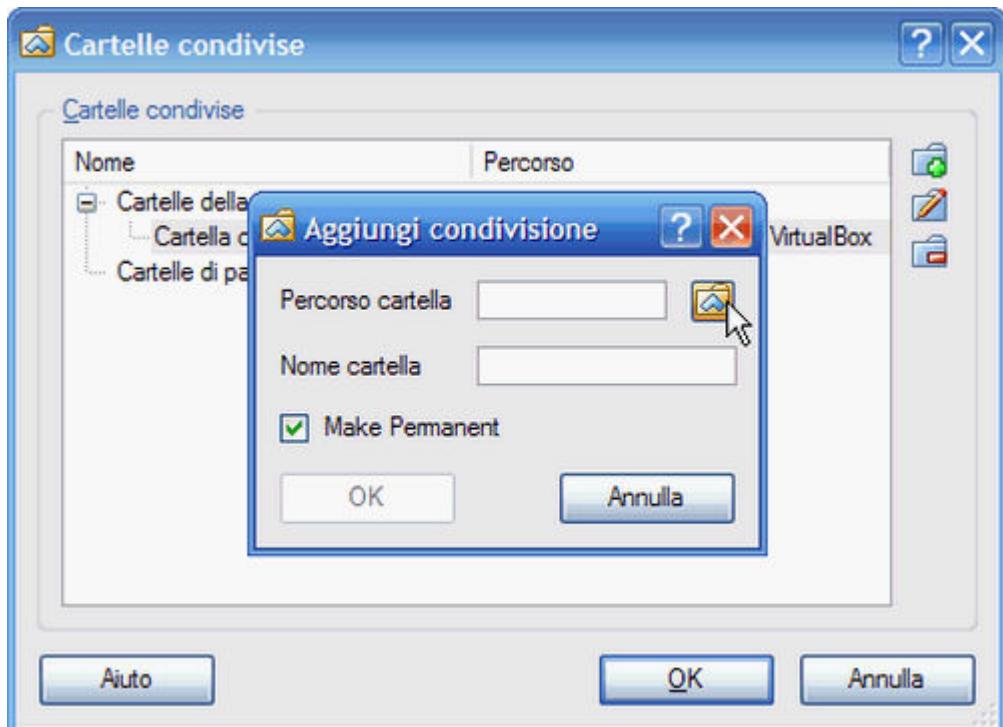


fig. 14

Indicate in Nome cartella (fig. 15), il nome con cui volete sia vista nel guest la cartella condivisa. Notate la spunta nel checkbox di Make Permanent (che strano, deve essere sfuggita la traduzione di questa voce), sta ad indicare che la cartella verrà condivisa in maniera permanente: tra un'accensione e l'altra della macchina virtuale, la condivisione della cartella non andrà persa.

Se invece vi interessa condividere la cartella solo per questa sessione di lavoro, togliete il segno di spunta. La cartella sarà una cartella transiente, ossia allo spegnimento della macchina la condivisione andrà persa. Quando avete finito, cliccate su OK per proseguire.

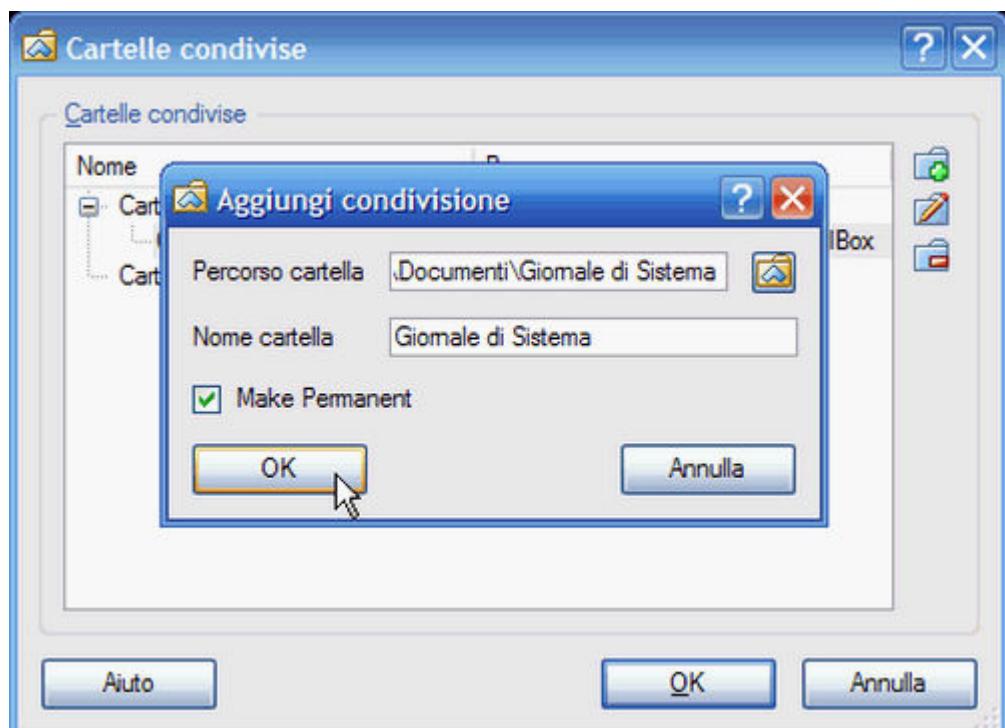


fig. 15

La cartella apparirà nell'elenco di quelle condivise (fig. 16), cliccate su OK per terminare.

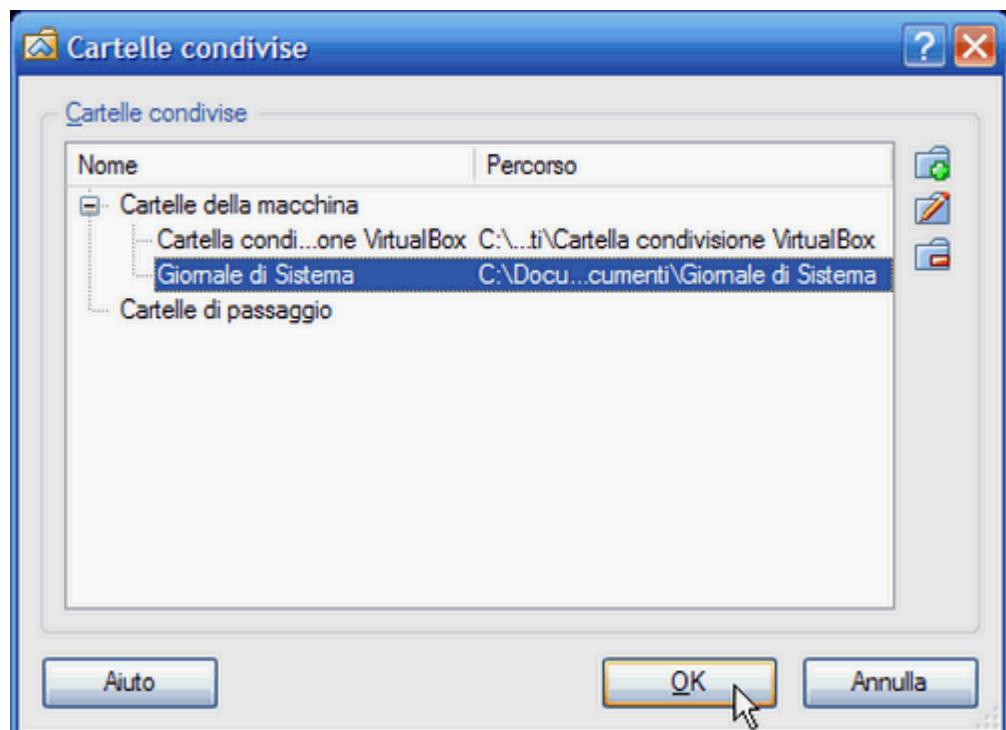


fig. 16

Tornate in Risorse del computer e premete F5 se occorre: comparirà la cartella appena condivisa fra le Risorse di rete (fig. 17). Ricordate però: compare fra le risorse disponibili in rete, ma non state utilizzando davvero la rete che è disattivata (fig. 3 e relativo testo nella prima parte del post).

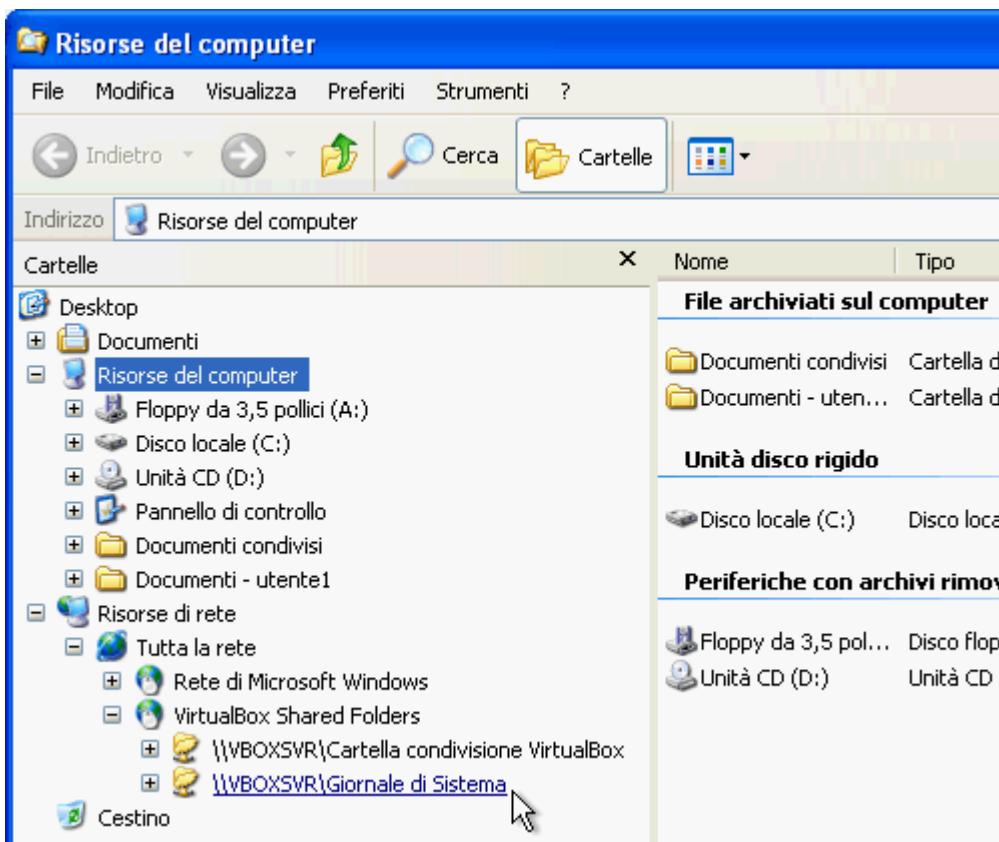


fig. 17

Come accennato all'inizio del post, vediamo infine come poter condividere l'intero disco rigido dell'host.

Aggiungete una nuova cartella da condividere come in fig. 13, ma al momento di sceglierla, scegliete l'intero disco rigido dell'host come in fig. 18 dove è già selezionato.

Notate un'altra cosa di fig. 18: il pulsante Crea nuova cartella indicato dal mouse. Grazie ad esso potrete creare una nuova cartella direttamente sull'host. Quando avete finito, cliccate su OK per continuare.

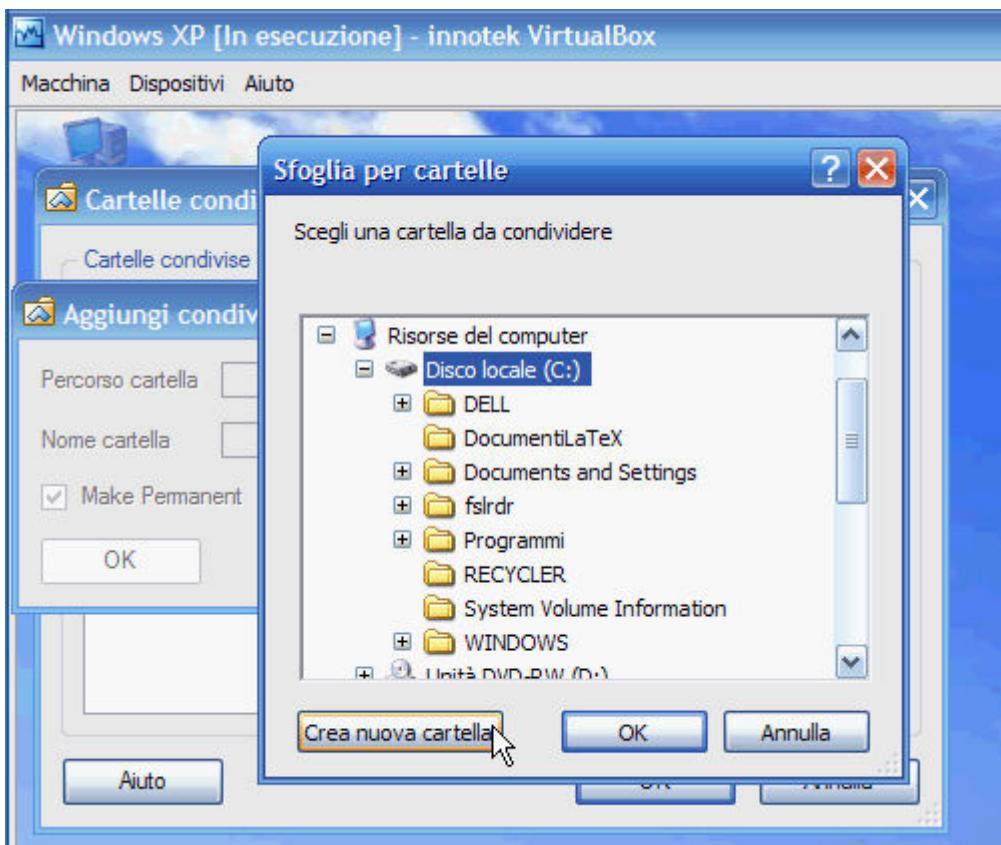


fig. 18

Come potete verificare in fig. 19, state per condividere l'intero disco rigido dell'host e se volete che questa condivisione sia permanente, lasciate selezionato Make Permanent, poi cliccate 2 volte di seguito sui 2 pulsanti OK che compaiono per terminare.

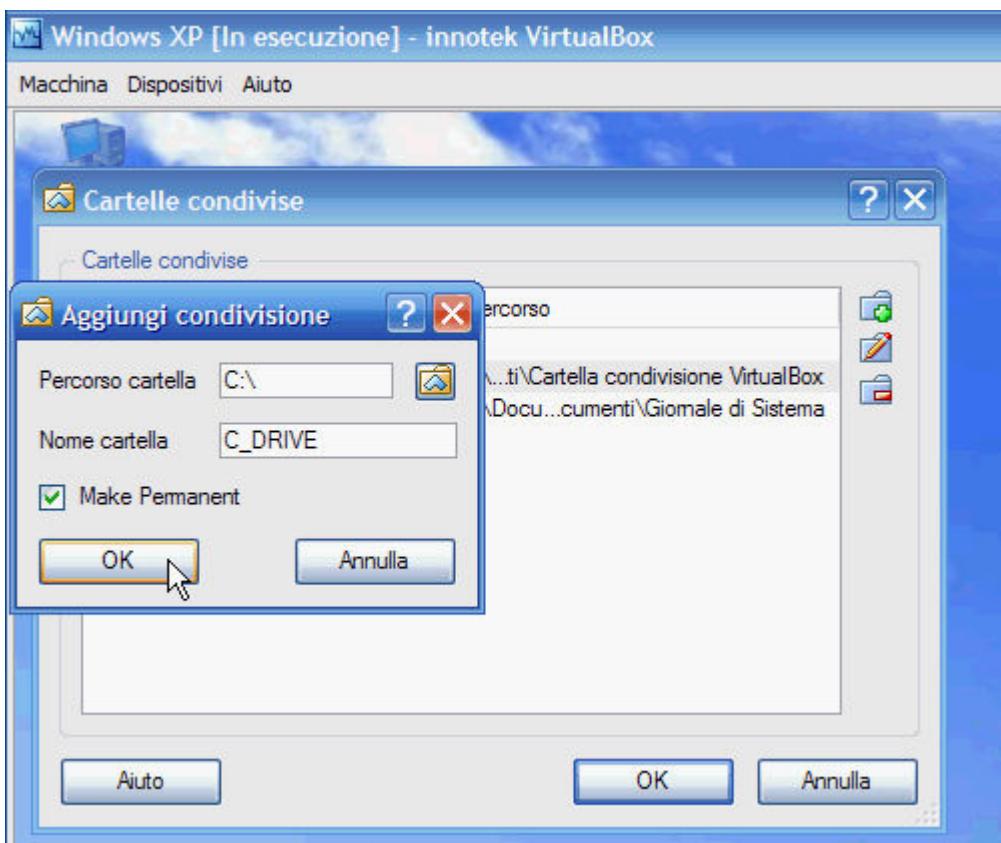


fig. 19

Ed ecco il risultato finale: l'host Windows che condivide l'intero disco rigido con il guest anch'esso Windows.

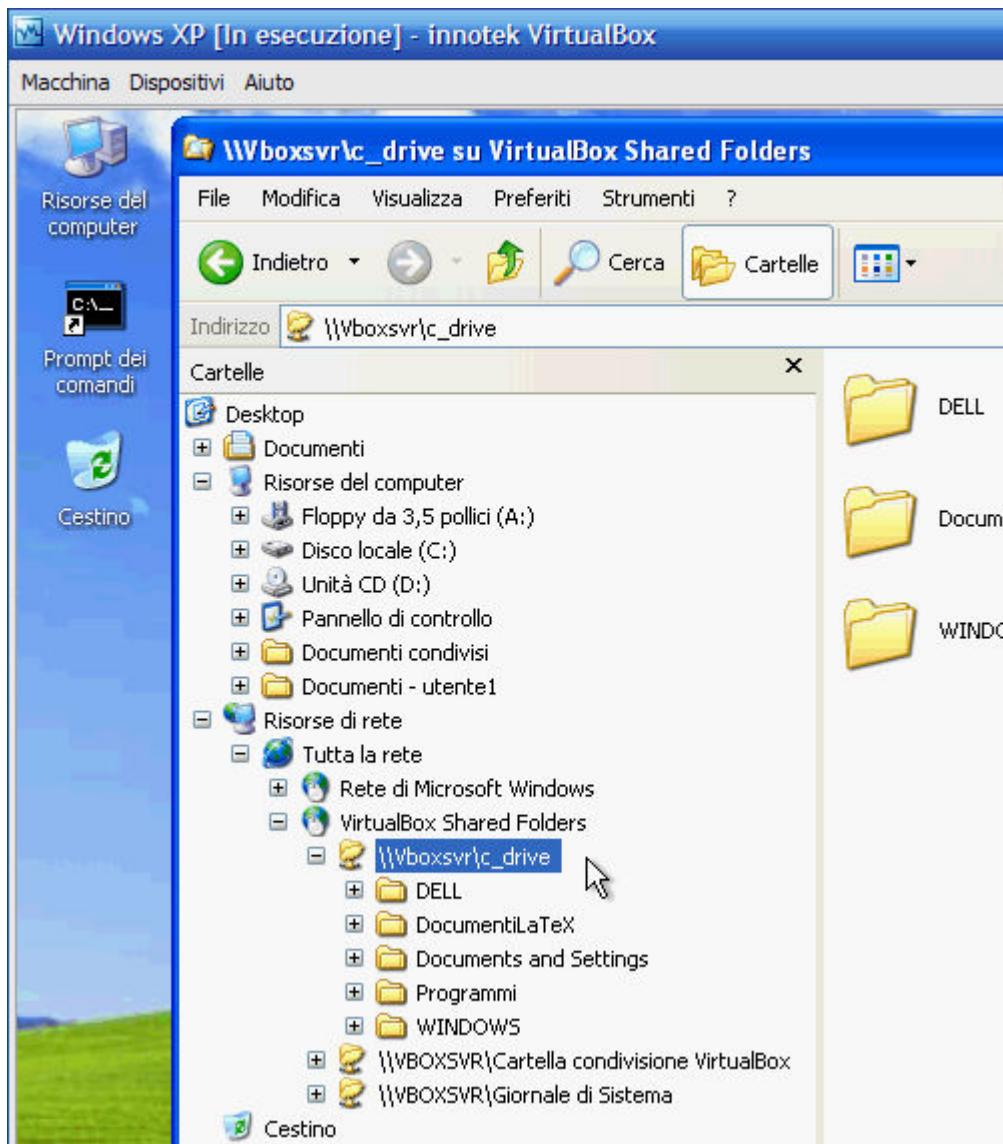


fig. 20

Abbiamo terminato per quanto riguarda la condivisione di cartelle con guest Windows. Nel prossimo post sull'argomento vedremo come condividere cartelle con guest Linux.

@:>

Pubblicato alle 9.48 Argomenti: guida, software, tutorial, Virtualbox, virtualizzazione, Windows

mercoledì 3 ottobre 2007

VirtualBox: togliere/mettere il Blocco Maiuscole (Caps Lock)

Vi è mai capitato di avere le maiuscole inserite sull'host con *blocco maiuscole* (Caps Lock) nell'host e nel momento che passate al guest non riuscite a toglierle? E naturalmente il viceversa: non le avete nell'host e quando passate alla finestra del guest volete inserire il blocco maiuscole. Semplicemente non c'è verso di farlo.

Piccolo trucco per superare questa idiosincrasia di VirtualBox: impostate preventivamente nell'host la presenza o meno del blocco maiuscole. E se proprio ve ne siete dimenticati, tornate su una finestra qualunque dell'host, inserite/disinserite il Blocco Maiuscole e tornate nella finestra del guest, vi porterete dietro le impostazioni fatte nell'host.

@:`>

Pubblicato alle 18.05 Argomenti: Linux, suggerimenti, Virtualbox, Windows

giovedì 4 ottobre 2007

VirtualBox: compattare dischi virtuali ad espansione dinamica (2 parte)

Nel precedente post sull'argomento, abbiamo inquadrato il problema: quando liberiamo spazio sul disco rigido virtuale, il corrispondente file vdi che lo contiene sull'host non si ridimensiona di conseguenza: occorre forzarne il ridimensionamento con un apposito comando, *modifyvdi*, disponibile solamente dalla console a linea di comando dell'host attraverso il gestore di comandi di VirtualBox, *VboxManage*.

Purtroppo però ciò non è sufficiente, occorre che sia soddisfatto un pre-requisito: lo spazio libero sul guest deve essere marcato a zero (il valore, non la cifra) prima che il comando possa agire su di esso compattandolo. Questo perché quando cancellate un file, questi non viene realmente cancellato: semplicemente lo spazio che occupa viene dichiarato libero in una apposita tabella, in questo modo lo si può riutilizzare per scrivervi nuovi file.

I settori del disco rigido quindi, contengono ancora i dati, ed è per questo che il comando *modifyvdi* non riesce a fare il suo lavoro: può infatti agire solo su spazio realmente vuoto e non su quello dichiarato semplicemente come tale. Occorre un programma che svuoti per davvero lo spazio dichiarato libero. Innotek a dire il vero qui è stata un po' latitante, tant'è che VirtualBox a differenza di prodotti analoghi, non ha uno strumento per marcare a zero i settori liberi del disco rigido virtuale.

In questo post, realizzato con il contributo di Michele Benvegnù, vediamo la soluzione a questo problema e la successiva compattazione del file vdi per guest Windows; nel prossimo post sull'argomento vedremo il tutto per guest Linux.

Si chiama *SDelete* ed è un programma di Mark Russinovich di Sysinternals. La sua funzione (tra le altre) è proprio quella di scrivere a zero i settori non utilizzati di un qualunque disco, compresi quelli virtuali di VirtualBox. Scaricate lo zip dall'indirizzo web <http://www.microsoft.com/technet/sysinternals/FileAndDisk/SDelete.mspx> o direttamente dal guest, oppure dall'host in una cartella condivisa, così da renderla immediatamente disponibile anche al guest.

Scompattate il file nel guest, è lì che lo dovremo usare: quello che ci interessa è l'eseguibile *sdelete.exe*, un programma utilizzabile solo da linea di comando in quanto privo di front-end grafico.

Vediamo gli screenshot della situazione iniziale così da poter poi apprezzare anche visivamente le differenze. In fig. 1 vedete la situazione dello spazio sul guest.



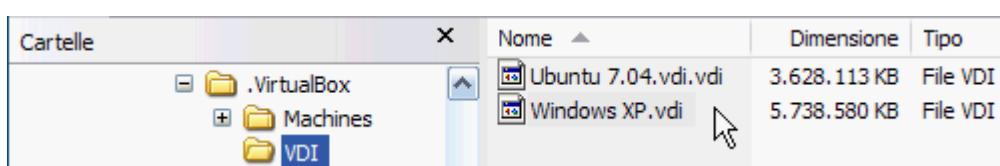
Con un disco rigido di capacità massima di 20 GB, vi ho copiato sopra 4 GB di file tramite condivisione cartelle: ho copiato 6 volte con 6 nomi diversi la iso d'installazione di Ubuntu:

700 MB x 6 = ~ 4 GB

che sommati ai circa 1-1,2 GB dell'installazione di Windows, fanno circa 5,2 GB occupati di spazio disco occupato. Ed infatti lo spazio libero sul disco del guest è:

20 GB - 5,2 GB = 14,8 GB

Il file vdi sull'host (fig. 2) è appunto di circa 5,8 GB, quanto serve a contenere tutto il disco rigido virtuale.



Ora elimino (non cestino, ma elimino definitivamente, anche dal cestino) i sei file di Ubuntu nel guest, in **fig. 3** la nuova situazione dello spazio sul guest.



fig. 3

Sono stati liberati i 4 GB e passa, quindi:

$$14,8 \text{ GB liberi} + \text{oltre 4 GB liberati} = 18,9 \text{ GB liberi}$$

Andate a controllare la situazione sull'host: come ci aspettavamo, non è cambiato nulla, il file vdi non si è ridimensionato di conseguenza e invece di scendere a soli 1-1,2 GB è rimasto di 5 e passa giga.

Ok, è il momento di far intervenire SDelete.

Aprite la finestra del DOS con: *pulsante start --> Esegui --> scrivete cmd e date invio*. Nella finestra DOS che si apre, date questo comando doppi apici compresi:

"C:\Documents and Settings\utente1\Desktop\`sdelete.exe" -c C:

Nel frattempo il comando porta a termine il suo compito (**fig. 4**), potranno servire anche una decina di minuti a seconda del sistema che avete, vi spiego la sua sintassi:

- C:\Documents and Settings\utente1\Desktop\`sdelete.exe" è il percorso dove si trova SDelete, l'avevo messo sul Desktop del guest, sostituitelo con il percorso della cartella in cui avete messo SDelete sul vostro sistema guest;
- i doppi apici servono a causa dello spazio in Documents and Settings, se nel percorso della vostra cartella non vi sono spazi, potete ometterli;
- c è un flag che dice a SDelete di mettere tutti zero nei settori liberi del disco rigido virtuale;
- C: è il disco su cui agire.

```
C:\Documents and Settings\utente1>"C:\Documents and Settings\utente1\Desktop\sdelete.exe" help
SDelete - Secure Delete v1.51
Copyright (C) 1999-2005 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Documents and Settings\utente1>"C:\Documents and Settings\utente1\Desktop\sdelete.exe" -c C:
SDelete - Secure Delete v1.51
Copyright (C) 1999-2005 Mark Russinovich
Sysinternals - www.sysinternals.com

SDelete is set for 1 pass.
Cleaning free space on C:: 2%
```

fig. 4

Se nel frattempo SDelete ha terminato, all'interno della finestra DOS sarà comparso un messaggio del tipo: Free space cleaned on C:. Spegnete la macchina guest.

E' arrivato finalmente il momento di compattare il file vdi sull'host: ora è possibile perché la precondizione che lo spazio libero sul disco rigido virtuale fosse marcata a zero è soddisfatta. Seguendo gli stessi passi fatti nella macchina guest, aprite una finestra DOS anche sull'host. Al suo interno date questo comando, doppi apici compresi:

"C:\Programmi\innotek VirtualBox\VBoxManage" modifyvdi "nomefile".vdi compact

Mentre il comando fa il suo lavoro vi spiego anche qui la sintassi:

- intanto i doppi apici anche qui servono ove ci siano spazi vuoti. Normalmente la linea di comando interpreta gli spazi vuoti come la fine di qualcosa; se non mettiamo i doppi apici, C:\Programmi\innotek VirtualBox\VBoxManage non viene interpretata come una cosa unica ma viene presa solo la prima parte, cioè C:\Programmi\innotek e li si ferma dando errore.
- stessa cosa con il nome del file vdi (e mi raccomando, **solo il nome**, l'estensione .vdi NON deve essere compresa nei doppi apici). I doppi apici vanno messi solo se il nome del file vdi presenta spazi al suo interno;
- VBoxManage è il comando che lancia il manager a linea di comando di VirtualBox, serve a dare da linea di comando tutti i comandi che normalmente date dall'interfaccia grafica, e altri che l'interfaccia grafica non implementa ancora ma sono comunque disponibili da linea di comando;
- modifyvdi è il comando impartito a VBoxManage, gli dice che volete modificare il file nomefile.vdi;
- compact dice che la modifica che volete effettuare sul file vdi è la sua compattazione.

Nel frattempo la compattazione dovrebbe essere finita anche sul vostro sistema, in **fig. 5** il risultato: come potete vedere, il file vdi è tornato all'incirca alle sue dimensioni iniziali.

Nome	Dimensione	Tipo
Ubuntu 7.04.vdi.vdi	3.628.113 KB	File VDI
Windows XP.vdi	1.442.900 KB	File VDI

fig 5

E se il sistema guest è Linux? I passi da fare sono i medesimi, li vedremo nel prossimo post sull'argomento.

@:)> ... con il contributo di Michele Benvegnù (www.benvegnu.it)

Pubblicato alle 9.20 Argomenti: guida, Linux, tutorial, Virtualbox, virtualizzazione, Windows

giovedì 11 ottobre 2007

VirtualBox: cartelle condivise in guest Linux (1a parte)

Nella precedente serie di post sull'argomento Cartelle condivise, vi avevo mostrato la condivisione di cartelle fra host e un guest Windows. Questa volta vi mostro come attivare le cartelle condivise con un guest Linux; non dimenticate però: potrete attivare le cartelle condivise solo dopo aver installato le VirtualBox Guest Additions. Le immagini di questo post fanno inoltre riferimento ad una distribuzione Ubuntu, ma tranquilli: le operazioni sono indipendenti dalla distribuzione usata.

Ok, iniziamo con il condividere una cartella a macchina virtuale spenta.

Avviate VirtualBox e selezionate la macchina virtuale relativa alla vostra distribuzione (fig. 1): notate dalla scheda dettagli come la *Rete* sia *Disabilitata*. Cliccate sulla voce *Cartelle condivise* per proseguire.

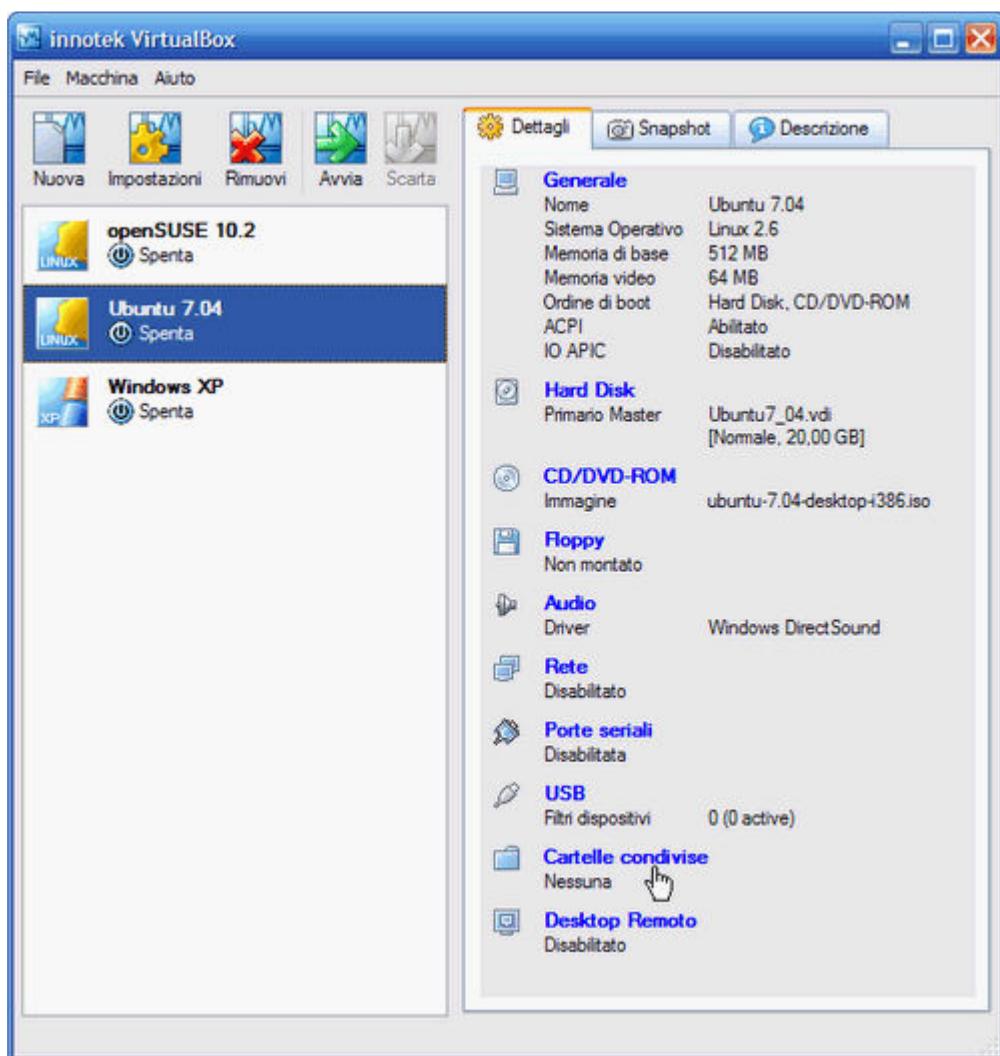


fig. 1

Si aprirà il pannello di gestione delle *Cartelle condivise* (fig. 2). Ancora non ve ne sono, ma la didascalia sopra il pulsante *OK* vi da già preziose informazioni su come condividere le cartelle una volta all'interno del guest.

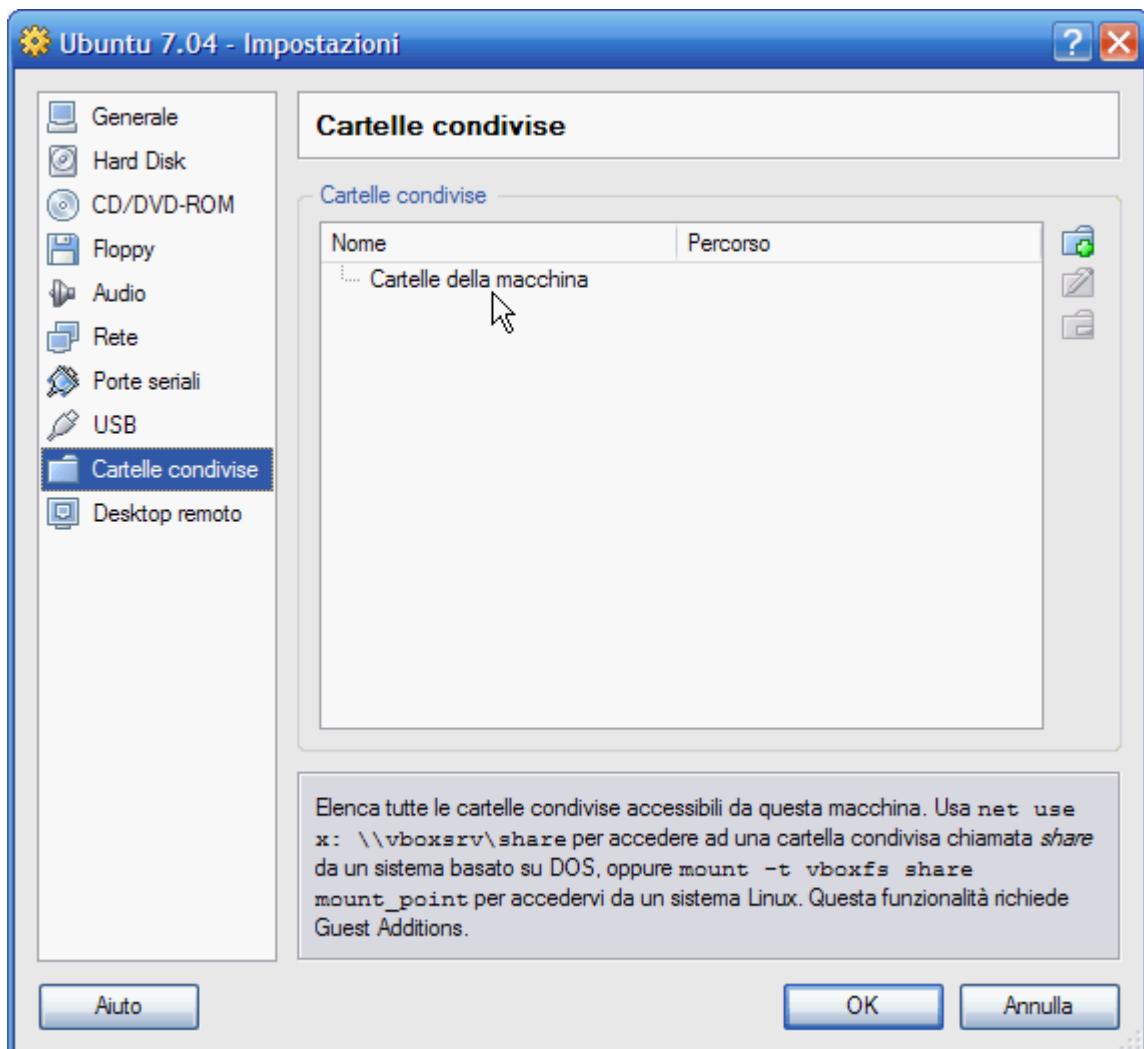


fig. 2

Cliccate sul pulsante di aggiunta di una nuova cartella condivisa, nella posizione indicata dal mouse (fig. 3):

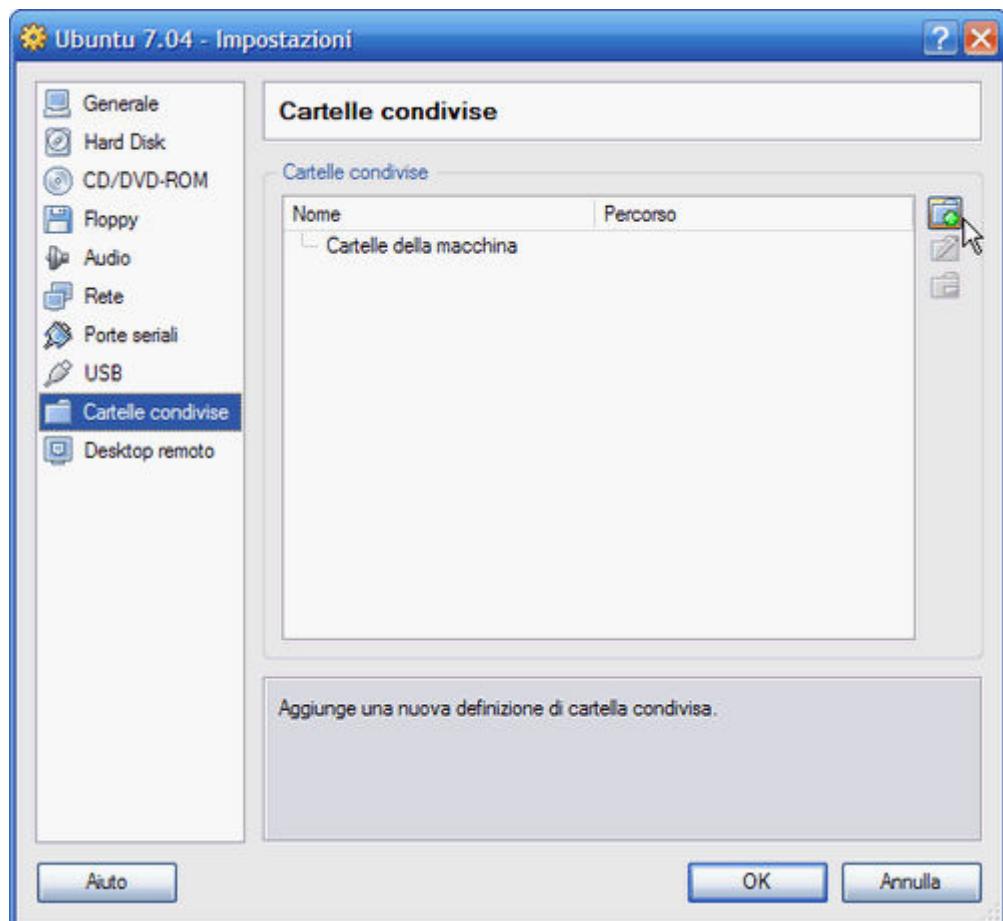


fig. 3

Si aprirà la finestra di *Aggiungi condivisione* (fig. 4), cliccate sul pulsante per trovare la cartella sul disco rigido dell'host, nel punto indicato dal mouse.

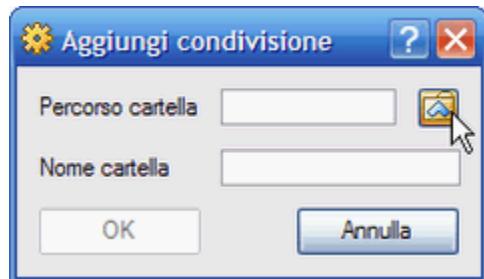


fig. 4

Navigate il vostro disco rigido fino a trovare la cartella che volete condividere (fig. 5) e cliccateci sopra per selezionarla; se la cartella non esiste, potete crearne una al volo usando il pulsante *Crea nuova cartella*. Quando avete selezionato la cartella che vi interessa, cliccate su *OK* per confermare.

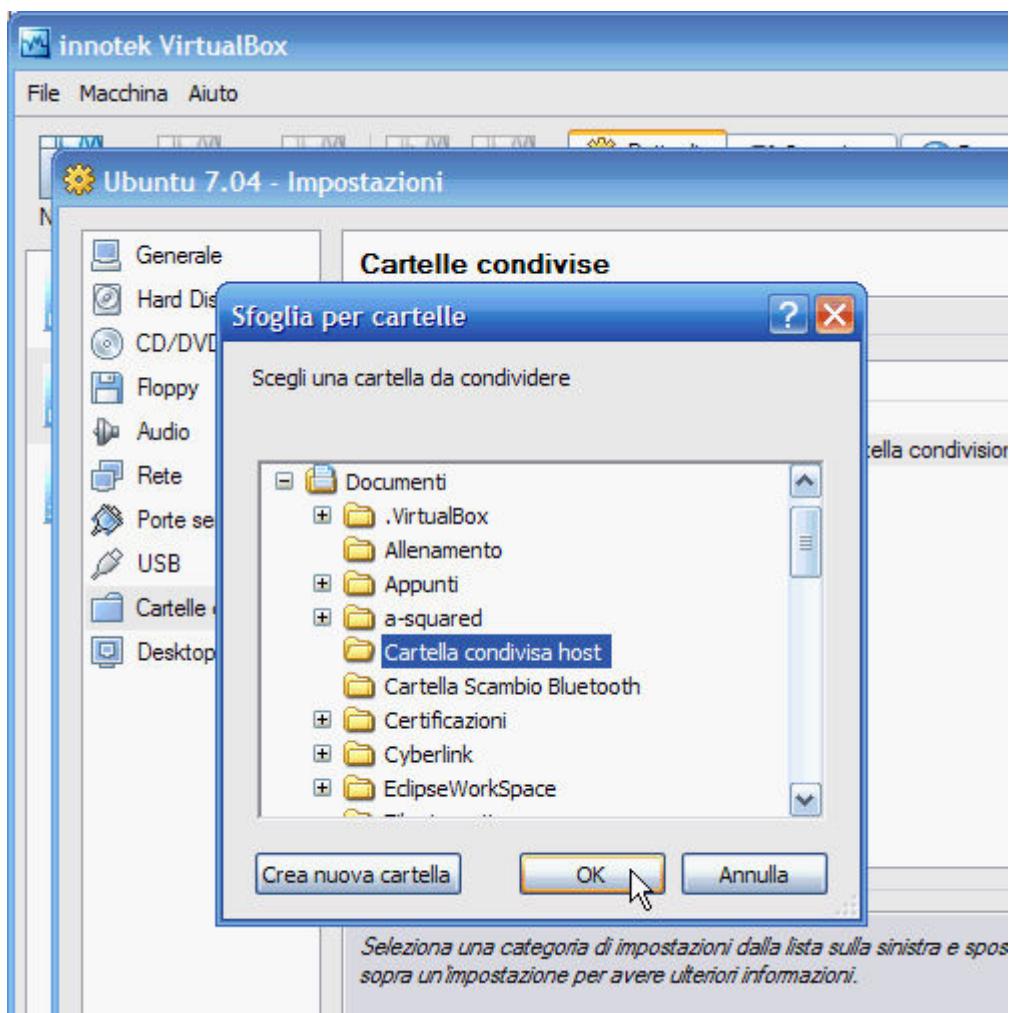


fig. 5

Vi ritrovate nella finestra precedente di *Aggiungi condivisione* che si è trasformata in *Modifica condivisione*. La voce *Nome cartella* fa riferimento al nome con cui la cartella condivisa sarà visibile nel guest e ne viene preimpostato uno, se non vi piace modificalo a vostro piacimento e poi cliccate su *OK* per confermare la scelta.

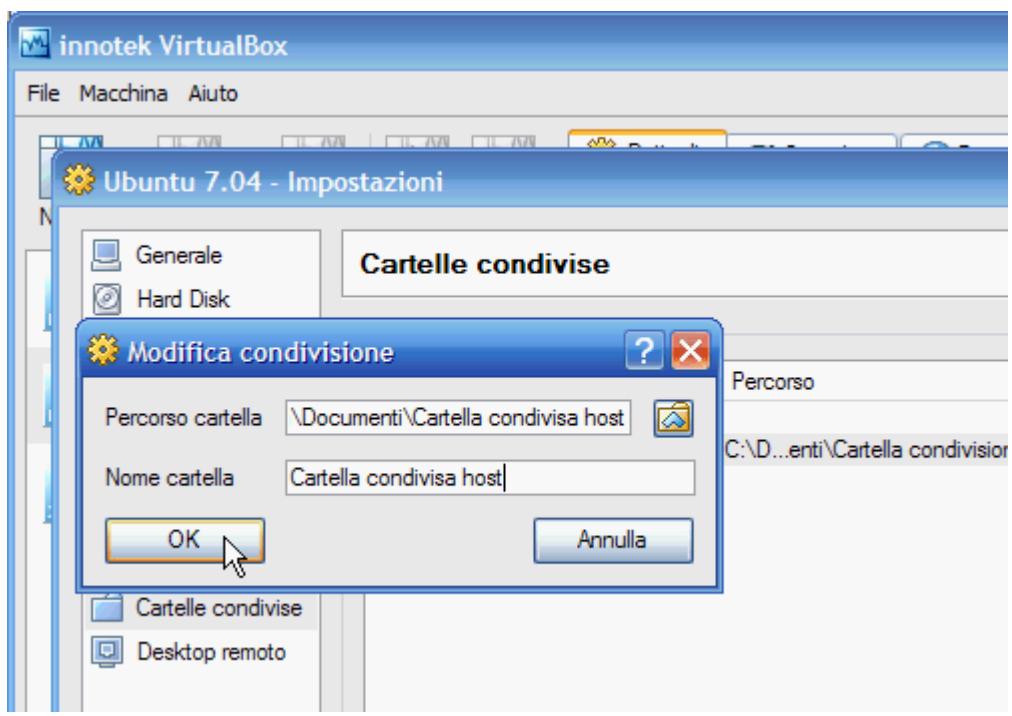


fig. 6

Eccola (**fig. 7**), la cartella sull'host che avete deciso di condividere con il guest, è nell'elenco delle *Cartelle condivise*; con lo stesso procedimento aggiungetene quante volete e cliccate su *OK* per terminare.

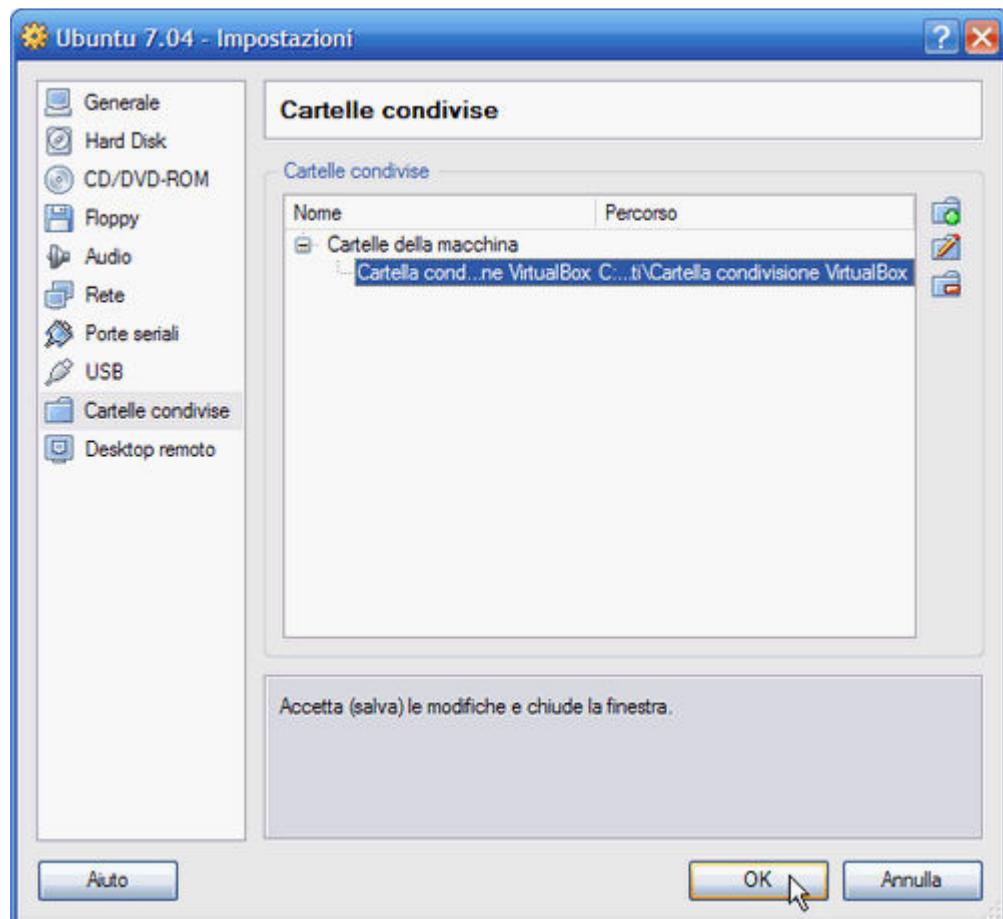


fig. 7

Ok, la condivisione è pronta e VirtualBox vi informa del numero di cartelle dell'host che la macchina virtuale condivide (**fig. 8**).

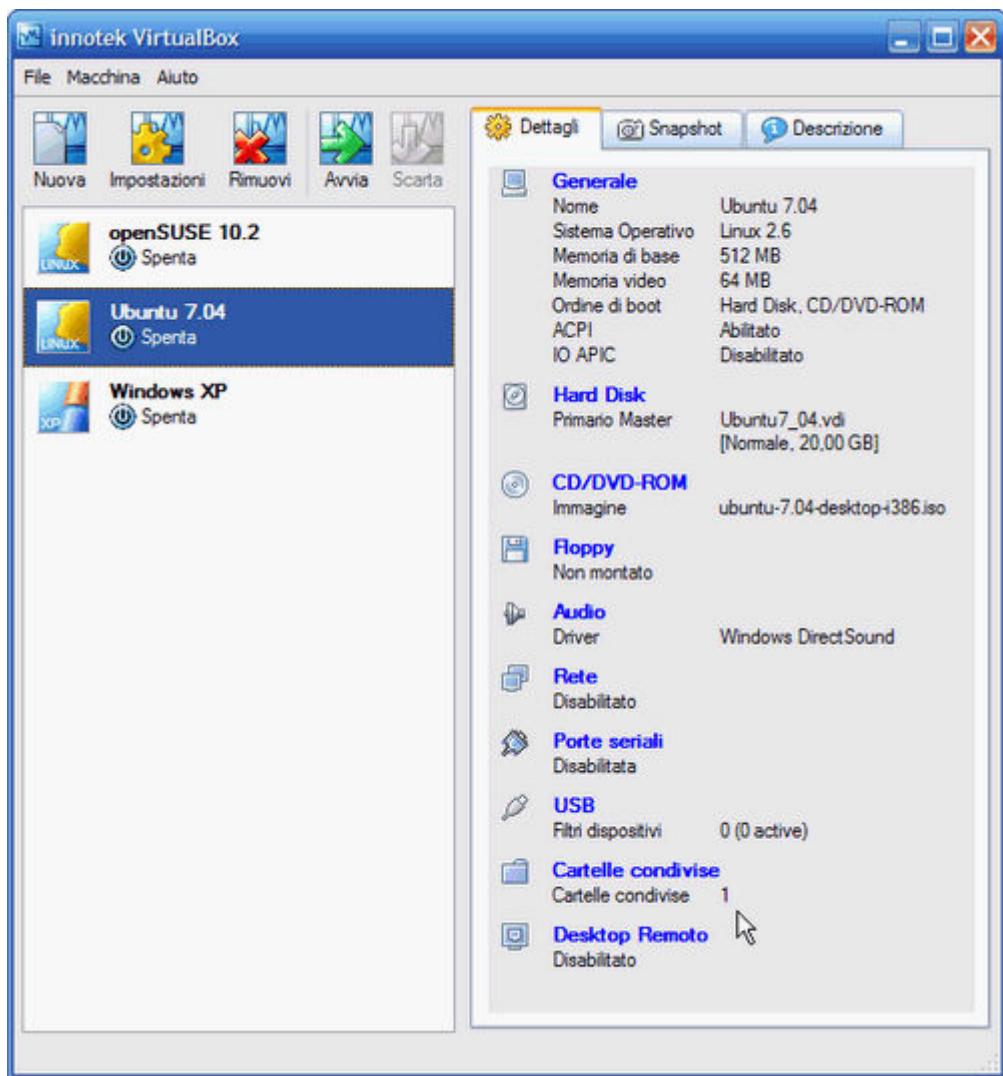


fig. 8

Ora avviate la macchina virtuale.

Avviate una sessione di terminale all'interno della vostra distribuzione Linux (fig. 9): come preannunciato, ho usato una distribuzione Ubuntu, ma i comandi sono identici sia la vostra distribuzione.



fig. 9

Se non vi trovate a vostro agio con i comandi della shell (terminale) e volete capire graficamente cosa sta per accadere, allora apriete il vostro file manager. Puntatelo sulla radice / del filesystem, pressappoco il C:\ di Windows, e da lì entrate nella cartella *mnt*. Questa è la cartella che per convenzione Linux utilizza per montare e rendere disponibili al sistema nuovi dispositivi di memoria quali: Hard disk, memorie USB, floppy... ed è qui che noi monteremo la cartella condivisa dell'host.

Ok, tornate nella shell e date il comando seguente apici singoli compresi (fig. 10):

```
sudo mkdir /mnt/'Cartella condivisa host'
```

Dopo aver dato la password il comando verrà eseguito.

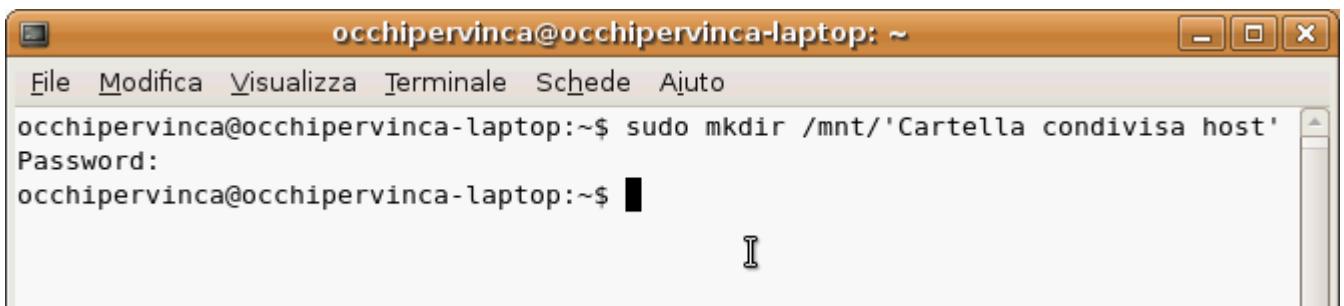


fig. 10

Ma cosa ha eseguito il comando?

- *sudo*: esegui i successivi comandi con le autorizzazioni da amministratore;
- *mkdir*: MaKe DIRectory, crea una directory;
- */mnt/'Cartella condivisa host'*: creala all'interno della directory *mnt* e chiamala *Cartella condivisa host*.

Gli apici singoli servono a far capire alla shell che gli spazi fanno parte del nome della cartella. Ora guardate nel vostro file manager (fig. 11): è comparsa la nuova cartella che farà da punto di mount nel filesystem della vostra distribuzione Linux.

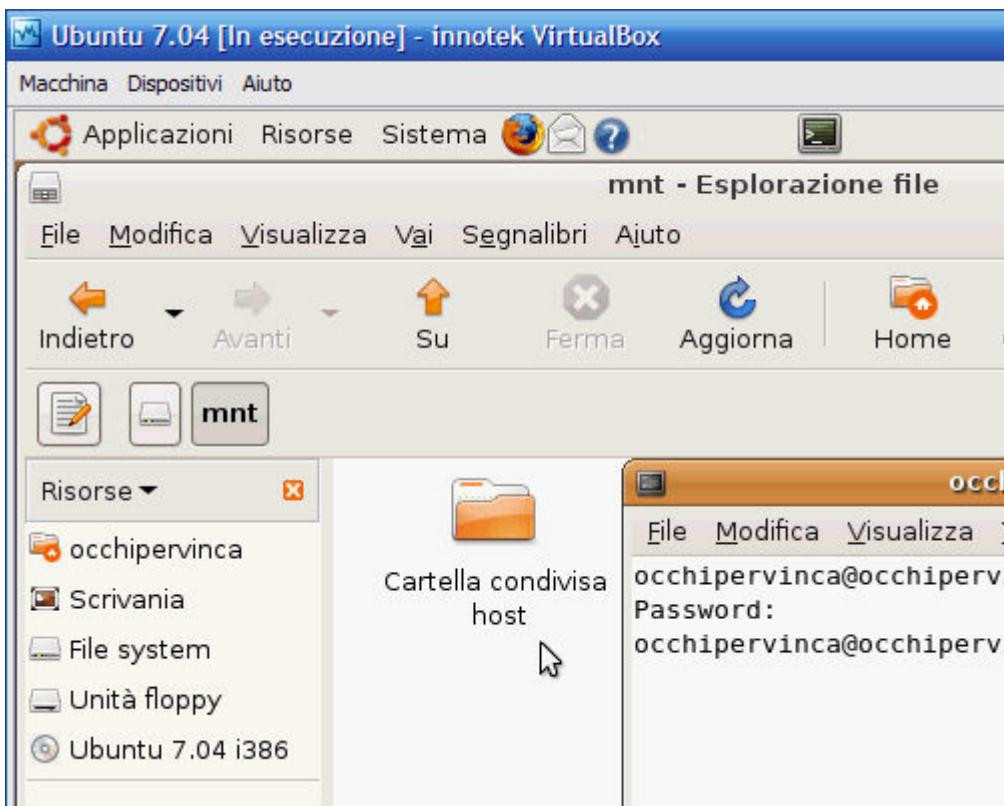


fig. 11

La directory (o cartella, in questi post indico la stessa cosa con nomenclatura Linux o Windows a seconda di dove siamo) è stata creata, ora possiamo montarvi la cartella condivisa. Nella shell date il comando (**fig. 12**):

```
sudo mount -t vboxsf 'Cartella condivisa host' /mnt/'Cartella condivisa host'
```

Se vi chiede la password inseritela e date invio per far eseguire il comando alla shell. Se tutto è andato bene, sembrerà non sia accaduto nulla, semplicemente la shell vi restituirà il prompt, ma...

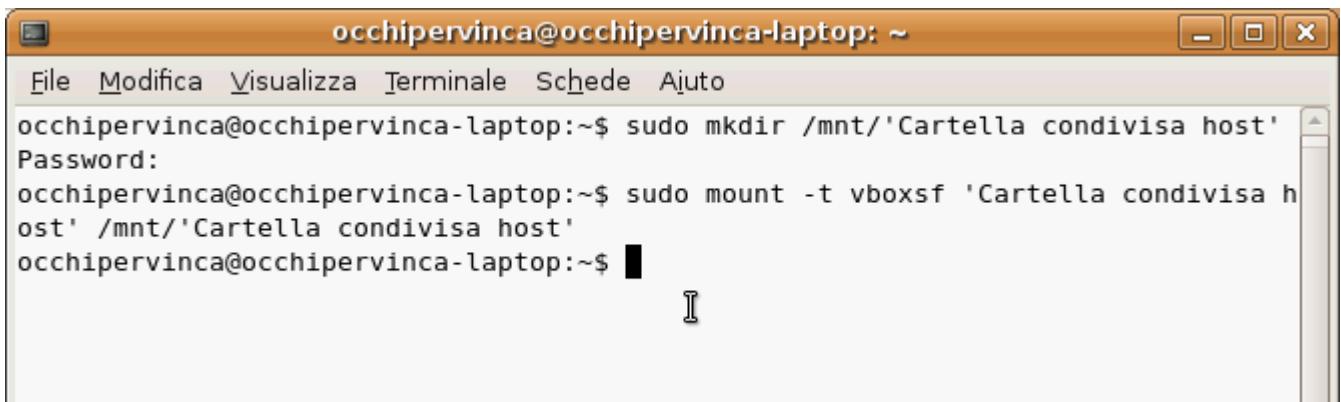


fig. 12

... date un'occhiata cosa trovate all'interno della *Cartella condivisa host*: et voilàaa... Signori e Signore, Ladyes and Gentlemen... il contenuto della cartella host condivisa ora è visibile ed a vostra disposizione nel guest (**fig. 13**)! :-D

Ho lasciato visibile a destra anche una parte della shell da cui abbiamo dati i comandi. Se avete già seguito il post per condividere cartelle con guest Windows, riconoscerete il programma Prevx CSI che avevo usato in quel post per farvi vedere come dalla macchina guest, si poteva avviare un programma che fisicamente era in una cartella sull'host.

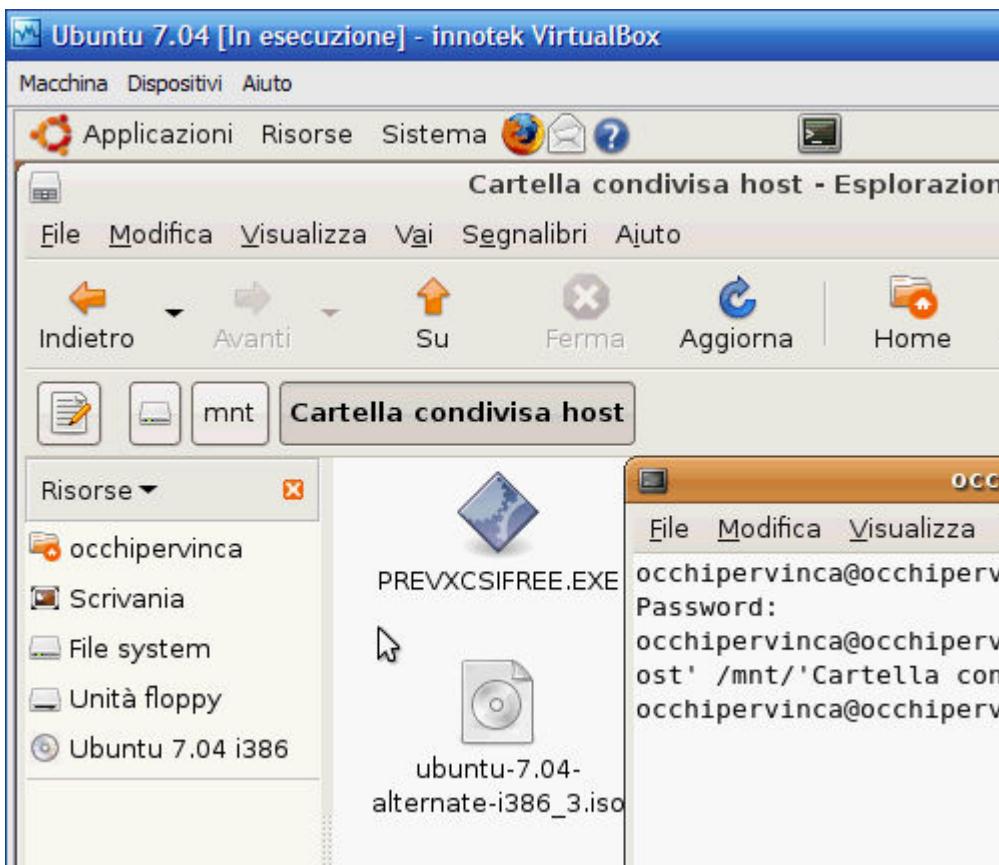


fig. 13

Ricordate infatti una cosa: i file in questo momento fisicamente sono ancora sull'host, è come se si fosse aperta una finestra che vi permette di osservarli dal guest: se volete portarli fisicamente sul guest, basterà che nel file manager li trasciniate in un'altra cartella del guest stesso, verranno copiati lì.

Beh, mi pare che per questa volta possa bastare, mi fermo qua. Non mancate però alla 2a parte del post, quando condivideremo l'intero disco rigido dell'host facendolo direttamente dalla macchina guest in funzione! ;-D

@:>

Pubblicato alle 15.30 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

2 commenti:

Michele Benvegnù ha detto...

Fatto, però se chiudo Ubuntu e lo faccio ripartire la condivisione non c'è più. Non dovrò dare il comando `sudo mount -t vboxsf 'Cartella condivisa host' /mnt/'Cartella condivisa host'` tutte le volte? Inoltre, perchè la cartella creata con mkdir non è cancellabile? (il comando "Sposta nel cestino" è disattivato)

11 ottobre 2007 23.46
Giornale di Sistema ha detto...

Ciao Michele! :-)

Come sempre attentissimo, ti sei accorto della magagna. Per ora sì, la soluzione è "sudare" a manella ogni volta, nella 3a parte del post invece, descrivo come ottenere l'auto-mount della cartella condivisa.

La cartella non è cancellabile perché è stata creata mentre eri con i super poteri da amministratore e solo tornando amministratore la puoi cancellare.

Fai così. Apri il terminale e scrivi:

- sudo rmdir /mnt/'nome cartella condivisa'

e dai la password quando richiesta; la cartella viene cancellata. Mi appunto la cosa così magari la aggiungo nel prossimo post.

@:\>

12 ottobre 2007 12.01

venerdì 12 ottobre 2007

VirtualBox: cartelle condivise in guest Linux (2a parte)

Prima di procedere con la condivisione dell'intero disco rigido dell'host direttamente dall'interno del guest, voglio mostrarvi un attimo il significato del comando di mount che avevamo dato nella 1a parte del post, tra fig. 11 e fig. 12. Il comando era:

```
sudo mount -t vboxsf 'Cartella condivisa host' /mnt/Cartella condivisa host'
```

- *sudo*, chiede di eseguire i comandi successivi con i diritti di amministratore;
- *mount*, monta un dispositivo come può essere un disco rigido; in questo caso la vostra cartella condivisa. L'opzione Cartelle condivise di **fig. 7** nella 1a parte del post la rende infatti disponibile al guest Linux proprio come se fosse un dispositivo, e come tale può essere montata;
- *-t*, una opzione di mount per indicare il tipo di filesystem (cioè di organizzazione dei dati) del dispositivo che vogliamo montare;
- *vboxsf*, è il filesystem di VirtualBox;
- 'Cartella condivisa host', il dispositivo da montare. Gli apici servono solo perché il nome del dispositivo contiene spazi ma non fanno parte del nome stesso; se non ci fossero spazi nel nome, gli apici singoli non comparirebbero;
- */mnt/Cartella condivisa host*, la posizione della cartella in cui il dispositivo (cioè la nostra cartella condivisa) viene montato e reso disponibile sotto Linux.

Vediamo adesso come condividere l'intero disco rigido host direttamente dalla macchina guest in funzione. Cliccate su *Dispositivi* --> *Cartelle condivise* della finestra di VirtualBox in cui gira il guest (**fig. 14**)



fig. 14

Si aprirà nuovamente la finestra *Aggiungi condivisione*: cliccate sul pulsante di ricerca della cartella da condividere come indicato dal mouse in **fig 15**.

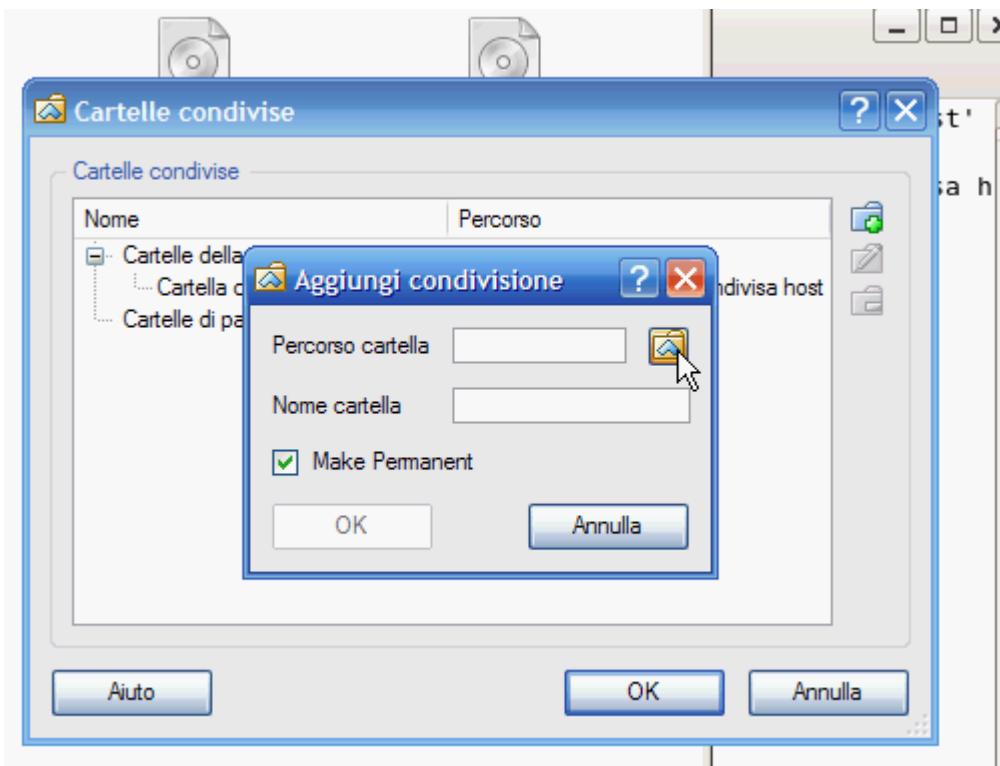


fig. 15

Questa volta però selezionate l'intero disco rigido: non c'è nulla infatti che ci vieta di farlo, anche se l'opzione che stiamo usando di VirtualBox si chiama Cartelle condivise! Cliccate su OK per confermare la scelta e proseguire.

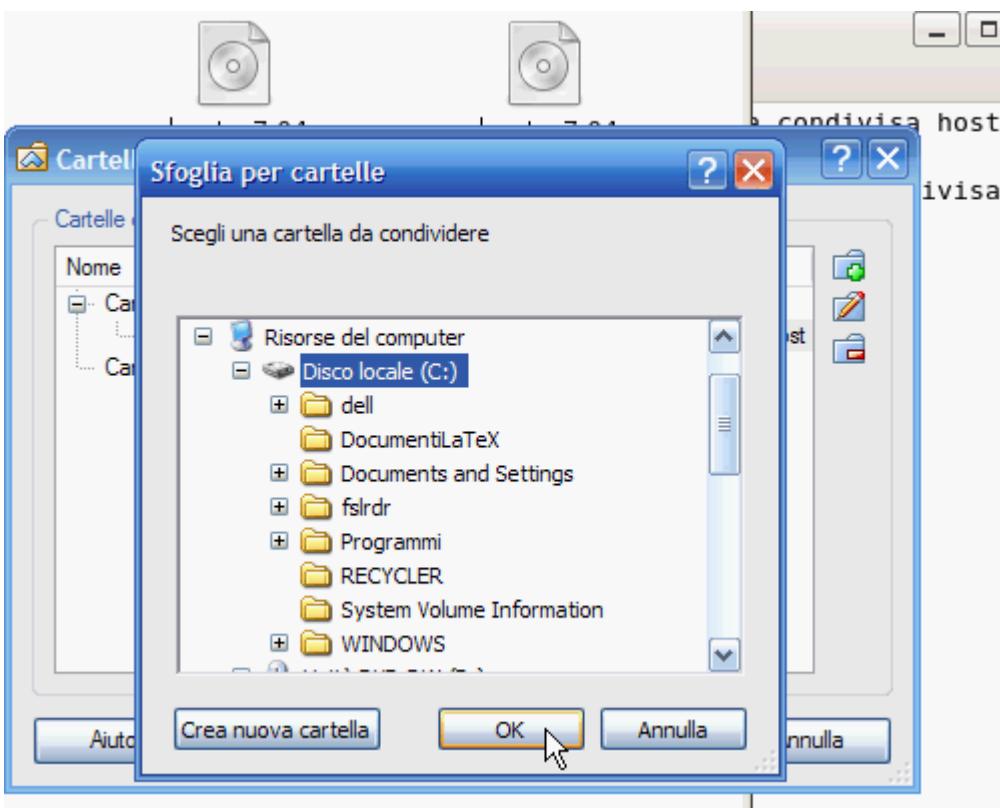


fig. 16

Tornerete così alla schermata precedente (fig. 17). Se volete cambiate pure il *Nome cartella* con cui il disco rigido sarà visibile nel guest, e poi scegliete se lasciare selezionata o meno l'opzione *Make Permanent*. Se è selezionata la condivisione è permanente, in caso contrario si tratterà di una *Cartella transiente*, la cui condivisione è valida solo per

questa sessione di lavoro. Quando spegnerete la macchina virtuale, badate, non un semplice riavvio, ma spegnimento e riaccensione, la condivisione andrà persa e occorrerà condividerla nuovamente se ne avete ancora bisogno.

Quando avete terminato, cliccate su *OK* per proseguire.

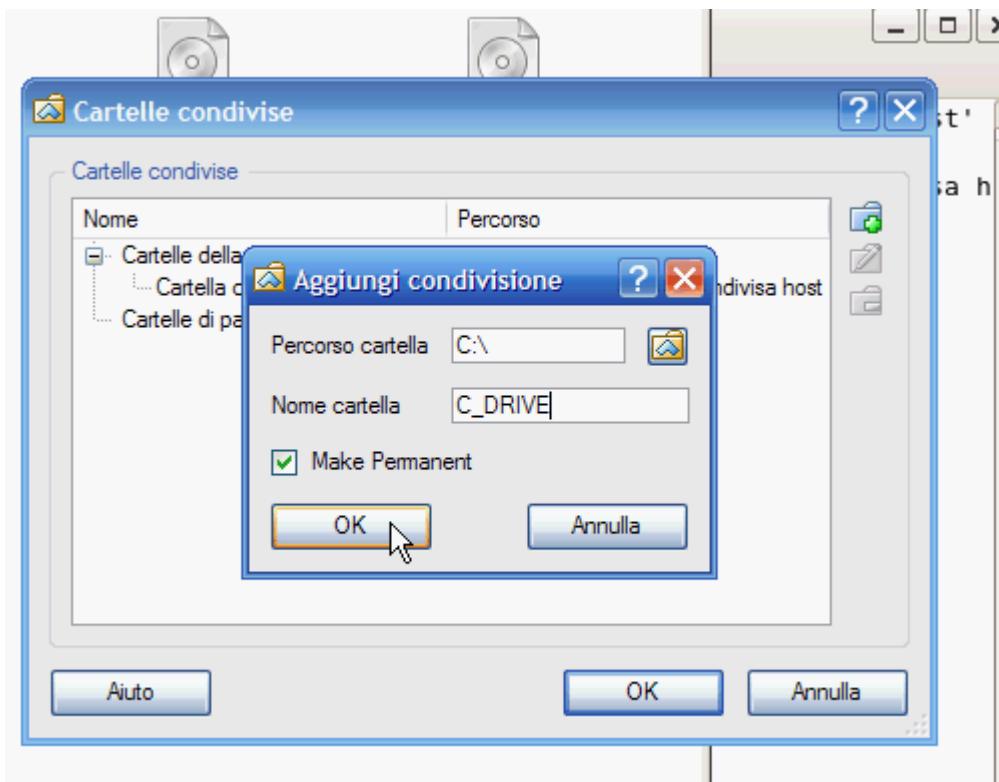


fig. 17

Ed eccola! Nel vostro file manager è apparsa l'icona di una nuova cartella: **C_DRIVE** (fig. 18); cliccateci sopra per aprirla.

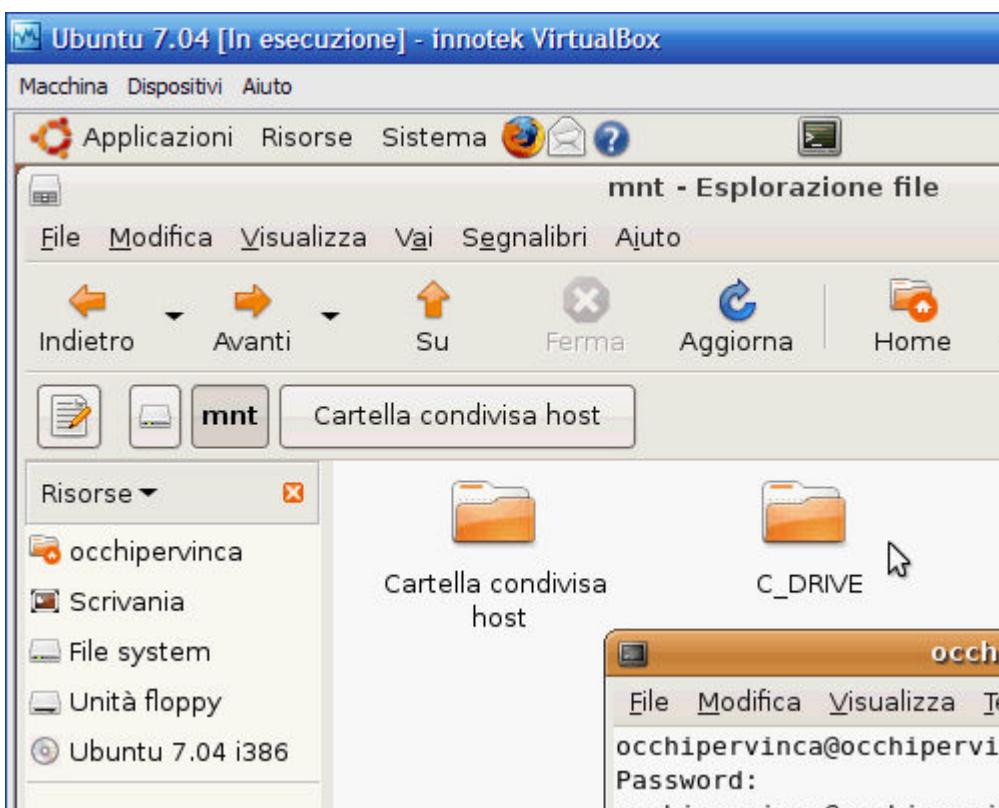


fig. 18

Vi ritroverete così a navigare il disco rigido dell'host (fig. 19) che era quello che volevamo! Ricordate però: state soltanto navigando il disco rigido dell'host, come se vi foste affacciati ad una finestra che vi fa dare un'occhiata al contenuto. I file perciò, anche se li vedete dal guest, fisicamente sono ancora sull'host. Se li volete fisicamente sul guest, allora dal vostro file manager trascinateli in una cartella del guest stesso e verranno copiati lì.

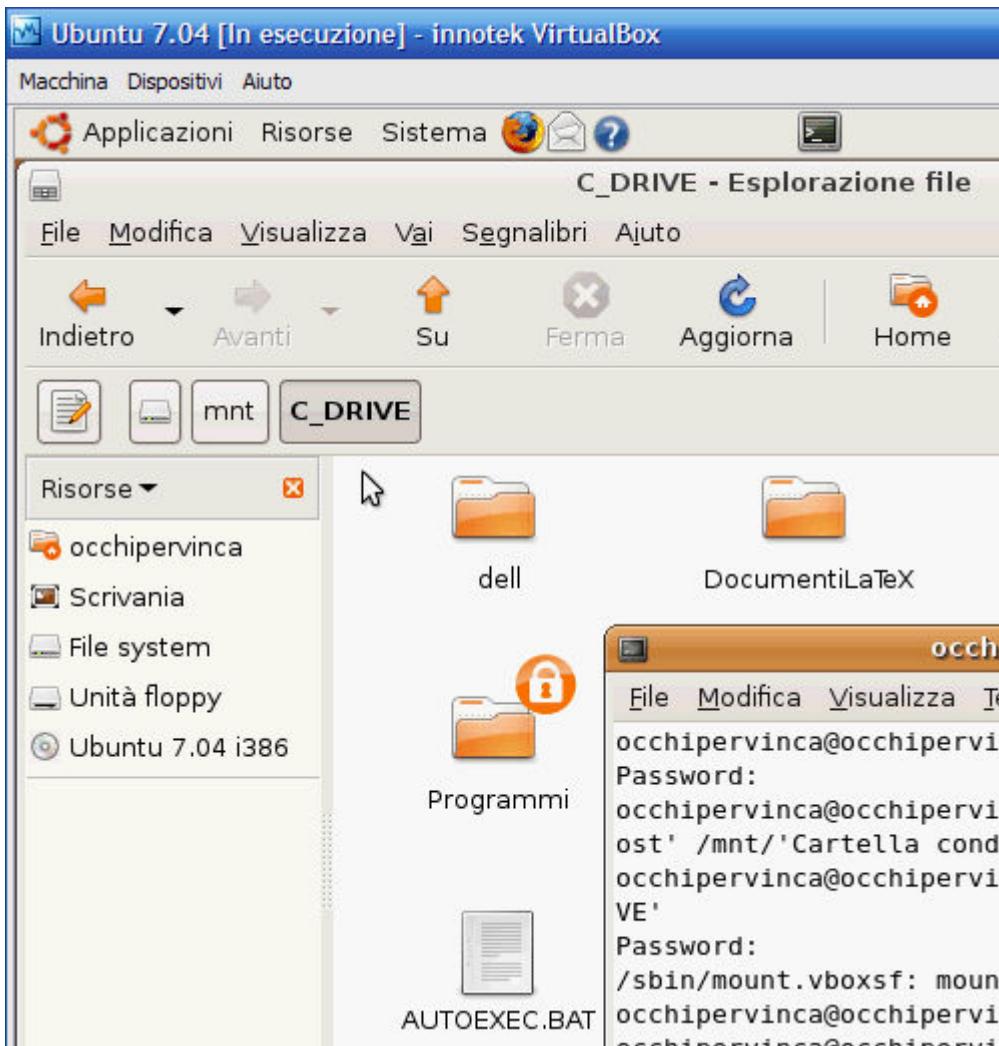


fig. 19

E' tutto! O forse no... forse no, anzi, no! Proprio per niente.

Come Michele Benvegnù ha segnalato (grazie Michele, sempre preziosissimo! :-D), quanto abbiamo visto finora nelle 2 parti di questo post è soltanto teoria (questo sono io a dirlo). Nella pratica VirtualBox non può scavalcare quelli che sono i meccanismi di funzionamento interni di Linux.

Risultato: se provate a riavviare la macchina virtuale, vi accorgerete che nei vostri punti di mount, *Cartella condivisa host* e *DRIVE_C* non è visibile più nulla. Occorre rifacciate ogni volta a manella il mount della cartella condivisa con i comandi:

```
sudo mount -t vboxsf 'Cartella condivisa host' /mnt/C_DRIVE
```

oppure

```
sudo mount -t vboxsf 'C_DRIVE' /mnt/C_DRIVE
```

Lo trovate noioso? Sono d'accordo con voi! Ecco perché nella terza parte di questo post vedremo come fare l'auto-mount della cartella condivisa. ;-D

@:1>

Pubblicato alle 11.39 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

sabato 13 ottobre 2007

VirtualBox: cartelle condivise in guest Linux (3a parte)

Chiudevo la 2a parte di questo post dicendo che VirtualBox non può scavalcare il funzionamento interno di Linux. Il risultato è che lo spegnimento, od anche il semplice riavvio della macchina virtuale, vanifica tutto il lavoro fatto per rendere visibile la cartella condivisa nel guest. Bene, vorrà dire che andremo a fare noi a manella il lavoro che, spero già dalla prossima versione, VirtualBox potrebbe fare da se: la modifica dei file di configurazione di Linux per fare l'auto-mount della cartella condivisa. Avviate la vostra macchina virtuale, si parte!

Mentre la macchina virtuale parte, vi spiego un attimo il lato tecnico del problema; se invece non siete curiosi di sapere perché Linux ad ogni spegnimento o anche semplice riavvio si perde la cartella condivisa, allora saltate direttamente al paragrafo sopra **fig. 21**.

Linux all'avvio esegue tutta una serie di operazioni, compresa quella di *mount*. Tramite questa rende disponibili nel filesystem tutti i dispositivi che trova elencati nel file */etc/fstab*, ad esempio il disco rigido oppure il lettore CD/DVD-ROM. Ed ecco il motivo per cui al riavvio si perde la cartella condivisa: essa semplicemente non viene montata perché non è indicata fra i dispositivi da montare nel file *fstab*; VirtualBox non effettua questa modifica, ed in realtà non servirebbe comunque.

Chi infatti rende disponibile sotto forma di dispositivo da montare la nostra cartella condivisa, è un driver delle VirtualBox Guest Additions (ecco perché vanno installate prima di fare questi giochetti con le cartelle condivise). Questo driver viene caricato molto dopo che Linux ha già fatto il *mount* dei dispositivi indicati in *fstab*, pertanto anche trovandovi l'indicazione di montare la cartella condivisa, in quel momento non è ancora disponibile e non verrebbe caricata.

Occorre dare il comando di *mount* da qualche altra parte, in un punto in cui il driver delle Guest Additions è già stato caricato e la cartella condivisa resa disponibile come dispositivo da montare. Un file letto da Linux sempre all'avvio ma dopo il driver delle Guest Additions è */etc/init.d/rc.local*. Vediamo come modificarlo aggiungendovi il nostro sospirato comando di *mount*.

Aprite il vostro terminale e date il comando *sudo gedit* che vi permetterà di lanciare gedit, l'editor di gnome, con permessi di amministratore. In alternativa, sul vostro sistema date *sudo nome_vostro_editor* (**fig. 21**). Date la password e poi un invio per proseguire.

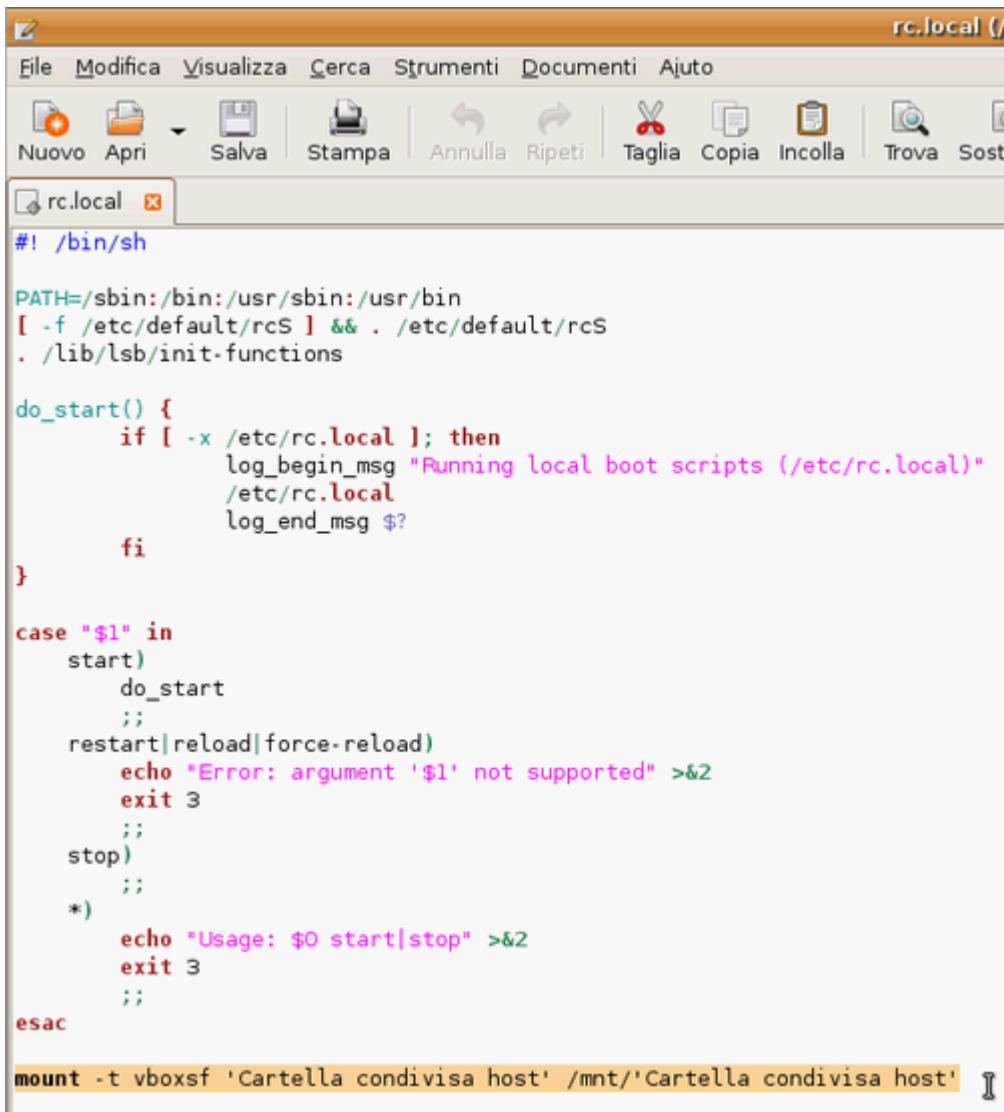


fig. 21

Si aprirà gedit con i permessi da amministratore, potrete pertanto modificare qualunque cosa sul vostro sistema, fate la massima attenzione a modificare esattamente come indicato se non volete trovarvi con un Linux in panne. Cliccate sul pulsante apri e navigate le cartelle fino a */etc/init.d/*, lì aprirete il file *rc.local* (**fig. 22**). In fondo a quello che è uno script, aggiungete la riga:

```
mount -t vboxsf 'Cartella condivisa host' /mnt/'Cartella condivisa host'
```

Uscite salvando le modifiche.



```
rc.local (/)
File Modifica Visualizza Cerca Strumenti Documenti Ajuto
Nuovo Apri Salva Stampa Annulla Ripeti Taglia Copia Incolla Trova Sost
rc.local x
#!/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin
[ -f /etc/default/rcS ] && . /etc/default/rcS
. /lib/lsb/init-functions

do_start() {
    if [ -x /etc/rc.local ]; then
        log_begin_msg "Running local boot scripts (/etc/rc.local)"
        /etc/rc.local
        log_end_msg $?
    fi
}

case "$1" in
    start)
        do_start
        ;;
    restart|reload|force-reload)
        echo "Error: argument '$1' not supported" >&2
        exit 3
        ;;
    stop)
        ;;
    *)
        echo "Usage: $0 start|stop" >&2
        exit 3
        ;;
esac

mount -t vboxsf 'Cartella condivisa host' /mnt/'Cartella condivisa host'
```

fig. 22

Riavviate la macchina virtuale e date un'occhiata nella cartella condivisa! ;-D

Se sul vostro sistema non dovesse essere presente questo file, potete fare le stesse modifiche nel file /etc/rc.d/rc.local, non le ho provate ma dovrebbero funzionare bene.

E se invece non volessimo fare l'auto-mount della cartella condivisa ma montarla al volo di volta in volta senza stare a ricordarsi tutto il comando per intero? Qualcuno ha detto script?!? Avete detto bene, ma tutto questo, nella 4a parte del post.

@:>

Pubblicato alle 15.27 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

3 commenti:

MCMXC ha detto...

Ha funzionato alla perfezione, grazie infinite!D

Purtroppo non funziona con Sabayon, perche' stranamente li' il file rc.local proprio non c'e', ma pazienza... (Credo siano le estensioni a non essere supportate in verita'...)

In compenso, non so come ma ora entrambe le distribuzioni finalmente vedono la chiavetta USB!):

13 ottobre 2007 17.01

Giornale di Sistema ha detto...

Oohhh... e finalmente anche questa benedetta chiavetta USB ci ha dato soddisfazione! :-D

Sono davvero contento abbia funzionato tutto bene e grazie a te per aver trovato la pazienza di leggere i post. :-)
)

13 ottobre 2007 19.53

MCMXC ha detto...

Grazie per averle scritte:)

13 ottobre 2007 20.58

martedì 16 ottobre 2007

VirtualBox: compattare dischi virtuali ad espansione dinamica (3 parte)

Nella 1a parte di questo post avevamo inquadrato la problematica: quando liberiamo spazio sul disco rigido della macchina virtuale, il file vdi sull'host che lo contiene non si ridimensiona di conseguenza: occorre forzarne il ridimensionamento con un apposito comando, *modifyvdi*, disponibile solamente dalla console a linea di comando dell'host attraverso il gestore di comandi di VirtualBox, *VboxManage*.

Purtroppo però ciò non è sufficiente, occorre che sia soddisfatto un pre-requisito: lo spazio libero sul guest deve essere marcato a zero (il valore, non la cifra) prima che il comando possa agire su di esso compattandolo. Questo perché quando cancellate un file, questi non viene realmente cancellato, semplicemente lo spazio che occupa viene dichiarato libero in una apposita tabella.

I settori del disco rigido quindi, contengono ancora i dati, ed è per questo che il comando *modifyvdi* non riesce a fare il suo lavoro: può infatti agire solo su spazio realmente vuoto e non su quello dichiarato semplicemente come tale. Occorre un programma che svuoti per davvero lo spazio dichiarato libero.

VirtualBox non ha un comando interno per marcare a zero i settori liberi del disco rigido virtuale, ma niente paura, come scopriremo tra breve ciò non è un problema con un **guest Linux**. Quanto vedremo poi è identico per tutte le distribuzioni, anche se le immagini per forza di cose devono far riferimento ad una in particolare ed in questo caso si tratta di una Ubuntu. Cominciamo.

Avviate la macchina virtuale della vostra distribuzione Linux e aprirete il file manager (l'analogo dell'Esplora risorse per intenderci). Se date un'occhiata alla cartella che rappresenta l'intero filesystem (**fig. 1**), vederete quanto spazio libero in questo momento vi è sul vostro disco rigido virtuale (guardate sotto il mouse), nella macchina di esempio ancora 15,4 GB.

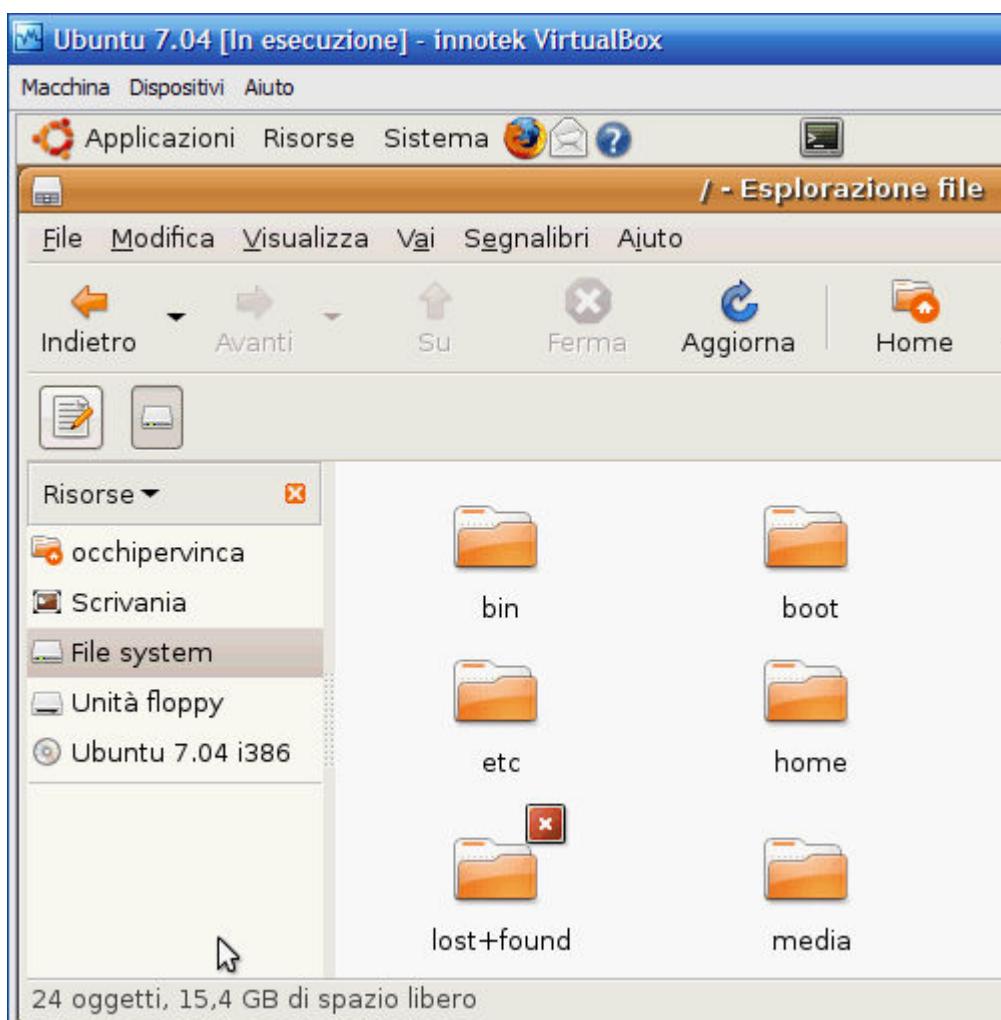


fig. 1

Diamo un'occhiata alle dimensioni del corrispondente file vdi sull'host contenente il disco rigido virtuale. Ci saremmo aspettati un file dalla dimensioni di circa:

20 GB - 15,4 GB = 5,6 GB

ma come potete vedere, ve ne è uno da 7 GB (**fig. 2**). La discrepanza è dovuta al fatto che il file manager non vi sta mostrando tutto lo spazio realmente utilizzato, manca ad esempio quello del file di swap: in Windows questo è un file come tutti gli altri (C:\pagefile.sys), su Linux è una vera e propria partizione separata adibita, esclusivamente a questo scopo e con un suo specifico tipo di filesystem.

Nome	Dimensione	Tipo
.VirtualBox	3.411.025 KB	File VDI
Machines	5.239.889 KB	File VDI
VDI	7.293.009 KB	File VDI
Application Data	1.442.900 KB	File VDI

fig. 2

Visto che ora sappiamo come copiare file dall'host tramite le cartelle condivise, ne possiamo approfittare per copiarci da lì dei file molto grandi; esempio: 6 file da 700 MB ottenuti copiando nella cartella condivisa, con 6 nomi diversi, la iso d'installazione di Ubuntu. Dalla cartella condivisa, copiamoli sul Desktop di Linux (**fig. 3**).

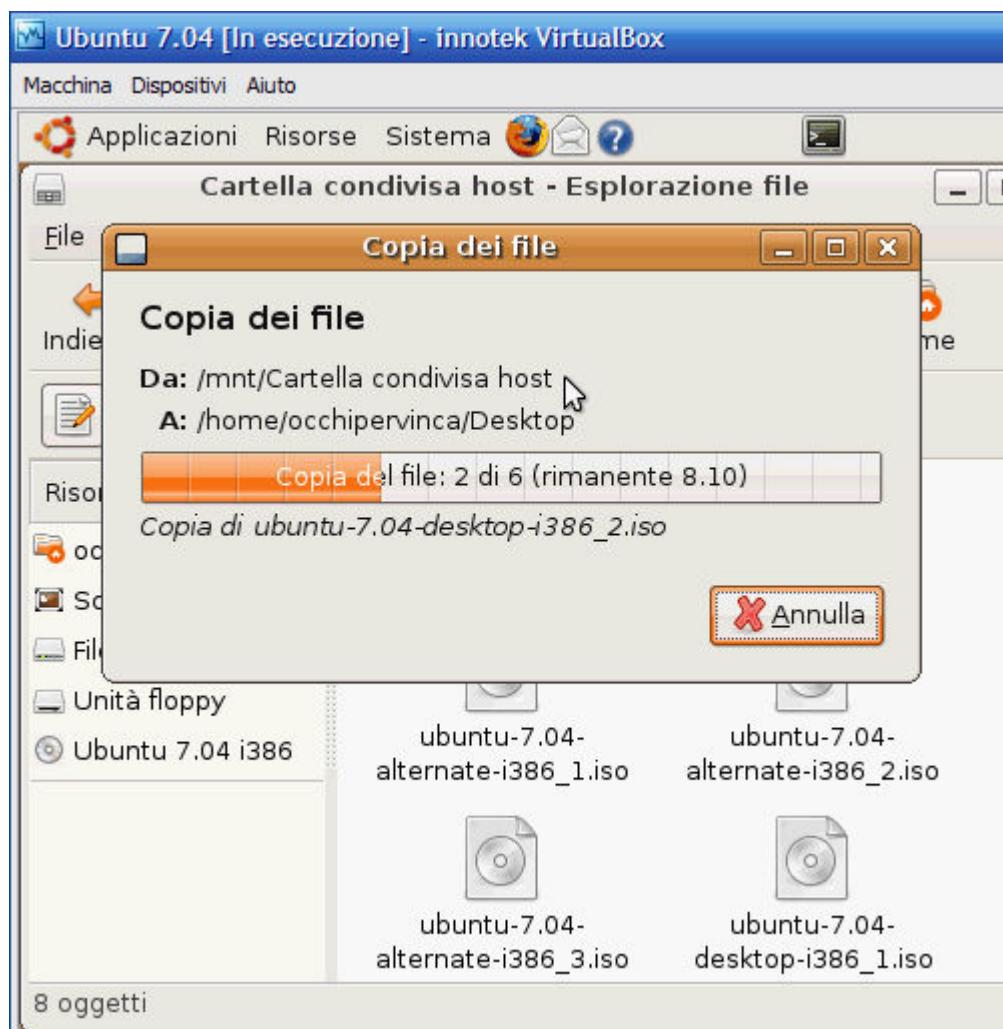


fig. 3

Terminata la copia, date un'occhiata nel file manager alla spazio libero, è diminuito a (**fig. 4**):

15,4 GB - (6 * 700 MB) = ~11,3 GB

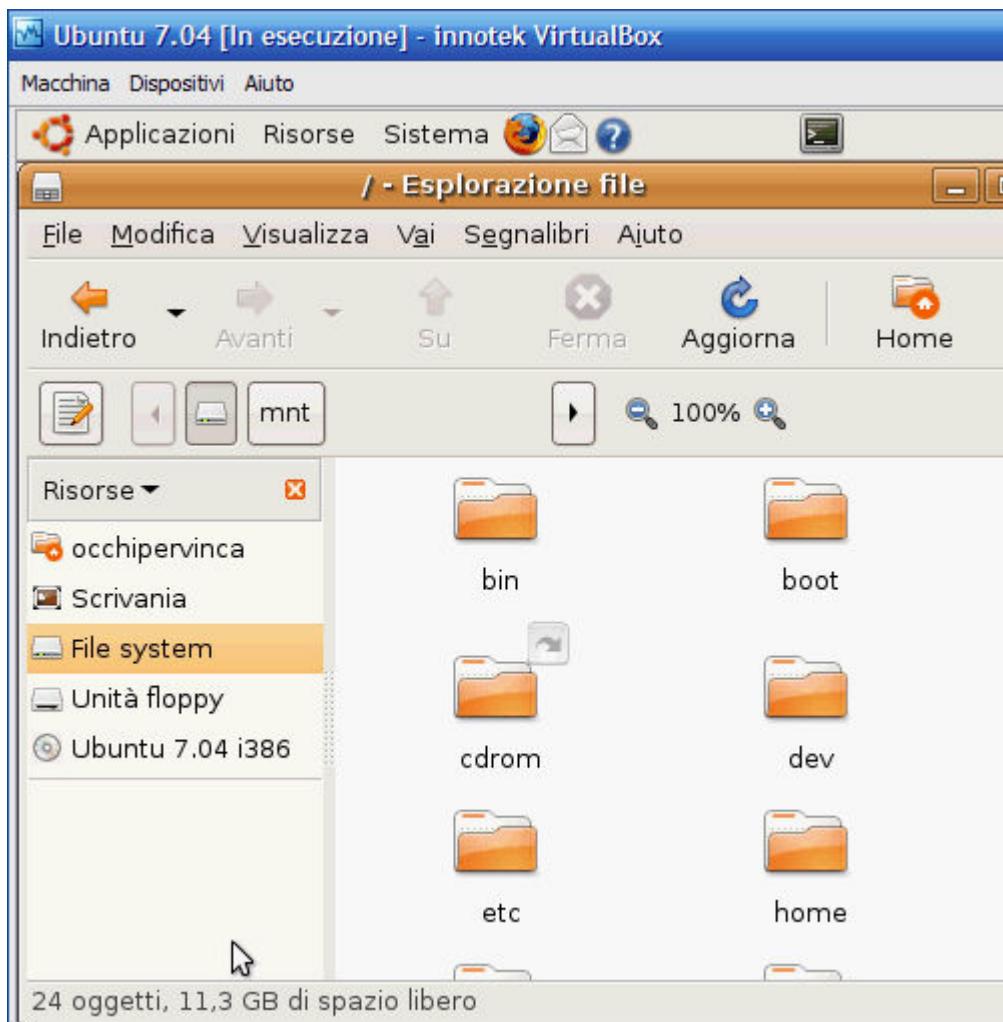


fig. 4

Corrispondentemente anche il file vdi sull'host è cresciuto, anche se con delle discrepanze per via della diversa gestione dello spazio disco fra Linux e Windows (fig. 5).

Cartelle	Nome	Dimensione	Tipo
.VirtualBox	Mandriva_2008.vdi	3.411.025 KB	File VDI
Machines	openSUSE10_2.vdi	5.239.889 KB	File VDI
VDI	Ubuntu7_04.vdi	10.313.809...	File VDI
Application Data	Windows XP.vdi	1.442.900 KB	File VDI

fig. 5

Ora eliminate i file copiati sul vostro Desktop Linux e verificate che lo spazio libero sia tornato quello originale (fig. 6). Verificate poi le dimensioni del file vdi sull'host; sono restate quelle di fig. 5: come ci aspettavamo, non si è ridimensionato.

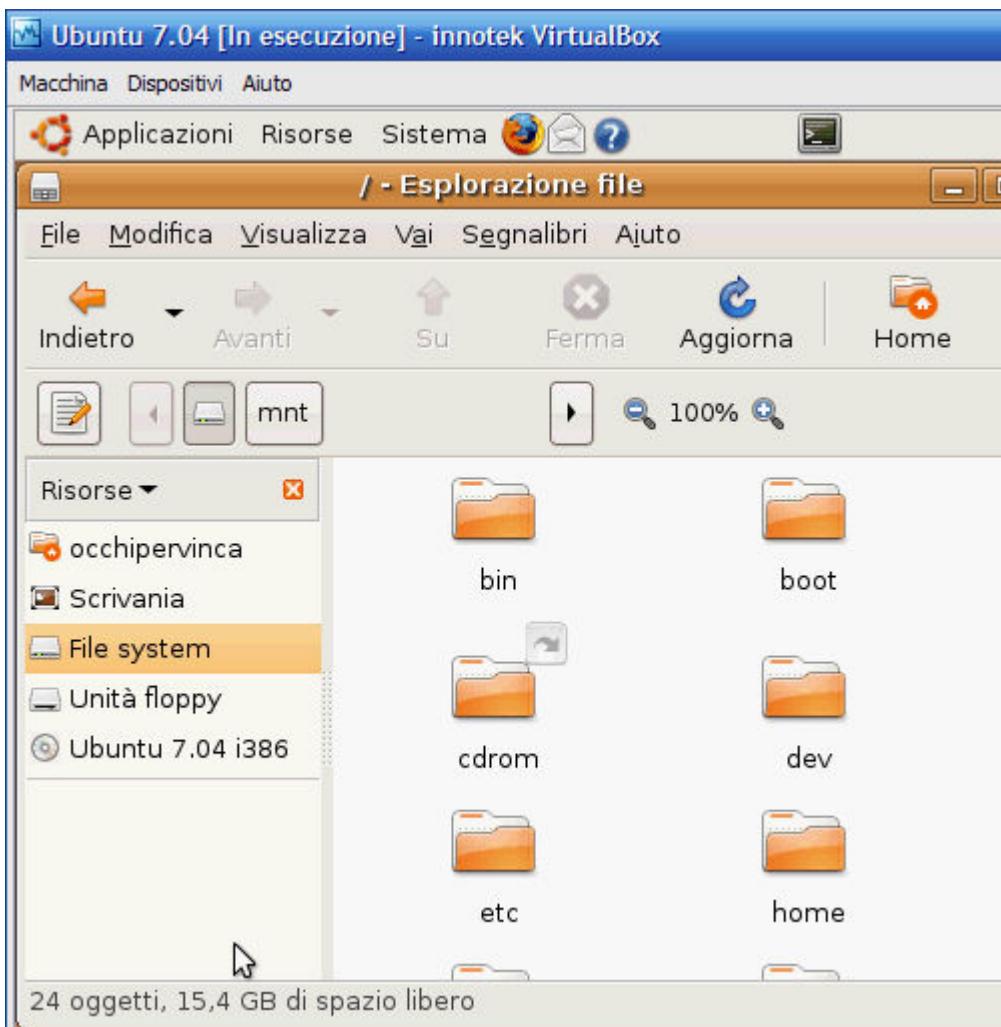


fig. 6

Bene, iniziamo allora a lavorare alla sua compattazione. Vi riassumo le fasi su Linux perché leggermente diverse da Windows, con dei passaggio in più:

- **fase 1:** creazione di un unico file contenente caratteri e grande quanto tutto lo spazio libero sul guest così da riempirlo completamente;
- **fase 2:** sovrascrittura del file con valori zero;
- **fase 3:** cancellazione del file;
- **fase 4:** compattazione del file vdi sull'hot.

Vediamole una ad una.

Fase 1: la prima cosa da fare è creare un file.

Per farlo, aprirete la finestra del terminale (shell) e date questo comando (fig. 7):

```
touch /home/utente/Desktop/filetuttiZero
```

Il comando `touch` crea il file di nome "filetuttiZero" nella cartella `/home/utente/Desktop`. Guardate infatti il vostro Desktop, è comparso un file con questo nome.

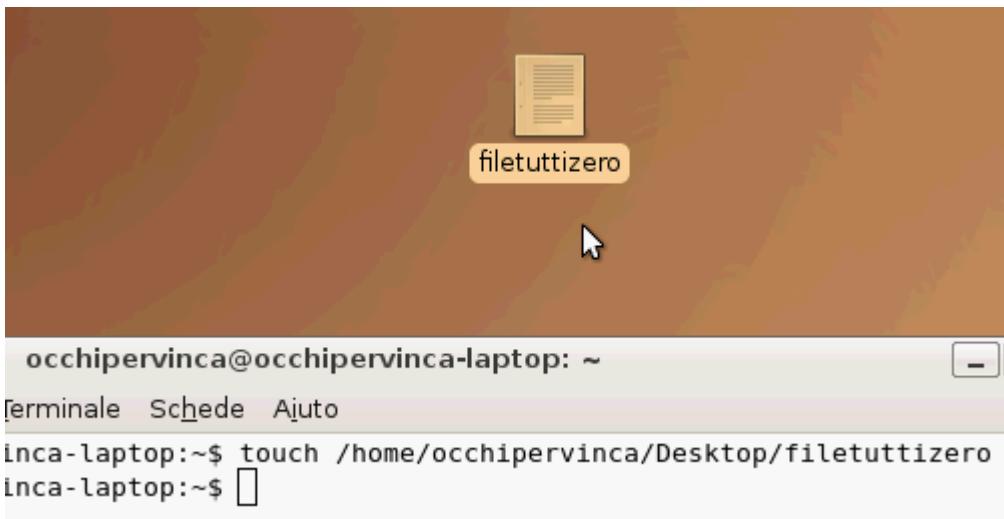


fig. 7

In questo momento il file è vuoto, (provate a guardarne le dimensioni, è zero byte), riempiamolo di caratteri: per comodità lo riempiremo di caratteri zero (caratteri questa volta, non valori, questo lo faremo dopo con un altro comando) perché su Linux c'è un dispositivo virtuale che ce li fornisce "aggratis". Sempre dalla vostra finestra di terminale, date il comando

```
cat /dev/zero > /home/ nomeutente/Desktop/filetuttizero
```

che significa:

- *cat* sta per conCATenare, concatena 2 file fra loro;
- */dev/zero* è il dispositivo virtuale che fornisce caratteri zero a tutto spiano;
- *>* è l'operatore di redirezione, prende ciò che lo precede, *cat /dev/zero*, ossia i caratteri zero che questi fornisce, e li manda su ciò che segue, cioè il file *filetuttizero*.

Andate adesso a prendervi un bel caffè mentre *filetuttizero* incamererà sempre più caratteri zero fino a saturare tutto lo spazio libero sul guest. Quando questo è avvenuto, nel terminale comparirà un messaggio del tipo di **fig. 8**:

cat: errore di scrittura: Niente spazio sul device.



fig. 8

Controllate lo spazio libero del vostro guest nel file manager (**fig. 9**), è effettivamente terminato (*0 byte di spazio libero*).

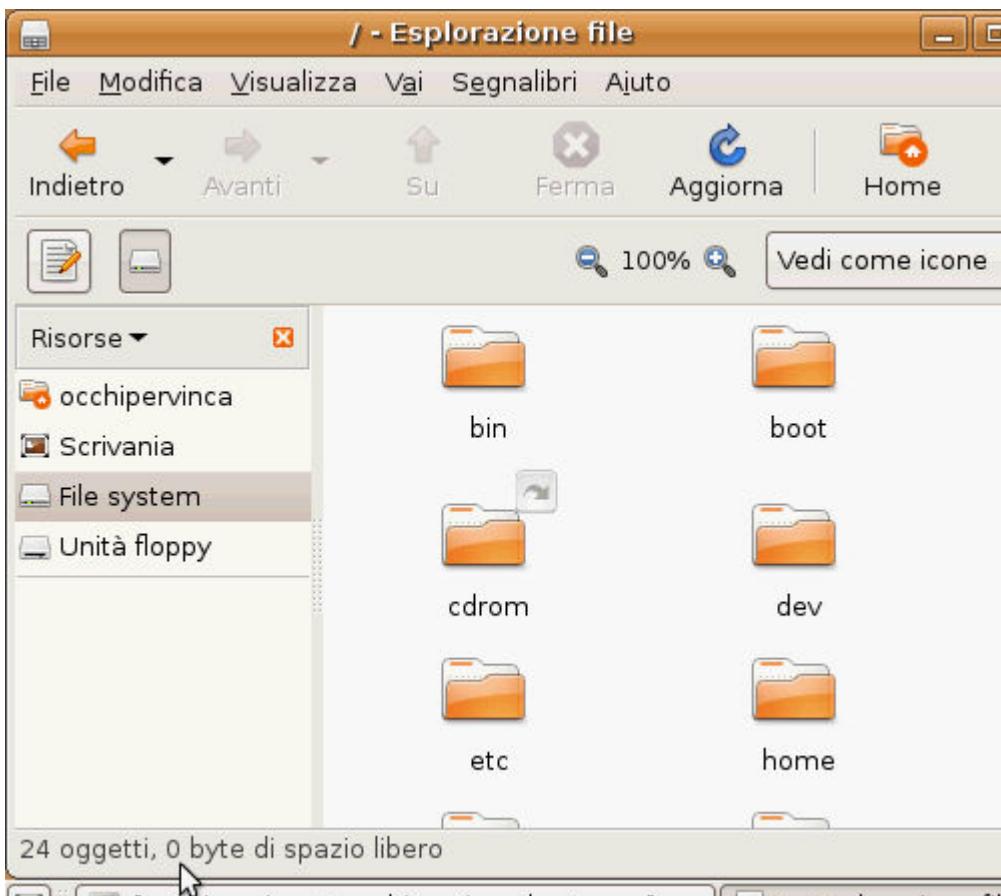


fig. 9

Idem se verificate le dimensioni del file vdi sull'host (fig. 10).

Cartelle	Nome	Dimensione	Tipo
.VirtualBox	Mandriva_2008.vdi	3.411.025 KB	File VDI
Machines	openSUSE10_2.vdi	5.239.889 KB	File VDI
VDI	Ubuntu7_04.vdi	19.046.481...	File VDI
Application Data	Windows XP.vdi	1.442.900 KB	File VDI

fig. 10

Fase 2-3: riempiamo *filetuttizer0* con dei valori zero (in questo momento contiene caratteri zero) ed infine lo cancelliamo (vi spiego tra poco perché).

Per farlo useremo un altro comando Linux: *shred*. Il suo compito è di sovrascrivere un file con una serie di caratteri casuali per il numero di volte che vogliamo. Quando ha finito l'ultima passata, possiamo aggiungere un'ulteriore passata di valori zero (proprio quello che ci serve) e infine cancellare il file.

Dalla finestra di terminale date il comando:

```
shred -f -n0 -z -v -u /home/occhipervinca/Desktop/filetuttizer0
```

f = force, forza i permessi sui file affinché possiate comunque eseguire sempre il comando;

n = 0, non ci interessa riempire *filetuttizer0* di caratteri casuali, perciò gli indichiamo di fare zero passate di scrittura di caratteri casuali, cioè nessuna;

z = riempie di valori zero il file dopo l'ultima passata, non vi sono passate precedenti (*n* = 0), perciò lo fa subito;

v = verbose, modalità chiacchierona del comando, vi aggiorna man mano su quello che sta facendo, giusto per capire a che punto siamo;

u = il file alla fine di tutti i passaggi viene cancellato.

In fig. 11 il comando ha iniziato il suo lavoro, approfittatene per un altro caffè! ;-)

```
occhipervinca@occhipervinca-laptop: ~
File Modifica Visualizza Terminale Schede Ajuto
occhipervinca@occhipervinca-laptop:~$ touch /home/occhipervinca/Desktop/filetutt
izero
occhipervinca@occhipervinca-laptop:~$ cat /dev/zero > /home/occhipervinca/Desktop/filetuttizero
cat: errore di scrittura: Niente spazio sul device
occhipervinca@occhipervinca-laptop:~$ shred -f -n0 -u -z -v /home/occhipervinca/Desktop/filetuttizero
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...192MiB/16
GiB 1%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...281MiB/16
GiB 1%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...383MiB/16
GiB 2%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...500MiB/16
GiB 3%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...632MiB/16
GiB 4%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...724MiB/16
GiB 4%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...832MiB/16
GiB 5%
shred: /home/occhipervinca/Desktop/filetuttizero: passo 1/1 (000000)...856MiB/16
GiB 5%
```

fig. 11

Quando ha terminato, a terminale comparirà il messaggio di **fig. 12** o uno analogo: il file è stato rimosso.

fig. 12

Sappiamo già che il file non viene realmente cancellato ma solo dichiarato libero lo spazio che occupava, spazio che è colmo di valori zero perché questo era il suo contenuto immediatamente prima di essere cancellato: proprio quello che serve al comando di compattazione *modifyvdi* per funzionare.

Fase 4: è arrivato finalmente il momento di compattare il file vdi sull'host: ora è possibile, la precondizione che lo spazio libero sul disco rigido virtuale fosse marcata a zero è soddisfatta. Aprite una finestra DOS (*pulsante start --> Esegui --> scrivete cmd e date invio*). Al suo interno date questo comando, doppi apici compresi anche su *nomefile* se questo contiene spazi vuoti:

"C:\Programmi\innotek VirtualBox\VBoxManage" modifyvdi "nomefile".vdi compact

Nel caso della nostra macchina virtuale di esempio diventa:

```
"C:\Programmi\innotek VirtualBox\VBoxManage" modifyvdi Ubuntu7_04.vdi compact
```

Mentre il comando fa il suo lavoro, vi spiego la sintassi per chi non la ricorda o non ha seguito la 2a parte del post:

- intanto i doppi apici anche qui servono ove ci siano spazi vuoti. Normalmente la linea di comando interpreta gli spazi vuoti come la fine di qualcosa; se non mettiamo i doppi apici, C:\Programmi\innotek VirtualBox\VBoxManage non viene interpretata come una cosa unica ma viene presa solo la prima parte, cioè C:\Programmi\innotek e li si ferma dando errore. Stessa cosa con il nome del file vdi, i doppi apici sono obbligatori solo se il nome del file vdi presenta spazi al suo interno;

- *VBoxManage* è il comando che lancia il manager a linea di comando di VirtualBox, serve a dare da linea di comando tutti i comandi che normalmente date dall'interfaccia grafica, e altri che l'interfaccia grafica non implementa ancora ma sono comunque disponibili da linea di comando;

- *modifyvdi* è il comando impartito a *VBoxManage*, gli dice che volete modificare il file nomefile.vdi;

- *compact* dice che la modifica che volete effettuare sul file vdi è la sua compattazione.

Nel frattempo la compattazione dovrebbe essere finita, in **fig. 13** il risultato: come potete vedere, il file vdi è tornato all'incirca alle sue dimensioni iniziali.

Nome	Dimensione	Tipo
Mandriva_2008.vdi	3.411.025 KB	File VDI
openSUSE10_2.vdi	5.239.889 KB	File VDI
Ubuntu7_04.vdi	7.286.865 KB	File VDI
Windows XP.vdi	1.442.900 KB	File VDI

fig. 13

Wooww... post lunghissimo, frutto della decisione di rendere indipendenti fra loro la seconda e la terza parte del post. Per far questo, ho duplicato in parte i contenuti, così da non costringervi a salti dall'uno all'altro post e favorire la comprensibilità dell'argomento; spero di esserci riuscito. :-)

Ok, è tutto, non mi resta che augurarvi, una serena compattazione!

@:\>

Pubblicato alle 6.53 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

giovedì 18 ottobre 2007

VirtualBox e Mandriva 2008: matrimonio perfetto

Non riuscita dopo 5 tentativi l'installazione in VirtualBox di openSUSE 10.3, sono passato oltre. In questi giorni poi è uscita anche la Mandriva 2008, e così non ho perso l'occasione di farci un giro (**fig. 1**, interfaccia grafica KDE).

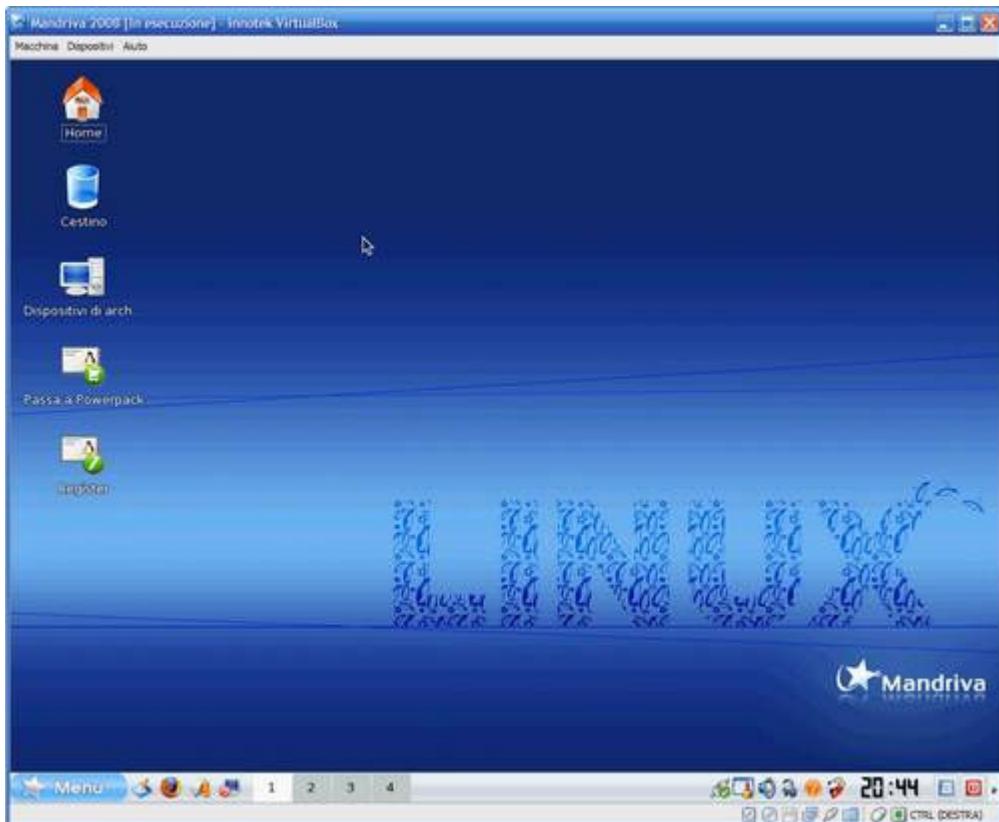


fig. 1

Beh, che dire, un vero spasso. Installazione semplice e rapida, ma soprattutto, una bella sorpresa se viene virtualizzata in VirtualBox (**fig. 2**).



fig. 2

Ebbene sì, avete visto bene! La Mandriva riconosce correttamente la scheda video come quella resa disponibile da VirtualBox, e sin da subito fornisce maggiori modalità grafiche rispetto ai classici 1024x768 pixel o anche meno.

E non è finita qua. Appena terminata l'installazione ed il primo riavvio, una volta spenta e riaccesa la macchina virtuale di Mandriva 2008, sorpresa: il mouse non casca nella finestra come accade normalmente. Fatto un giro di verifica, ecco spiegato il motivo (**fig. 3**).

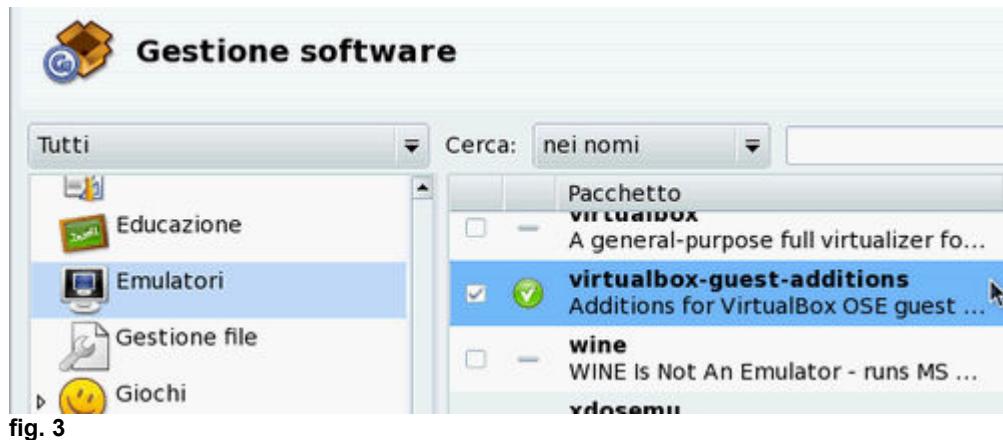


fig. 3

Le VirtualBox Guest Additions in versione OSE sono già installate sul sistema, una gran bella novità che rende sin da subito più usabile il sistema.

@:\>

Pubblicato alle 13.41 Argomenti: Linux, news, Virtualbox

giovedì 18 ottobre 2007

Virtuabox 1.5.2 è arrivato

Proprio stamane scrivevo in una e-mail di sperare in una nuova versione di VirtualBox come regalo per Natale, e invece: è online la versione 1.5.2 di VirtualBox!

Dal changelog e come testimonia anche il numero di versione, non ci sono novità eclatanti, ma tutta una serie di bugfix e miglioramenti che dovrebbero migliorarne la stabilità d'uso e la velocità. Installo e vi dico.

@:)\>

Pubblicato alle 21.18 Argomenti: [news](#), [Virtualbox](#), [VirtualBox Guest Additions](#), [virtualizzazione](#)

venerdì 19 ottobre 2007

VirtualBox e Ubuntu 7.10: matrimonio possibile

E mai siffù un cotal delirio... ma dato il mortal sospiro de la pazienza mia, rimedio posì in aramaico antico. Ora: non crediate che conosca l'italiano, ne quello corrente ne quello colto dei nostri avi... è solo l'effetto collaterale dello sclero per installare la nuova Ubuntu 7.10.

Dite la verità, anche voi avete avuto problemi d'installazione su Virtualbox (dite di sì, almeno per consolarmi): forse anche voi avete scaricato la iso, avviato VirtualBox, e creata al volo una macchina virtuale vi siete catapultati ad installarla, fermandovi tristemente davanti alla prima schermata (**fig. 1**).

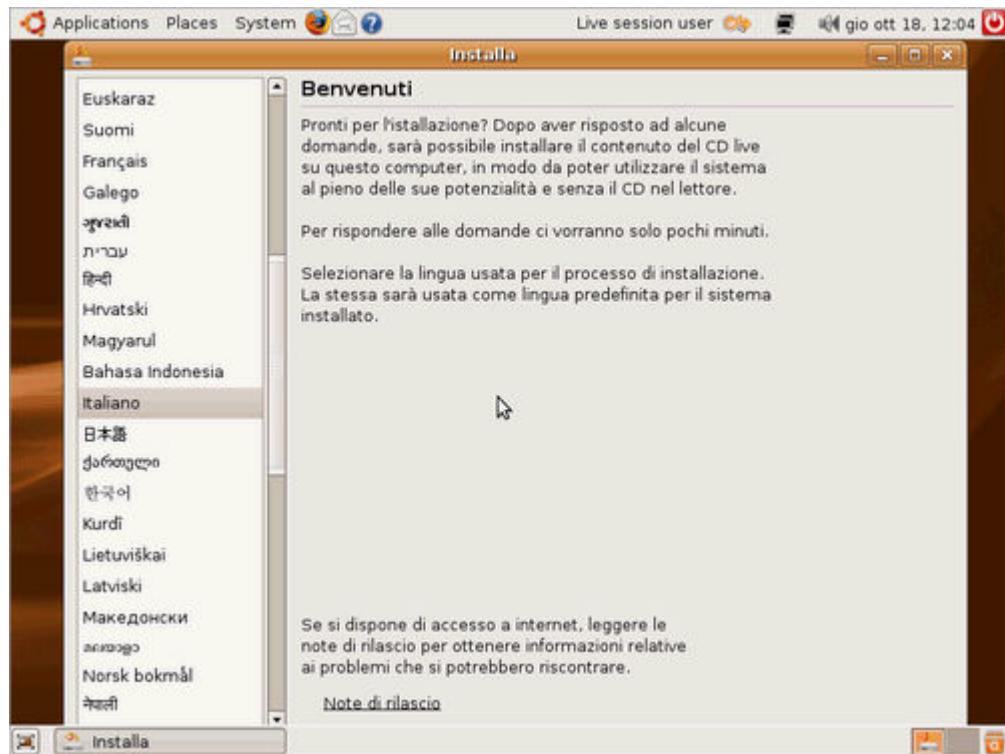


fig. 1

Problemi di riconoscimento dell'hardware della macchina virtuale portano infatti Ubuntu ad usare un monitor ed una scheda video generici settati a 800x600 pixel; peccato che anche le schermate d'installazione abbiano minimo questa risoluzione. Risultato: non si vedono i pulsanti Avanti-Indietro per proseguire nell'installazione, ne si possono ridimensionare le schermate d'installazione, sono già alla loro dimensione minima. Se anche voi avete questo problema allora vediamo insieme come risolverlo.

Cliccate su System --> Amministrazione --> Schermi e grafica (**fig. 2**).



fig. 2

Cliccate su Model (fig. 3).



fig. 3

Scegliete un modello di monitor che sia il vostro o ci si avvicini, scegliendo fra quelli generici ad esempio (fig. 4).



fig. 4

Scegliete una risoluzione video che sia adeguata alle caratteristiche del monitor scelto, comunque almeno un 1024x768, la risoluzione minima per riuscire a superare il problema della schermata di installazione non interamente visibile (**fig. 5**). Quando avete terminato, cliccate sul pulsante Test in basso.

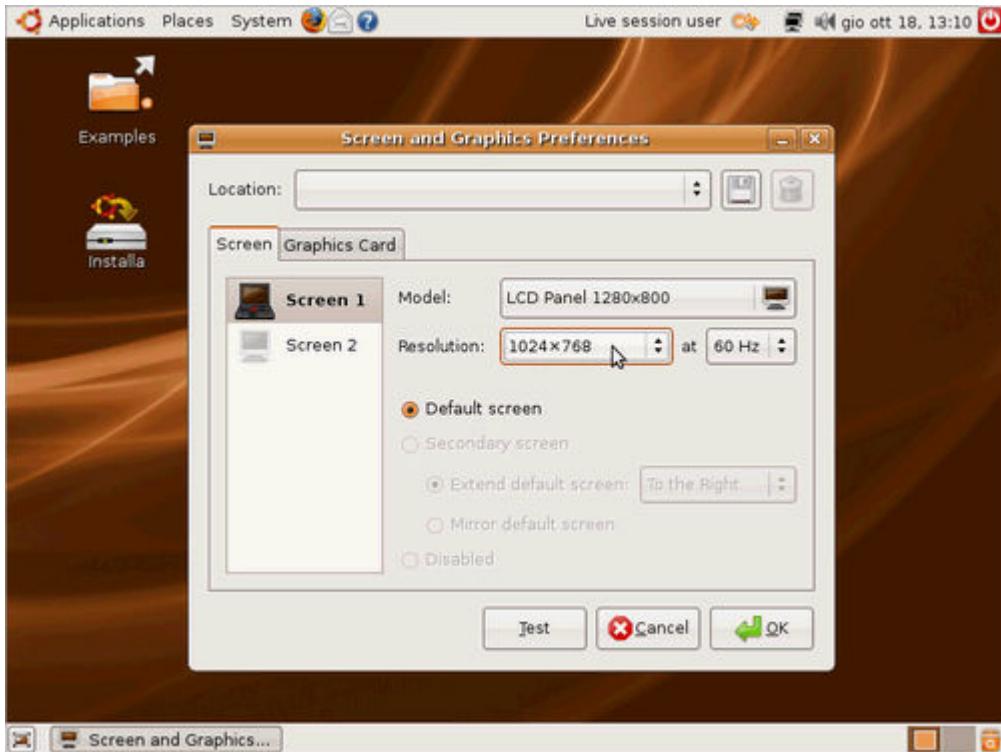


fig. 5

Se compare la schermata di **fig. 6** allora la nuova configurazione va bene, accettate le nuove impostazioni cliccando su Keep configuration. Se invece non dovesse comparire nulla, attendete pazientemente: dopo 15 secondi circa ritornerà tutto come prima con in più un messaggio che avvisa che la nuova configurazione non ha funzionato; ricominciate da **fig.**

3 e ritentate finché non trovate una configurazione che funziona con una risoluzione di almeno 1024x768 pixel.

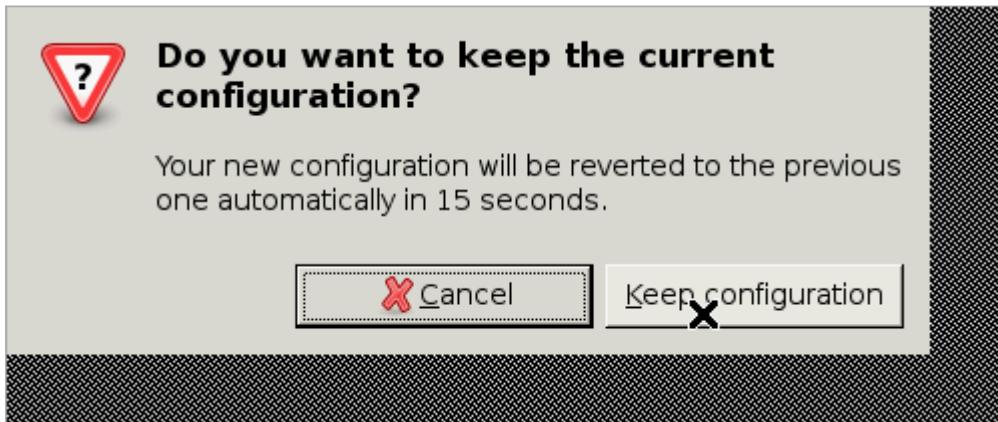


fig. 6

Quando avrete trovato la vostra configurazione corretta e l'avrete accettata, fate molta attenzione: deve assolutamente comparire il messaggio di **fig. 7**. Ho notato che non compare sicuramente quando fate il test della nuova configurazione video cliccando sull'apposito pulsante di **fig. 5**. Quindi, appena trovata una configurazione video che funziona bene con una risoluzione di almeno 1024x768, appuntatevi qual'è e ricominciate un'ultima volta tutta la configurazione da **fig. 3**. Quando arriva il momento di testarla, non fate! Sapete già che funziona dalla volta precedente, cliccate semplicemente sul pulsante OK di **fig. 5**. Comparirà finalmente il messaggio di **fig. 7**, cliccate su Close per proseguire.

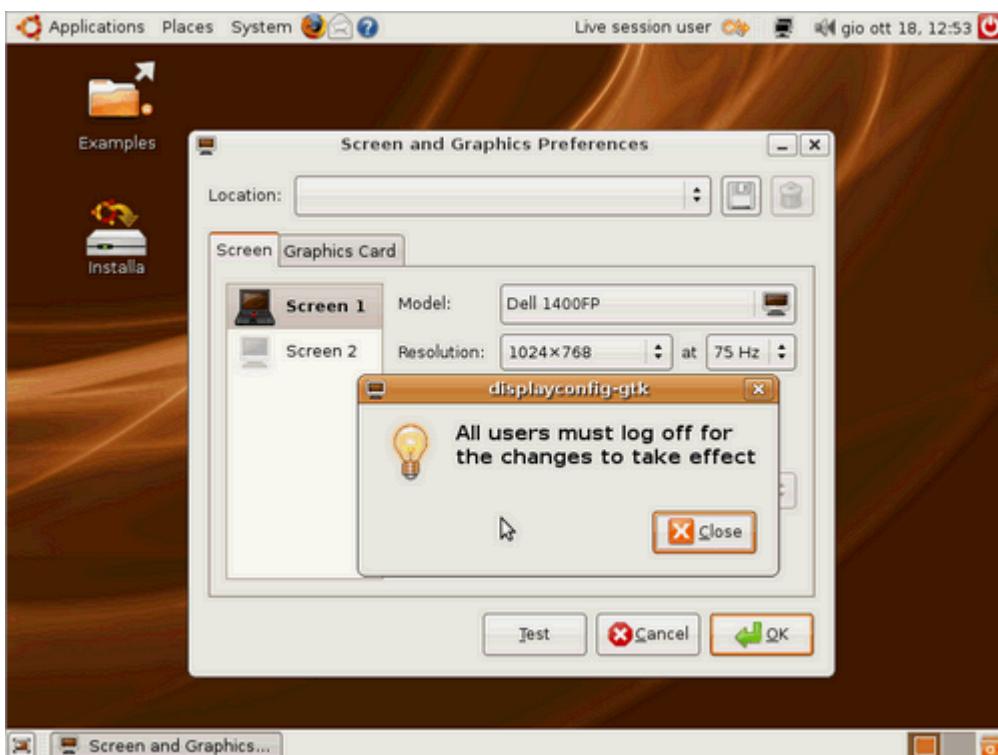


fig. 7

Fate il Log Out come vi è stato chiesto; basta cliccare sul pulsante a forma di interruttore di alimentazione in alto all'estrema destra del desktop di Ubuntu, poi, nella finestra che compare, cliccate su Log Out (**fig. 8**).

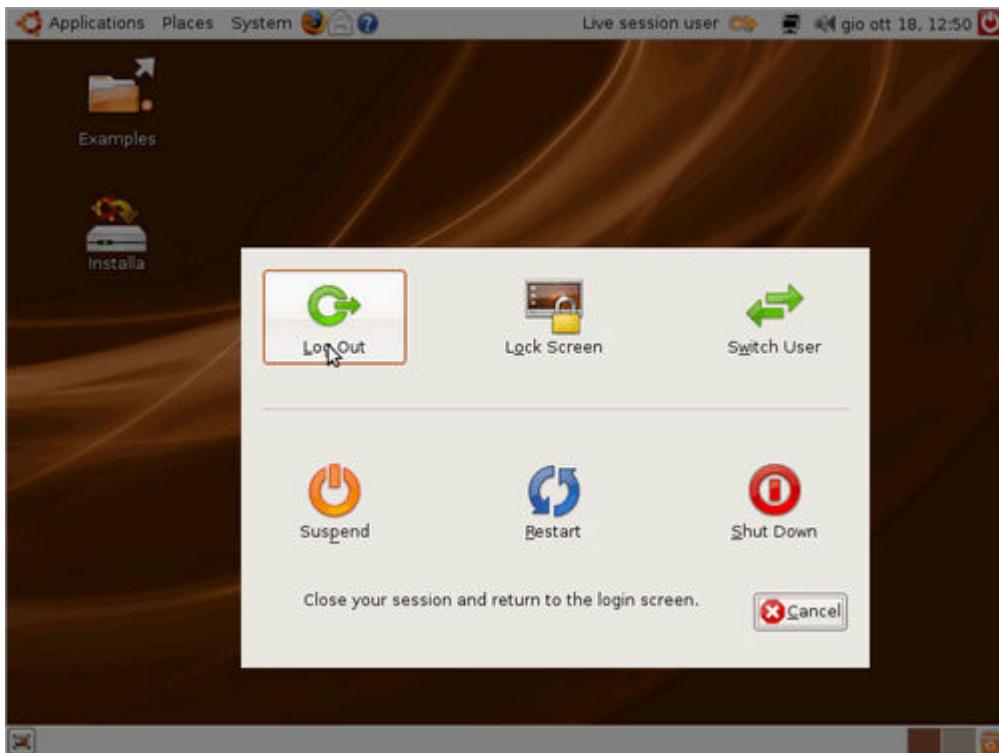


fig. 8

Ripartita la sessione, dopo aver dato il vostro vostro nome utente e password vi ritroverete finalmente in una finestra abbastanza grande da vedere per intero le schermate d'installazione (fig. 9).

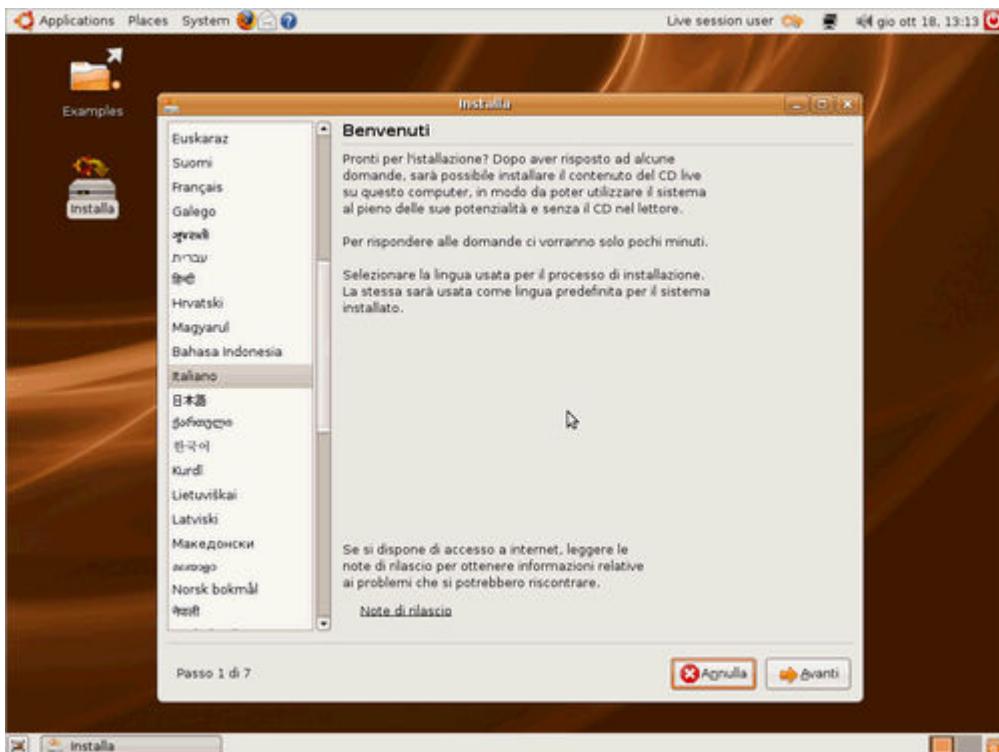


fig. 9

Terminata l'installazione, potreste comunque avere dei problemi con il settaggio di una risoluzione video adeguata, fate riferimento al post "VirtualBox: Guest Additions e risoluzioni video in Ubuntu Linux" per una possibile soluzione.

Io ormai ho fatto, ma se proprio posso darvi un suggerimento, fate un bel salvataggio del file vdi della distribuzione Ubuntu precedente (se non sapete di cosa si tratta, date un'occhiata ai post sulla compattazione del file vdi, se ne parla

abbondantemente) giusto per cautelarsi, poi procedete direttamente al suo upgrade dall'apposito menù Sistema --> Amministrazione --> Gestore aggiornamenti (**fig. 10**).



fig. 10

Ho provato anche questo metodo, giusto perché mi piaceva smanettare anche con questa possibilità. Beh, con una connessione ADSL indispensabile a tirar giù i 1000 e passa pacchetti per quasi 700 MB di aggiornamenti, tempo un'ora e un quarto e la Ubuntu 7.04 era diventata 7.10 senza nessuna rottura di bip!

Solo una cosa: non impauritevi se dopo l'aggiornamento le VirtualBox Guest Additions smettono di funzionare, è normale; basta reinstallarle.

@:)>

Pubblicato alle 7.32 Argomenti: guida, Linux, suggerimenti, tutorial, Virtualbox, virtualizzazione

4 commenti:

Michele Benvegnù ha detto...

Effettivamente io ho utilizzato la seconda strada, quella dell'upgrade direttamente da Ubuntu; anzi, è stato Ubuntu stesso ad accorgersi della presenza dell'aggiornamento ed a propormelo. I tempi sono quelli da te indicati, e tutto è filato perfettamente liscio.

Qualche momento di panico al momento di reinstallare le GuestAdditions, ma dovuto solamente alla mia totale inesperienza di Linux. Infatti, mentre con la 7.04 le GuestAdditions erano localizzate in /cdrom ora con la 7.10 ho dovuto cercarle in /media/VBOXADDITIONS_1.5.0_24068.

Pertanto il comando corretto è:

```
cd /media/VBOXADDITIONS_1.5.0_24068
sudo sh ./VBoxLinuxAdditions.run
```

Addirittura la directory /cdrom è vuota.

Probabilmente il motivo sarà chiaro a tutti i lettori, a me non, ma solo per la mia inesperienza.

Più tardi provo ad installare Mandriva in ViartualBox

19 ottobre 2007 14.46

Giornale di Sistema ha detto...

Ciao Michele.

Lo spostamento del punto di mount del cdrom da /cdrom a /media è una delle ultime tendenze nelle distribuzioni più recenti. Non c'è un vero motivo tecnico, solo scelte su come organizzare le varie cartelle e cosa devono contenere.

Mandriva purtroppo ha dei problemi con la nuova versione 1.5.2 delle Guest Additions. Anche se la loro installazione va a buon fine e riavviando tutto funziona correttamente, allo spegnimento e riaccensione della macchina virtuale tutto torna come se le Guest Addition non fossero installate, mouse che resta intrappolato e tutto il resto. Prova anche tu, magari hai più fortuna. :-)

Ciao ciao.

@:\>

19 ottobre 2007 17.39

Giornale di Sistema ha detto...

Opss... la stanchezza! ;-(

In realtà a fare i capricci è la openSUSE 10.3 e non la Mandriva 2008; chiedo venia.

@:\>

19 ottobre 2007 18.12

Michele Benvegnù ha detto...

Vorrei segnalare che doppo l'aggiornamento alla 7.10, le GuestAdditions erano disabilitate ed ho dovuto reinstallarle; tuttavia Ubuntu è partito con risoluzione precedente, 1280x1024.

Questo significa che le risoluzioni video non dipendono dalle GuestAdditions, ma solo dalla modifica del file xorg.conf?

20 ottobre 2007 19.56

sabato 20 ottobre 2007

VirtualBox 1.5.2: un miglioramento... ma...

Ho installato e testato un po' la nuova versione 1.5.2 di VirtualBox. Devo dire che questa volta la differenza di velocità è davvero tangibile (almeno sul mio portatile).

Le finestre della macchina virtuale, una Mandriva 2008 con KDE e una Ubuntu 7.10 (Gnome), sono davvero più scattanti nella risposta, ad un punto tale che la differenza fra il prima ed il dopo aggiornamento di VirtualBox e delle Guest Additions in ogni macchina virtuale, è visibile ad occhio nudo. Anche una openSUSE 10.3, che con la versione 1.5.0 di VirtualBox si era rifiutata ostinatamente di farsi installare, qui non ha fatto una piega.

Ma... come spesso accade, non è tutto oro ciò che luccica. La openSUSE ad esempio: stavolta sono riuscito ad installarla, poi però ci sono stati problemi con le Guest Additions: una volta installate correttamente e fatta ripartire la macchina virtuale, tutto funziona ma guai a spegnerla. Alla riaccensione si comporta come se le Guest Additions non fossero mai state installate e a nulla vale reinstallarle: ad ogni spegnimento e successiva riaccensione se ne perde completamente traccia.

Nelle Mandriva 2008 si ottiene un messaggio relativo ad un file che non può essere modificato perché manca, solo Ubuntu sembra non faccia i capricci... incrociamo le dita. :-)

@:\>

Pubblicato alle 7.29 Argomenti: [Linux](#), [Virtualbox](#), [VirtualBox Guest Additions](#), [virtualizzazione](#), [Windows](#)

lunedì 22 ottobre 2007

VirtualBox: cartelle condivise in guest Linux (4a parte)

Vi ricordate? Nella 3a parte del post concludevo dicendo che in fondo non è necessario fare il l'auto-mount della cartella condivisa alla partenza del guest Linux, e per la verità è anche meglio non farlo per almeno 3 motivi:

- occorre modificare dei file di configurazione all'interno di Linux da cui è sempre meglio star lontani;
- condividiamo ogni volta una cartella che magari ci serve solo ogni tanto;
- se vogliamo condividere un'altra cartella occorre riandare a cercare quel file, modificare le indicazioni della cartella condivisa al suo interno, salvare, riavviare la macchina virtuale per caricare la nuova cartella condivisa.

Troppo macchinoso?!? Assolutamente! Esiste però un modo non solo più semplice di caricare la cartella condivisa, ma anche più elastico: può essere caricata o scaricata al volo senza riavviare la macchina, e solo quando effettivamente ci occorre. Useremo uno script, un semplice file di testo con all'interno i comandi che abbiamo dato a mano nella shell della 2a parte del post.

Lo faremo usando come esempio una Ubuntu, e per lavorare nella massima comodità, faremo altre 3 cose:

- monteremo la cartella condivisa direttamente in una cartella sul desktop, così da averla sempre immediatamente a disposizione senza dover vagare nel filesystem di Linux alla sua ricerca;
- creeremo un ulteriore script per smontare sempre al volo la cartella (umount);
- creeremo un paio di pulsanti sul pannello di Gnome (l'interfaccia grafica di Ubuntu), quello in alto con sopra i vari menù di Ubuntu per intenderci, da cui avviare direttamente gli script.

Iniziamo creando la cartella sul desktop in cui montare la cartella condivisa: cliccate direttamente sul desktop con il tasto destro e poi su *Crea cartella* con il tasto sinistro (fig. 23), datele un nome, per esempio *Cartella condivisa host*, e salvatela.

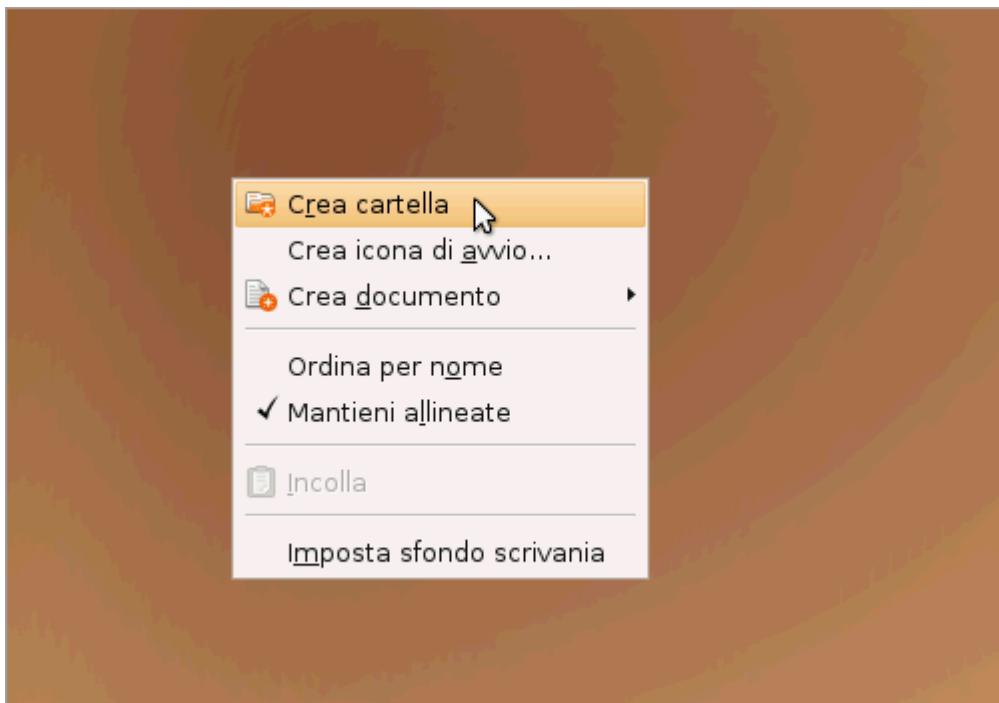


fig. 23

E' il turno dei 2 file che conterranno gli script di *mount* e *umount* (smontaggio) della cartella condivisa. Aprete l'*Editor di testo* (si tratta di gedit, fig. 24),

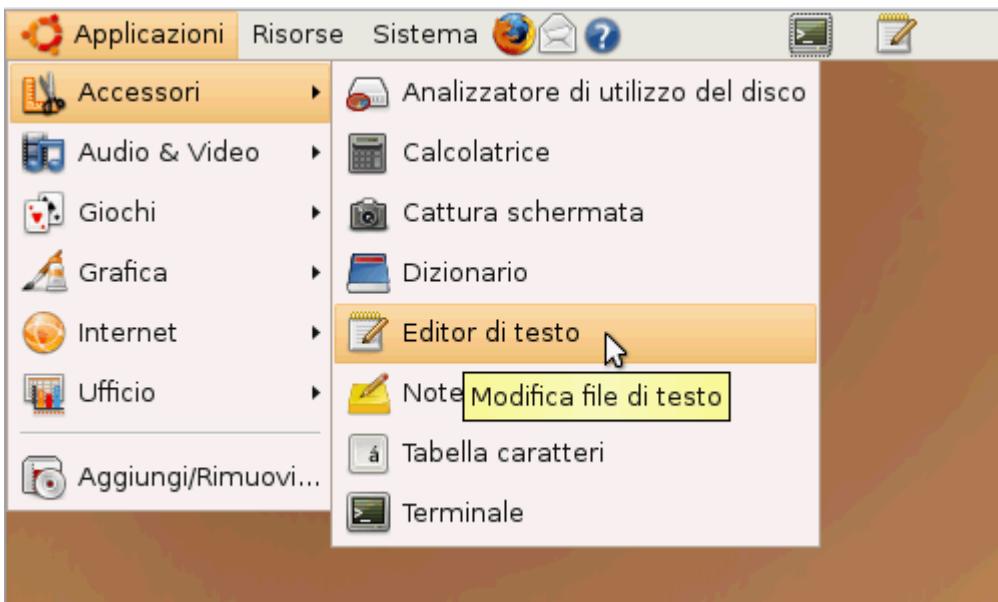


fig. 24

e al suo interno scrivete le 2 seguenti righe (fig. 25):

```
#!/bin/bash
sudo mount -t vboxsf 'Cartella condivisa host' /home/utente/'Cartella condivisa host'
```

Poi salvate il tutto nella vostra home con il nome *mountcc* (senza tanta fantasia, sta per Mount Cartella Condivisa); di default l'editor si dovrebbe essere aperto proprio al suo interno. Verificate comunque, alla voce *Salva nella cartella* di fig. 25 deve esserci il vostro nome utente.

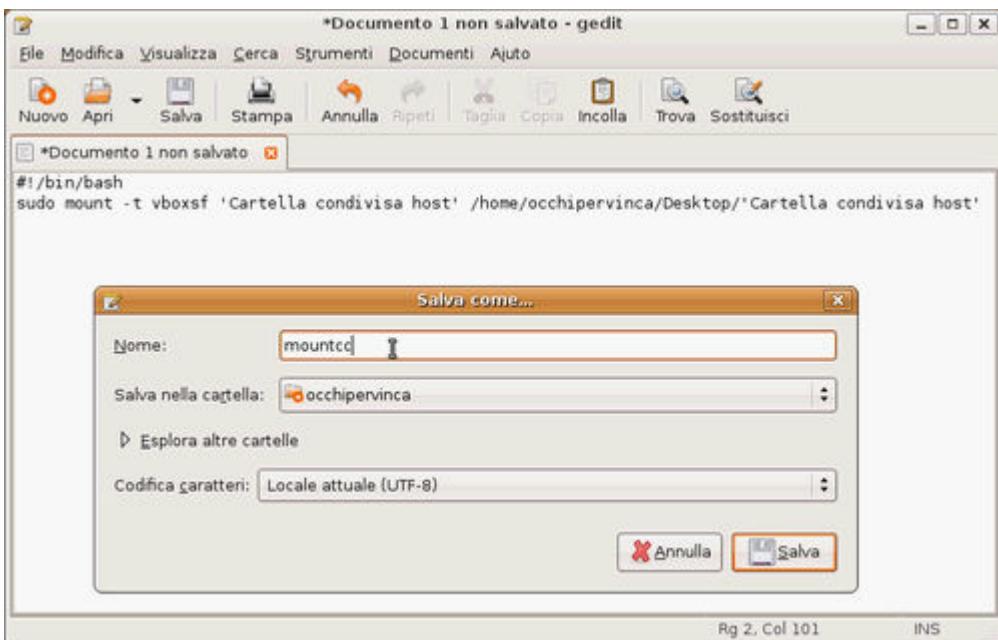


fig. 25

Create un secondo documento nell'editor, inserite le 2 righe sotto,

```
#!/bin/bash
sudo umount 'Cartella condivisa host'
```

e salvate il documento sempre nella vostra home con il nome di *umountcc*. Vediamo un attimo cosa abbiamo scritto. La prima riga in comune ad entrambi gli script, `#!/bin/bash`, avvisa il sistema che:

- (#!) tutto quanto segue è un script di shell (o terminale che dir si voglia, ma d'ora in poi diremo shell);

- la shell da eseguire si trova nella directory `/bin` in un file che si chiama `shell`;

La seconda riga invece chiama con i privilegi di amministratore (`sudo`):

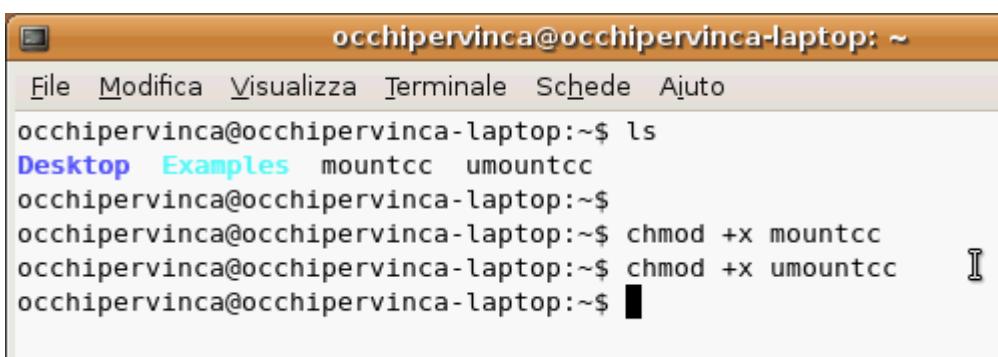
- il comando di `mount` (spiegato nella parte iniziale della 2a parte del post) nel primo script per montare la cartella condivisa,
- il comando di `umount` (senza la `n`) per smontare la cartella condivisa.

Il motivo per cui vi ho chiesto di salvarli nella vostra home è presto detto: quando aprite la finestra di terminale che fra poco useremo, questa di default si apre proprio sulla vostra home. In questo modo potrete semplicemente aprire il terminale e dare subito i comandi `mountcc` e `umountcc` senza doverli andare a cercare da qualche parte.

Bene, gli script sono creati, ma i file che li contengono, così come sono verrebbero letti come normali file di testo; dobbiamo renderli eseguibili. Aprite il terminale (guardate fig. 24, è l'ultima voce del menù) e date il comando `ls` (LiSt, analogo a `dir` per Windows) per vedere il contenuto della vostra home: eccoli lì i 2 file di script, `mountcc` e `umountcc` (fig. 26). Sempre nella finestra di terminale date i comandi:

```
chmod +x mountcc  
chmod +x umountcc
```

Se tutto è andato bene, il comando vi restituirà ogni volta semplicemente il prompt mentre entrambi i file sono diventati eseguibili.



occhipervinca@occhipervinca-laptop: ~

```
File Modifica Visualizza Terminale Schede Ajuto  
occhipervinca@occhipervinca-laptop:~$ ls  
Desktop Examples mountcc umountcc  
occhipervinca@occhipervinca-laptop:~$  
occhipervinca@occhipervinca-laptop:~$ chmod +x mountcc  
occhipervinca@occhipervinca-laptop:~$ chmod +x umountcc  
occhipervinca@occhipervinca-laptop:~$
```

fig. 26

Proviamo i due script. Dall'interno del terminale date il comando `mountcc`: dopo aver dato la password da amministratore, andate a dare un'occhiata in *Cartella condivisa host* sul desktop: et voilà! Il contenuto della cartella condivisa sull'host ora è visibile direttamente sul desktop di Linux. :-D

Vi piace? Sii?!? Ma non abbiamo mica finito qua. Sempre nel terminale, ora date il comando `umountcc`: la cartella condivisa così viene smontata, ed il suo contenuto scompare. ;-)

Bene, constatato che gli script funzionano, diamo un tocco di eleganza al tutto: niente lancio a mano degli script (cosa comunque sempre possibile nel modo che abbiamo visto), ma due bei pulsanti su cui cliccare per lancerli.

Cliccate con il tasto destro sul pannello di Gnome (fig. 27) e poi con il sinistro sulla voce *Aggiungi al pannello*:



fig. 27

Nella finestra che si è aperta, cliccate su *Icona avvio applicazioni personalizzata* (fig. 28).



fig. 28

Fate le opportune selezioni e riempite i vari campi in modo tale da ottenere un risultato analogo a fig. 29; in particolare nel campo *Comando*, tramite il tasto accanto *Esplora*, cercate e selezionate lo script *mountcc*. Cliccate anche su *Nessuna icona* se volete associare al vostro pulsante una icona, quando avete finito su *OK* per terminare.



fig. 29

Ed eccola, sul vostro pannello (fig. 30) è comparsa l'icona del pulsante da cui lanciare la script di mount della cartella condivisa. Ripetete tutti i passaggi per creare un analogo pulsante di lancio dello script di umount.



fig. 30

A questo punto non vi resta che provarli. Aprite *Cartella condivisa host* sul desktop e cliccate sul pulsante di mount: si aprirà la finestra di terminale dove inserire la password da amministratore (le operazioni di mount e umount sono operazioni riservate all'amministratore se non diversamente specificato). Inserita e dato invio, vedrete comparire in tempo reale il contenuto della cartella condivisa sull'host. Cliccando sull'altro pulsante e data la password se richiesta, vedrete scomparire davanti ai vostri occhi il contenuto della cartella condivisa.

Bene, e con questo credo abbiamo finito anche l'argomento cartelle condivise. In una mia riflessione "sotto la doccia" qualche post addietro, descrivevo la virtualizzazione come un ponte fra sistemi operativi, e più in generale fra idee, concezioni, modi di essere, in definitiva fra persone. Vedo le cartelle condivise come l'esemplificazione pratica e immediata in VirtualBox di questo ponte.

E allora, buona condivisione a tutti! :-D
@:\>

Pubblicato alle 12.18 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

martedì 23 ottobre 2007

VirtualBox: struttura e backup delle macchine virtuali

In questi ultimi post, fra compattazione dei dischi rigidi ad espansione dinamica, condivisione di cartelle, sia per guest Windows che guest Linux, smanettamenti vari che continueremo a vedere nei prossimi post, è arrivato il momento di cominciare a prendere qualche precauzione, caso mai Murphy fosse nei paraggi. ;-)

Personalmente ho già perso un'intera installazione Linux mentre smanettavo per capire come compattarne il disco rigido virtuale. Vedremo perciò quel tanto che basta della struttura e dell'organizzazione dei file di VirtualBox sull'host, per capire dove mettere mano e salvare le nostre preziose macchine virtuali con tutte le loro configurazioni.

Questa organizzazione è per fortuna molto semplice. Di default VirtualBox crea una cartella .VirtualBox in C:\Documents and Settings\NomeUtente\ (fig. 1). Al suo interno ci sono: un file xml su cui sono salvate le impostazioni di VirtualBox stesso, e 2 sottocartelle, Machines e VDI su cui concentreremo un attimo la nostra attenzione.

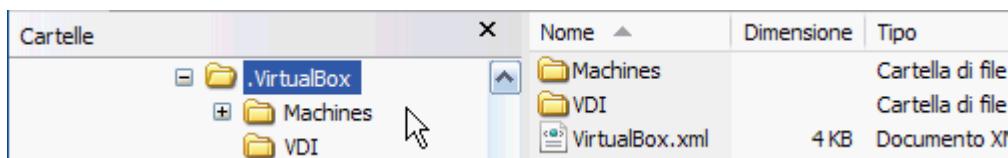


fig. 1

Machines è la cartella che contiene le nostre macchine virtuali. Al suo interno troverete tante sottocartelle con lo stesso nome e nello stesso numero delle macchine virtuali (fig.2).

Prendiamo come esempio la cartella Ubuntu 7.04. Le configurazioni della macchina virtuale, ossia sistema operativo installato, memoria RAM e scheda video, disco o dischi rigidi collegati alla macchina virtuale e così via, sono salvate nel file NomeMacchinaVirtuale.xml.

La cartella Logs contiene invece i log di VirtualBox relativi alla macchina virtuale.

La cartella Snapshot è molto interessante: è qui che vengono salvati gli snapshot che sono un po' come i punti di ripristino di Windows. In realtà VirtualBox fa il furbetto e non salva realmente un'immagine dell'intera macchina virtuale, ma questo lo vedremo in un altro post.

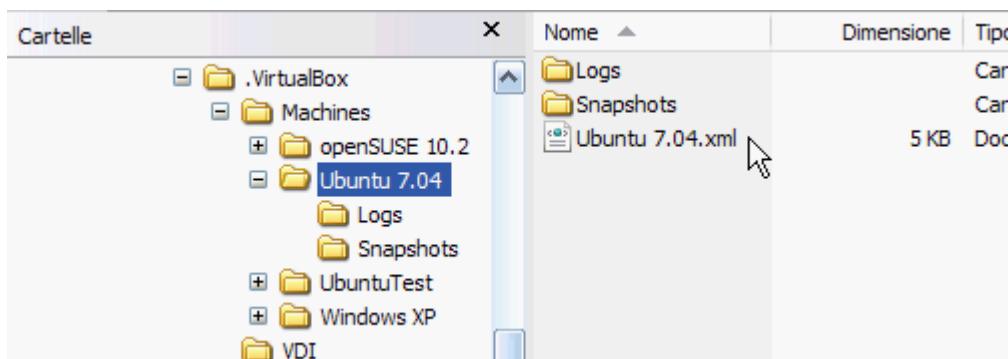


fig. 2

Se Machines contiene la macchina virtuale, la cartella VDI contiene il suo o i suoi dischi rigidi (per un massimo di 3): notate infatti come la macchina virtuale UbuntuTest ne abbia 2, UbuntuTest.vdi e UbuntuTest2.vdi.

E' qui dunque che viene salvato il file più importante: all'interno del file vdi è contenuto infatti tutto il disco rigido della macchina virtuale e quindi tutto il sistema operativo che vi è installato sopra. Ed è sempre questo il file di cui ci stiamo occupando nella serie di post dedicata alla compattazione di dischi rigidi ad espansione dinamica, citata all'inizio del post.

Cartelle	Nome	Dimensione	Tipo
- .VirtualBox	openSUSE10_2.vdi	6.063.185 KB	File
- Machines	Ubuntu7_04.vdi	3.656.785 KB	File
+ openSUSE 10.2	UbuntuTest2.vdi	81.941 KB	File
+ Ubuntu 7.04	UbuntuTest.vdi	3.667.025 KB	File
+ UbuntuTest	WindowsXP.vdi	1.445.972 KB	File
+ Windows XP			
+ VDI			

fig. 3

Per mettere al sicuro le nostre macchine virtuali, possiamo allora fare così: copiare da qualche altra parte, l'ideale sarebbe un'unità di backup esterna, per esempio uno dei tanti dischi rigidi USB da collegare all'host, l'intera cartella .VirtualBox. Successivamente, ad ogni modifica importante del sistema operativo guest, per esempio l'aggiornamento da Ubuntu 7.04 a 7.10, salvataggio del solo file vdi della macchina virtuale interessata.

Naturalmente questa è una soluzione come dire, casalinga. Facile, immediata, senza tanti fronzoli, il suo limite più grande è che la macchina virtuale da salvare deve essere spenta. In un ambito home questo non dovrebbe essere un problema, ma in futuro non è da escludere qualche altro post sull'argomento per descrivere modalità di salvataggio un po' più sofisticate.

@:)>

Pubblicato alle 8.38 Argomenti: guida, Linux, suggerimenti, tutorial, Virtualbox, virtualizzazione, Windows

giovedì 25 ottobre 2007

Compattazione dischi rigidi guest Linux: ci siamo

Buone nuove dal fronte compattazione dischi rigidi virtuali ad espansione dinamica: proprio ieri ho terminato le prove ed ora anche i guest Linux hanno la loro compattazione al pari dei guest Windows! :-D

Non mi resta che esporvi il tutto: si tratta solo di compilare il programma dal sorgente dopo aver installato tutte le dipendenze necessarie, lavorare come amministratore in modalità monoutente da linea di comando, smontare e rimontare filesystem... pauuraa eehhh?!?

Su su, tranquilli, sono qui apposta per mostrarvelo con semplicità e tante immagini; solo un pò di pazienza e sono da voi. Nel frattempo che cambio anche il titolo ai post già pubblicati, per renderlo più consono al contenuto (perciò se dovesse ricomparire nel vostro lettore di feed non ci fate caso), approfittatene per salvare la vostra macchina virtuale. Stavolta ci avvicineremo molto più del solito al cuore di Linux e lo faremo con grande cautela, ma prevenire è sempre meglio che curare. ;-)

@:\>

Pubblicato alle 9.52 Argomenti: [Linux](#), [news](#), [Virtualbox](#), [virtualizzazione](#), [Windows](#)

venerdì 26 ottobre 2007

VirtualBox: su, sudo e cartelle condivise in guest Linux

Prologo: questo post riguarda **tutte le distribuzioni Linux** che usano il comando su invece che sudo come in Ubuntu per eseguire operazioni riservate all'amministratore di sistema. Come distribuzione di esempio è stata usata Mandriva 2008.

Dicevo qualche post fa che Mandriva è una delle distribuzioni che più mi piacciono fra quelle che non uso. Visti però i recenti problemi di installazione della openSUSE 10.3 in VirtualBox, ho deciso di farne la distribuzione da affiancare alla Ubuntu quando ci sono da fare esempi concreti nei post.

Una delle cose che inizialmente mi ha messo in difficoltà nell'usarla, è stata la differente gestione della modalità amministrativa rispetto ad Ubuntu. In poche parole Mandriva usa la modalità classica con il comando su che cambia la nostra utenza trasformandoci direttamente nell'amministratore; Ubuntu invece usa il comando sudo che eleva momentaneamente i nostri privilegi a quelli dell'amministratore restando però chi siamo.

Il comando sudo ha una serie di vantaggi rispetto a su: permette infatti un controllo molto più fine su quali attività amministrative l'utente può fare (scelte sempre dall'amministratore), usando direttamente la password utente: l'amministratore in questo modo non deve esporre la sua password ma può comunque concedere a determinati utenti la possibilità di fare cose che solo lui potrebbe. Ubuntu in realtà è andata oltre, disabilitando del tutto l'amministratore e usando solo il comando sudo per le attività amministrative; risulta comunque sempre possibile riattivare su con il comando sudo passwd root.

A seconda che la distribuzione usi il comando su oppure il comando sudo il comportamento della stessa cambia. Chi come Michele Benvegnù ha provato infatti a seguire quanto indicato nel post relativo alla condivisione cartelle su guest Linux, si è sicuramente accorto che le cose con Mandriva non funzionano. Vediamo allora **come attivare il comando sudo** per tutte le distribuzioni che, come Mandriva, di default usano il comando su.

Il file da modificare si trova in /etc e si chiama sudoers. Aprite la shell, in Mandriva e tutte le distribuzioni con l'interfaccia grafica KDE, si chiama Konsole (**fig. 1**), quando l'interfaccia grafica è Gnome si chiama Terminale (l'abbiamo visto in molti altri post con Ubuntu dove appunto si usa Gnome).

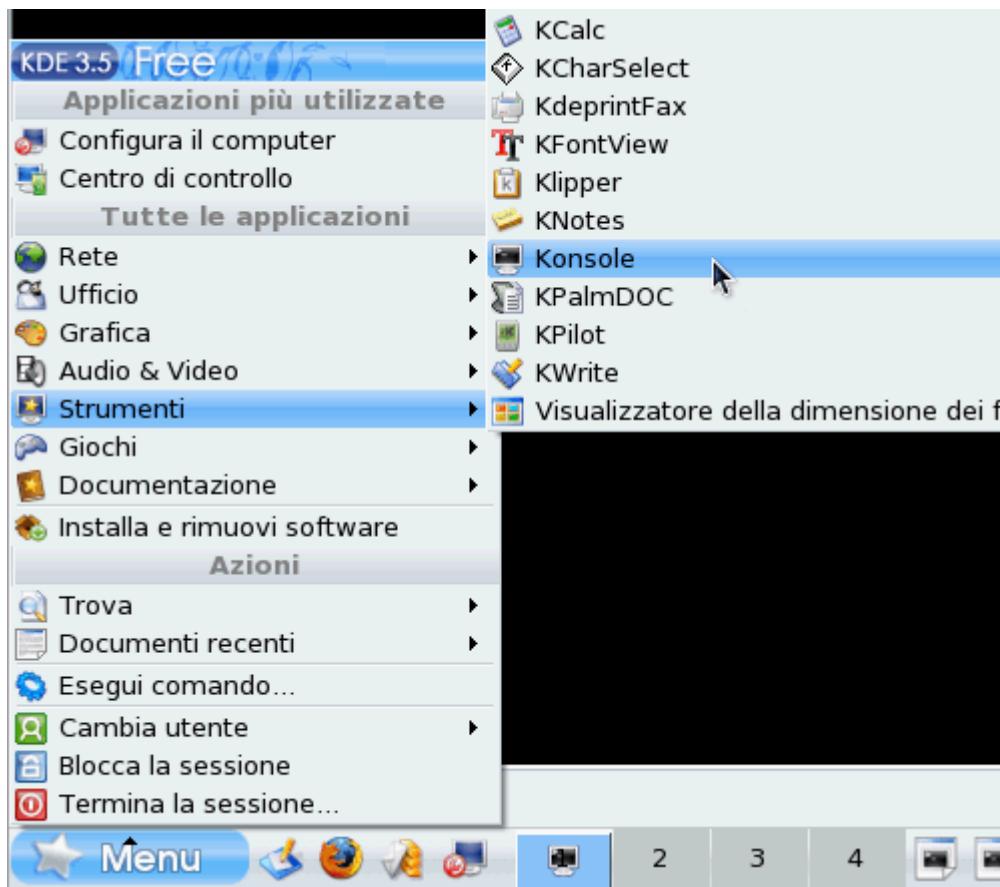


fig. 1

Ora date i seguenti comandi (**fig. 2**):

```
cd /etc [invio]  
su [invio]
```

Con questi comandi vi siete spostati nella directory etc e siete diventati amministratore (necessario per modificare il file sudoers altrimenti immodificabile per un utente normale).

Potete notare anche visivamente la differenza fra utente normale e amministratore: se guardate la forma del prompt, noterete che prima di dare il comando su, quindi utente normale, il prompt terminava sempre con \$, dopo essere diventati amministratore la riga termina sempre con #. Quando non sapete se siete utente normale o amministratore, guardate l'ultimo carattere della riga: \$ utente normale, # amministratore.

Ora date il comando:

```
visudo [invio]
```

fig. 2

Il contenuto del file sudoer si apre all'interno di un editor (**fig. 3**), fate molta attenzione: come avvisa anche la 3a riga del file, occorre **usare esclusivamente il comando visudo per modificare sudoers**. Questo perché quando salverete le modifiche, vengono eseguiti ulteriori controlli per evitare che errori di sintassi nel file taglino fuori tutti gli utenti del sistema.

Ora fate ancora più attenzione: l'editor in cui vi trovate funziona come vi (pronuncia vi ai): come utenti Windows non assomiglia a nulla di ciò che conoscete, perciò vi prego di fare esattamente come vi dico.

Prima cosa da sapere: ogni riga che inizia con il simbolo cancelletto #, è un commento (ogni file di configurazione di Linux è così, è una sua caratteristica generale, ricordatela).

Seconda cosa: vi, l'editor, ha due modalità di funzionamento: appena entrati, come in questo momento, è in modalità comando. Se scrivete una lettera adesso (non lo fate, per carità!), questa non comparirebbe come testo ma verrebbe interpretata come comando, perciò buoni e fermi, non premete nulla di più di quanto vi dico.

Ok, ora portate con i tasti freccia il cursore d'inserimento al punto dove lo vedete in **fig 3** (il rettangolo chiaro sullo sfondo nero sul lato sinistra della schermata). Questo perché è in quel punto che vogliamo inserire del testo.

```
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers
#
# Host alias specification
#
# User alias specification
#
# Cmnd alias specification
#
# Defaults specification
#
# Runas alias specification
#
# User privilege specification
root    ALL=(ALL) SETENV: ALL

# Uncomment to allow people in group wheel to run all commands
# and set environment variables.
# %wheel      ALL=(ALL) SETENV: ALL
#
# Same thing without a password
# %wheel      ALL=(ALL) NOPASSWD: SETENV: ALL

# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now
~
```

fig. 3

Premete il ":" (due punti) sulla tastiera. Vedrete il quadratino schizzare verso il fondo della schermata e lì comparirà il ":". Aggiungete la lettera "i" (**fig. 4**) e date invio.

fig. 4

Sia il ":" che la "i" scompariranno ma il cursore resterà sul posto. Siete entrati in modalità inserimento, e tutto ciò che da questo momento scrivete viene inserito come testo nel punto in cui si trovava inizialmente il cursore (ricordate? Nello spazio vuoto sotto la prima riga visibile in **fig. 4**). Bene, ora con la massima attenzione, date un invio, e anche se sembra non sia successo nulla, scrivete:

VostraUser ALL=(ALL) SETENV: ALL

dove al posto di "VostraUser" mettete il nome della vostra user, quella che usate per entrare nel sistema. Date un altro invio, poi premete il tasto Esc sulla tastiera. Dopo qualche attimo l'editor uscirà dalla modalità inserimento tornando alla modalità comando, e ciò che avete inserito comparirà magicamente al suo posto (**fig. 5**, la riga che inizia per gds, la mia user).

```
# %wheel      ALL=(ALL) SETENV: ALL

gds      ALL=(ALL) SETENV: ALL

# Same thing without a password
# %wheel      ALL=(ALL) NOPASSWD: SETENV: ALL

# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now
~
```

fig. 5

A questo punto salvate il tutto e uscite dall'editor con il comando ":wq" e dando invio.

Se per qualche motivo dovreste sbagliare, non allarmatevi: tornate in modalità comando con Esc e date il comando ":q!" seguito da invio. Uscirete dall'editor senza salvare, rientrate con il comando visudo e riprendete d'accapo.

Bene, se avete fatto tutto correttamente, ora il comando sudo è attivo anche per la vostra user. A questo punto non vi resta che seguire il post sulle cartelle condivise in guest Linux e vedrete che questa volta tutto funzionerà correttamente anche su Mandriva e le altre distribuzioni.

@:>

Pubblicato alle 18.10 Argomenti: amministrazione di sistema, guida, Linux, tutorial, Virtualbox, virtualizzazione

lunedì 29 ottobre 2007

VirtualBox: compattare dischi virtuali ad espansione dinamica guest Linux (2a parte)

Come dicevo? "Ok, è tutto..."? E invece no. La soluzione proposta nella 1a parte del post era infatti si molto immediata, usava direttamente comandi Linux disponibili in tutte le distribuzioni, ma scarsamente efficiente, soprattutto nella gestione dello spazio disco sull'host. Vediamo questa volta una soluzione che usa un programma dedicato, *zerofree*: non può rivaleggiare in facilità con la precedente soluzione ma è estremamente più efficiente.

Zerofree ed è il frutto del lavoro di Ron Yorston e di un successivo miglioramento di *the_camel*; colgo l'occasione per ringraziarli entrambi. Proprio come *SDelete* per i guest Windows, anche *zerofree* va a cercare tutti i blocchi liberi del disco rigido virtuale e li riempie con il valore zero. Funziona su filesystem EXT2 ed EXT3 (un filesystem è il tipo di organizzazione dei file sul disco rigido, Windows usa NTFS) che è la scelta di default della stragrande maggioranza delle distribuzioni Linux, e sicuramente della Ubuntu e della Mandriva che useremo come esempio.

In questo post ci occuperemo in particolare di Ubuntu; Mandriva la vedremo successivamente come esempio valido per qualunque distribuzione. Salvate le vostre macchine virtuali e iniziamo.

Andate sulla pagina di Ron Yorston e scaricate il file *zerofree-1.0.1.tgz*. Potete farlo direttamente dal guest dove lo dovremo usare, oppure dall'host e poi tramite la condivisione cartelle copiarlo nel guest Linux. Portate per comodità il file sul desktop e osservate (**fig. 1**): si tratta di un file *.tgz*, un formato compresso tipico di Linux (simile ai file *jar* di Java per fare un esempio) contenente il sorgente del programma: dobbiamo decomprimere e compilarne il contenuto.

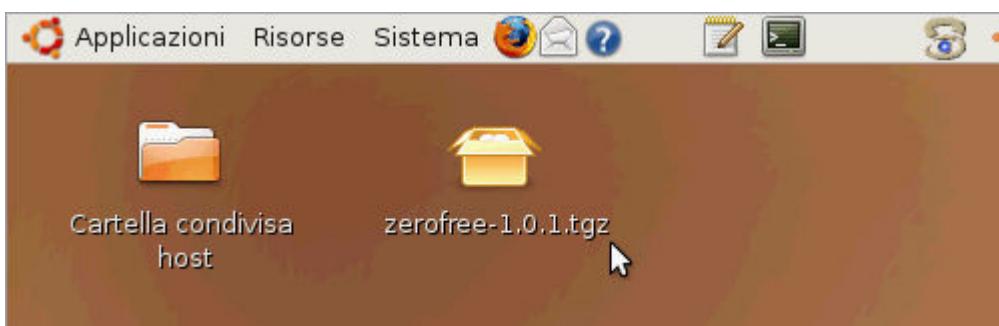


fig. 1

Aprite la shell (*Applicazioni* --> *Accessori* --> *Terminale*) e portatevi nella directory del Desktop con il comando

```
cd Desktop
```

Attenzione al fatto che tutte le distribuzioni più recenti hanno tradotto Desktop con Scrivania, perciò se il comando non funziona verificate che non sia il vostro caso. Ora date il comando:

```
tar xvzf zerofree-1.0.1.tgz
```

Verrà creata la directory *zerofree-1.0.1* come sottodirectory di Desktop (**fig. 2**), ed infatti comparirà la relativa icona sul Desktop.

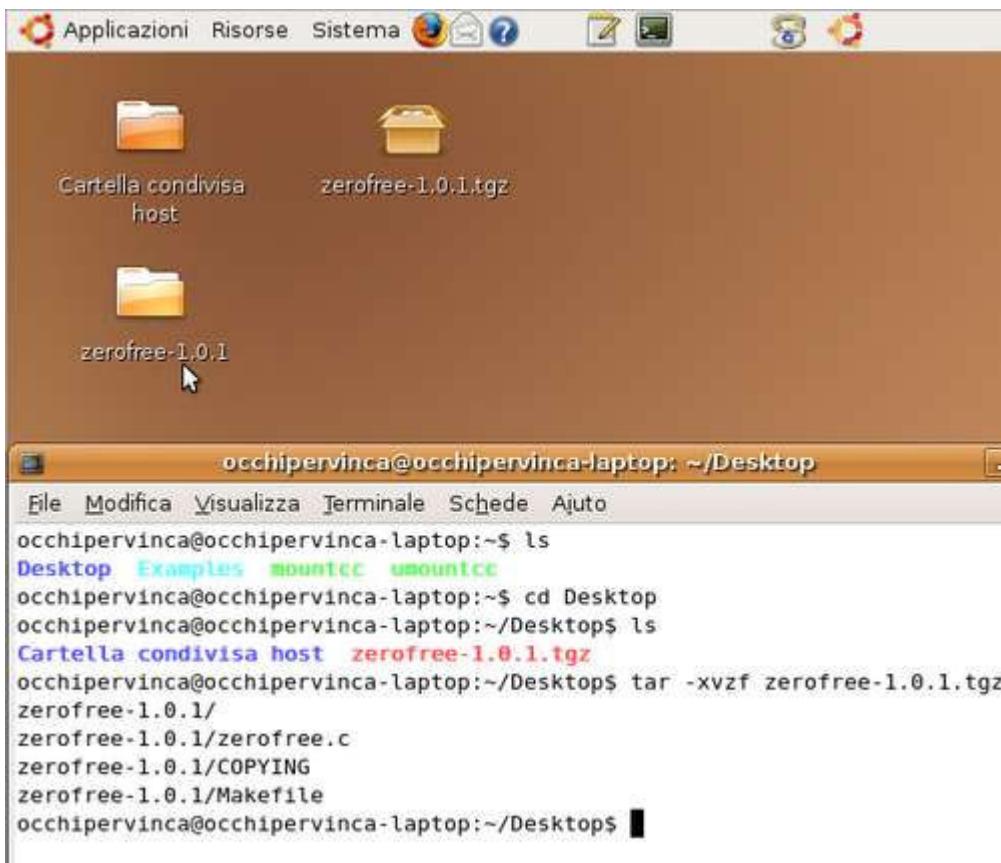


fig. 2

Ora entriamo nella nuova directory con il comando:

```
cd zerofree-1.0.1.
```

Se date il comando `ls` per vederne il contenuto, troverete i due file che ci interessano: `zerofree.c`, il sorgente C del programma che dovremo compilare, e `Makefile`, contenente le istruzioni per il compilatore (il programma che a partire dal sorgente del programma crea il file eseguibile).

Ubuntu però (ed anche altre distribuzioni), nella sua installazione di default non ha gli strumenti essenziali alla compilazione di un programma. No problem: provvediamo subito. Visto che siamo già nella shell, approfittiamone per fare da qui l'installazione del pacchetto *build-essential* che contiene ciò che cerchiamo:

```
sudo apt-get install build-essential
```

Accettate gli ulteriori pacchetti che eventualmente vi indicherà come necessari e proseguite. Se non amate lavorare dalla shell, o magari volete dare un'occhiata all'installer grafico di Ubuntu, *Synaptic*, allora *Sistema --> Amministrazione --> Gestore pacchetti Synaptic* (fig. 3).



fig. 3

Con il comando *Cerca* in alto a destra (fig. 4) usate la stringa "*build-essential*" per cercare e selezionare il pacchetto d'installazione corrispondente, poi cliccate su *Applica* per installare il tutto. Accettate l'eventuale software aggiuntivo che vi verrà segnalato (le cosiddette *dipendenze*) e proseguite.

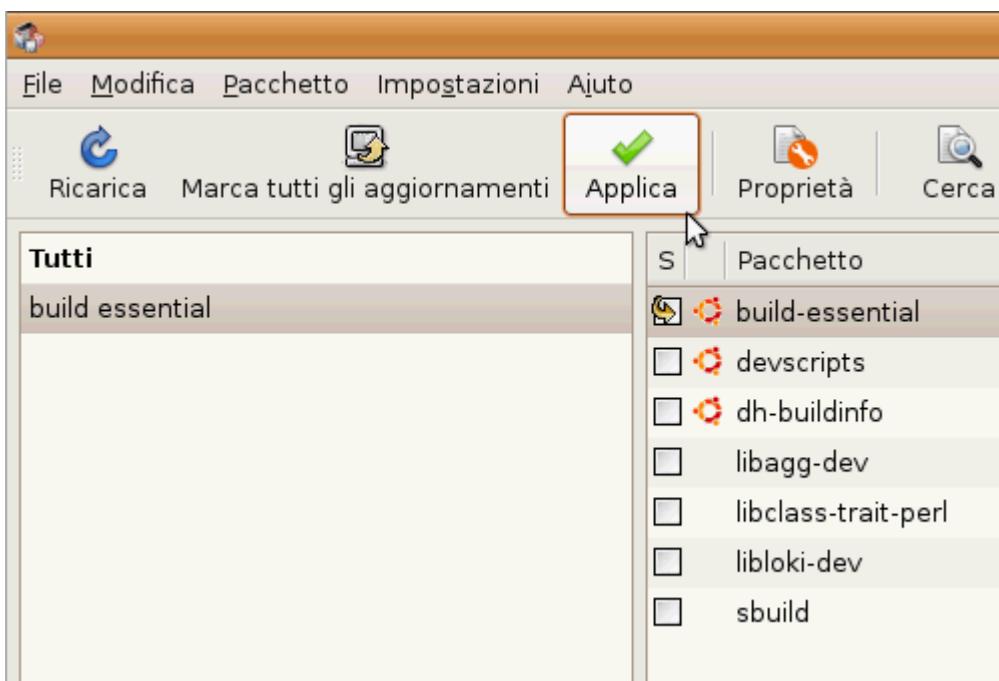


fig. 4

Ma non basta: manca ancora una libreria, la *ext2fs*, usatela come stringa in ricerca nel comando *Cerca* ed installate

anche questa (il nome esatto è e2fslibs-dev, **fig. 5**).

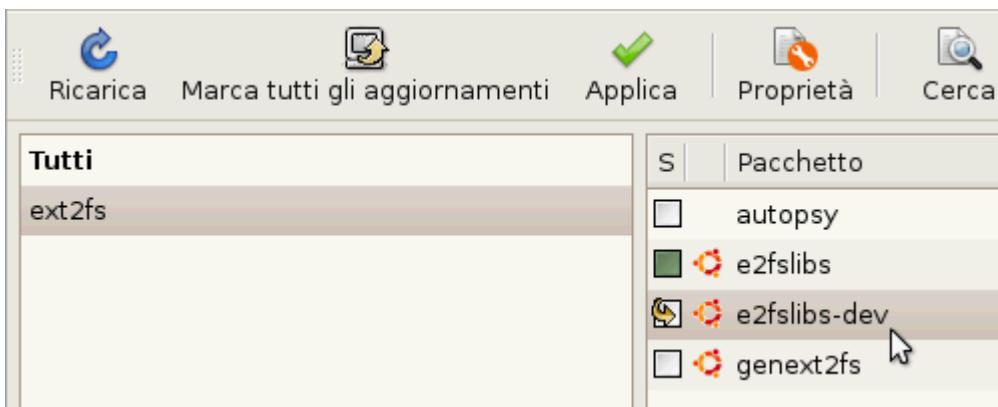


fig. 5

Ok, ora siamo davvero pronti alla compilazione di zerofree. Dalla shell date il comando:

make

Tutto qua; pochi istanti ed il programma sarà compilato. Se date un ls nella shell, lo vedrete nella lista dei file contenuto all'interno della directory in cui vi trovate, oppure potete vederne l'icona dal file manager (**fig. 6**).



fig. 6

Benissimo, non ci resta che copiarlo nella cartella root dell'amministratore visto che è appunto uno strumento di amministrazione:

sudo cp zerofree /root

Che dite, ci vogliamo fermare un pò a riprendere fiato? Il prossimo post sull'argomento sarà davvero molto intenso... e tutto da linea di comando. ;-)

@:)>

Pubblicato alle 12.08 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

martedì 30 ottobre 2007

VirtualBox: compattare dischi virtuali ad espansione dinamica guest Linux (3a parte)

Ok, ripreso fiato? Bene, ora fate un altro bel respiro profondo. Se la 2a parte del post era faticosina, questa che è interamente da linea di comando lo è ancora di più. Ma siamo finalmente in dirittura d'arrivo, non vorrete fermarvi proprio adesso che siete ad un passo dal compattare i dischi rigidi ad espansione dinamica anche su guest Linux, vero? Su, forza, iniziamo. ;-)

Avete ancora la shell aperta? No? Riapritela, e preparatevi: porteremo Linux in modalità monoutente. Quasi che fosse un vecchio DOS, in modalità monoutente Linux non accetta più di un utente, è scollegato dalla rete, non ha interfaccia grafica, insomma, l'ideale per non avere rotture di scatole quando dobbiamo fare lavori di amministrazione molto profondi all'interno della macchina. Da shell date il comando

sudo telinit 1

che porterà la vostra macchina al runlevel 1. I runlevel in Linux specificano cosa deve essere in funzione e cosa no per ognuno di essi; portarsi a runlevel 1 permette appunto di rendere la macchina monoutente, tirar via tutto ciò che non serve per la sua amministrazione (interfaccia grafica, rete...) e lavorarci in tranquillità. Se tutto è andato bene, vi ritroverete in una schermata simile a quella di **fig. 8**. Date il comando

clear

per dare una bella ripulita allo schermo, e cominciate a guardarvi intorno: noterete, se guardate l'ultima lettera del prompt (#), che in questo momento siete amministratore della macchina.

```
root@occhipervinca-laptop:~# df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/hda1        19734388  2689356  16042580  15% /
varrun          257992       52  257940   1% /var/run
varlock          257992       0  257992   0% /var/lock
procbususb      257992       68  257924   1% /proc/bus/usb
udev            257992       68  257924   1% /dev
devshm          257992       0  257992   0% /dev/shm
lrm             257992     33788  224204  14% /lib/modules/2.6.20-16-g
eric/volatile
root@occhipervinca-laptop:~# mount -n -o remount,ro -t ext2 /dev/hda1 /
root@occhipervinca-laptop:~#
root@occhipervinca-laptop:~# fsck -f /dev/hda1
fsck 1.40-WIP (14-Nov-2006)
e2fsck 1.40-WIP (14-Nov-2006)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hda1: 120255/2506752 files (0.4% non-contiguous), 751014/5012272 blocks
root@occhipervinca-laptop:~#
root@occhipervinca-laptop:~# cd /root
root@occhipervinca-laptop:~#
root@occhipervinca-laptop:~# ./zerofree /dev/hda1_
fig. 7
```

Ora date il comando:

df

Df permette di visualizzare dimensioni, spazio libero ed utilizzato di una data partizione, dato senza nessun argomento come sopra, ci mostra i dati di tutte le partizioni. In **fig. 8** potete vedere il tipico risultato di un'installazione standard di Ubuntu.

La riga che ci interessa è la prima (la terza nello screenshot): nella prima colonna dice che esiste un dispositivo (/dev = device) che corrisponde al vostro disco rigido (/hda1), dopo un po' di dati sullo spazio occupato o meno, ci dice che è montato all'inizio del filesystem (estrema destra, la colonna "Montato su" e sotto "/").

Non entro in ulteriori dettagli, vado al sodo: i dati che ci interessano sono la prima e l'ultima colonna, "/dev/hda1" e "/". Quello che ora faremo è rimontare il vostro disco rigido in sola lettura, indispensabile per fare le operazioni che ci occorrono senza far danni. Date il comando:

mount -n -o remount,ro -t ext2 /dev/hda1 /

che tradotto significa: fai il mount, senza aggiornare il file mstab (-n), facendo il remount in sola lettura (remount,ro) come se fosse un filesystem di tipo EXT2 (-t ext2), del dispositivo /dev/hda1, rimontandolo in /, dove cioè si trovava all'inizio (l'ultima colonna del comando df, ricordate?). Per migliorare la leggibilità, in **fig. 8** ho aggiunto io un riga vuota fra un comando e l'altro, se tutto va bene il comando in realtà non restituisce nulla. Ora date il comando:

```
fsck -f /dev/hda1
```

farete così un bel *FileSystemCheck*, un controllo approfondito del vostro disco, per assicurarvi che tutto è a posto. Se riscontrasse dei problemi e vi propone dei rimedi, accettateli, a meno di essere dei "filesystem guru" non riusciremmo a fare meglio di quanto propone. Quando ha terminato ed è tutto ok, date un bel

```
cd /root
```

per portarvi nella home dell'amministratore. Ricordate? E' lì che abbiamo messo *zerofree*: è arrivato il momento di usarlo. Per eseguirlo:

```
./zerofree /dev/hda1
```

Il comando dice a Linux di eseguire il programma *zerofree* cercandolo nella directory corrente (./) e di passare al programma il disco su cui agire (/dev/hda1). Sembrerà che non accada nulla ma in realtà il programma sta lavorando intensamente; guardate l'icona di attività dei dischi rigidi virtuali, in basso a destra sulla barra di stato della finestra di VirtualBox che contiene il guest, e ve ne accorgerete.

Quando *zerofree* avrà terminato il suo lavoro ricomparirà semplicemente il prompt: ridate il comando *fsck -f /dev/hda1* per essere sicuri dell'integrità del disco rigido, infine spegnete la macchina virtuale con il comando:

```
halt
```

Sul guest è tutto.

Ora che le zone libere del disco rigido virtuale sono state segnate con valori zero da *zerofree*, sull'host non vi resta che dare il solito comando

```
"C:\Programmi\innotek VirtualBox\VBoxManage" modifyvdi "filename".vdi compact
```

ed avete terminato del tutto (per maggiori informazioni sul comando, leggete la fase 4 della 1a parte del post). Ricordate che i doppi apici che racchiudono *filename* servono solo se *filename* contiene degli spazi, altrimenti potete ometterli.

Anche per questa volta abbiamo finito. Per questa volta! Infatti nel prossimo post sull'argomento vedremo come fare tutto ciò per una distribuzione Mandriva e più in generale per una qualunque distribuzione Linux. Nei due post dedicati ad Ubuntu infatti, ho tirato via dritto sulle eventuali difficoltà incontrate indicandovi preventivamente cosa fare per evitarle.

Con Mandriva non faremo così: mi mostrerò quali sono i messaggi di errore che ho incontrato usando un'installazione di default della distribuzione, in fase di compilazione del programma *zerofree* per esempio, oppure i messaggi di *fsck* quando incontra un problema, e come farci guidare da questi alla ricerca della soluzione. Al prossimo post sull'argomento allora; a presto.

```
@:\>
```

Pubblicato alle 10.24 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

lunedì 5 novembre 2007

VirtualBox: compattare dischi virtuali ad espansione dinamica - guest Linux (4a parte)

Nella 3a parte del post sull'argomento avevamo visto come compilare `zerofree`, il programma per mettere a zero i blocchi liberi del disco rigido della macchina virtuale Linux, cosa indispensabile per poter poi procedere alla sua compattazione con il comando `modifyvdi` di VirtualBox. Come esempio avevamo usato una distribuzione Ubuntu, ed il programma così ottenuto dalla compilazione è già pronto per essere usato in qualunque altra distribuzione.

E se non avete installata una Ubuntu come fare a compilare il programma? Proprio questo è l'argomento del post: *come compilare zerofree in una generica distribuzione Linux*, e quali sono le eventuali difficoltà che potete incontrare sia nella compilazione che nel suo uso. Distribuzione di esempio, una Mandriva. Accendete la vostra macchina virtuale e iniziamo.

Un'avvertenza: quando lavoreremo nella shell, questa volta al posto di immagini troverete direttamente il copia-incolla del testo, così se non avete voglia di digitarvelo a manina (cosa sempre consigliata per dare un taglio più concreto alle cose) potete semplicemente copiarlo dal post e incollarlo nella vostra shell. Inoltre il risultato del comando è preceduto sempre da "\$".

Do per scontato che abbiate già scaricato sul vostro desktop il sorgente del programma (fate riferimento alla 3a parte del post per i particolari), ora aprite la finestra della shell (si chiama *Konsole* se state usando l'interfaccia grafica KDE, *Terminal* se usate Gnome). Date i seguenti comandi per portarvi nella directory del vostro Desktop (tradotto in Scrivania nelle distribuzioni più recenti) e visualizzarne il contenuto:

```
cd Scrivania  
ls
```

```
$ Cartella condivisa host/ media.desktop trash.desktop zerofree-1.0.1.tgz Home.desktop register.desktop  
upgrade.desktop
```

Ecco lo il nostro sorgente: `zerofree-1.0.1.tgz`.

E' in formato tgz, detto più comunemente "tarball", un formato tipico di Linux ottenuto in 2 fasi: raccogliendo più file in un unico file tramite il comando tar e poi sottoponendolo a compressione tramite gzip. Decomprimiamolo con il comando:

```
tar -xvzf zerofree-1.0.1.tgz
```

```
$ zerofree-1.0.1/  
$ zerofree-1.0.1/zerofree.c  
$ zerofree-1.0.1/COPYING  
$ zerofree-1.0.1/Makefile
```

Come potete vedere, ha creato una directory `zerofree-1.0.1` (la seconda riga) e dentro ha creato tre file: `zerofree.c` è il sorgente C del programma che poi compileremo, `Makefile` contiene le istruzioni che vengono passate al compilatore quando daremo il comando `make` di compilazione. Portiamoci nella nuova directory e guardiamone il contenuto:

```
cd zerofree-1.0.1  
ls
```

```
$ COPYING Makefile zerofree.c
```

Eccoli là i nostri tre file; diamo il comando `make` di compilazione:

```
make
```

```
$ bash: make: command not found
```

Come avevo preannunziato, questa volta non vi avrei risparmiato gli errori che si ottengono quando si prova a compilare `zerofree`, ed ecco il primo: il programma `make` (ricordate dal post sull'architettura di base di Linux? In Linux tutto è un programma, anche i comandi) non esiste (`make: command not found`), dobbiamo installarlo. In Mandriva si tratta di usare *Install e rimuovi software* (fig. 1), ma ogni distribuzione Linux ha qualcosa di analogo, cercate e avviate questa funzione.

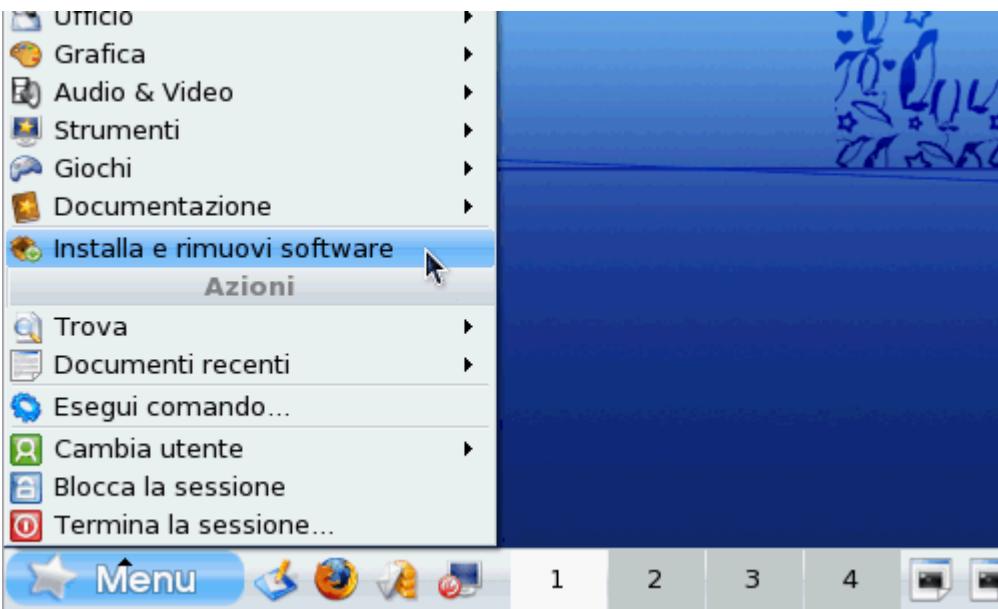


fig. 1

Usando il box *Cerca* (e l'analogo in altre distribuzioni) cerchiamo nei nomi la stringa "make": spuntate il relativo box e installatelo (fig. 2).

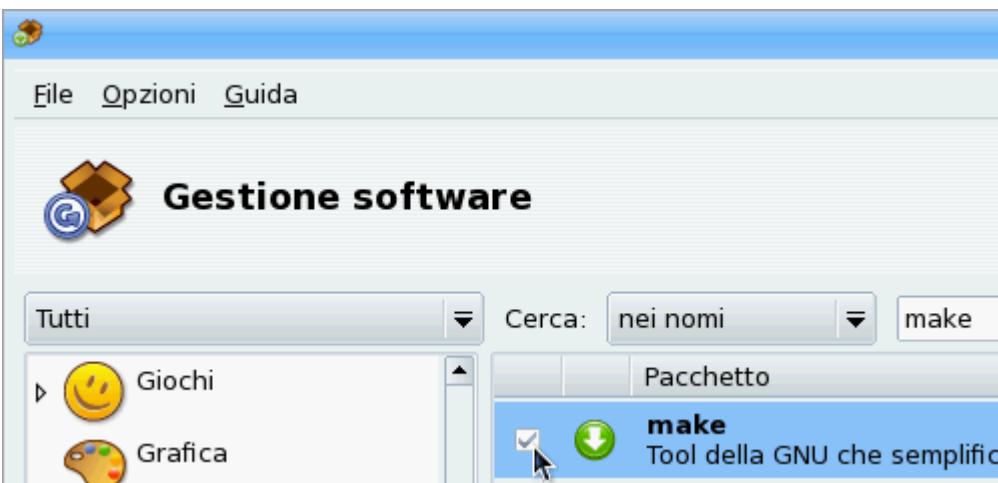


fig. 2

Lasciate aperta questa finestra e tornate alla shell, qui ridate il comando *make*:

make

```
$ gcc -o zerofree -lext2fs zerofree.c
$ make: gcc: Command not found
$ make: *** [all] Error 127
```

Ora *make* c'è ed infatti compaiono nuove righe. La terza riga in particolare, quella subito sotto al comando *make*, lancia il compilatore (*gcc*) con una serie di parametri, peccato che *gcc* manchi (*make: gcc: Command not found*): installiamolo. Tornate nuovamente alla finestra di *Gestione software* e questa volta cercate con la stringa "gcc"; spuntate ed installate il relativo pacchetto (fig. 3).

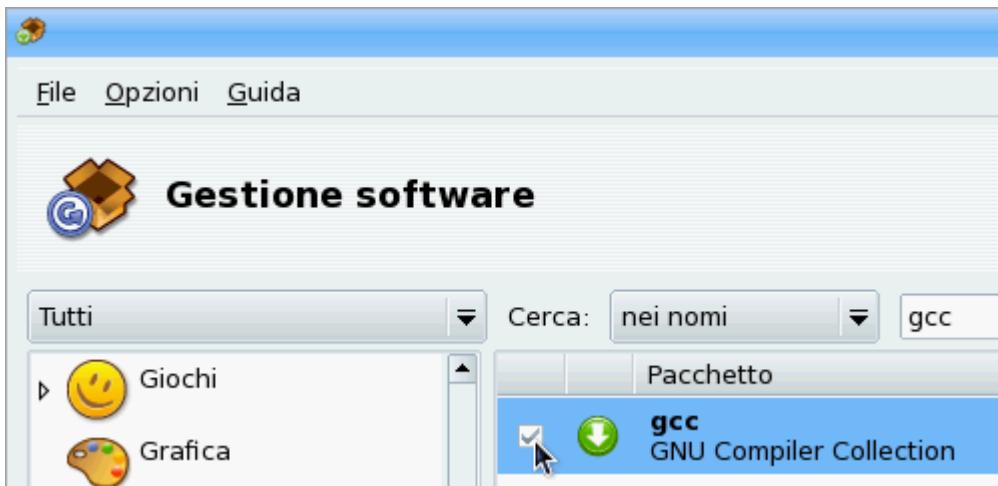


fig. 3

Lasciate la finestra aperta e tornate alla shell, ridate nuovamente il comando *make*:

```
>>gcc -o zerofree -lext2fs zerofree.c
$ zerofree.c:15:27: error: ext2fs/ext2fs.h: No such file or directory
$ zerofree.c: In function 'main':
$ zerofree.c:24: error: 'errcode_t' undeclared (first use in this function)
$ zerofree.c:24: error: (Each undeclared identifier is reported only once
$ zerofree.c:24: error: for each function it appears in.)
$ zerofree.c:24: error: expected ';' before 'ret'
[...]
```

Niente, errore anche questa volta, ma guardate con attenzione la terza riga: l'errore è cambiato. Ora cerca qualcosa che si chiama *ext2fs* e *ext2fs.h*. Torniamo alla finestra di *Gestione software* e cerchiamo il termine "ext2fs" (fig. 4). Quando trovate più pacchetti che rispondono alla stessa stringa di ricerca, tenete presente che stiamo cercando di compilare un'applicazione: il pacchetto di supporto alla compilazione nel nome ha sempre qualcosa come "dev" oppure "devel" (development = sviluppo, ricordatelo perché è una convenzione usata in Linux). Spuntate il pacchetto giusto e installatelo.

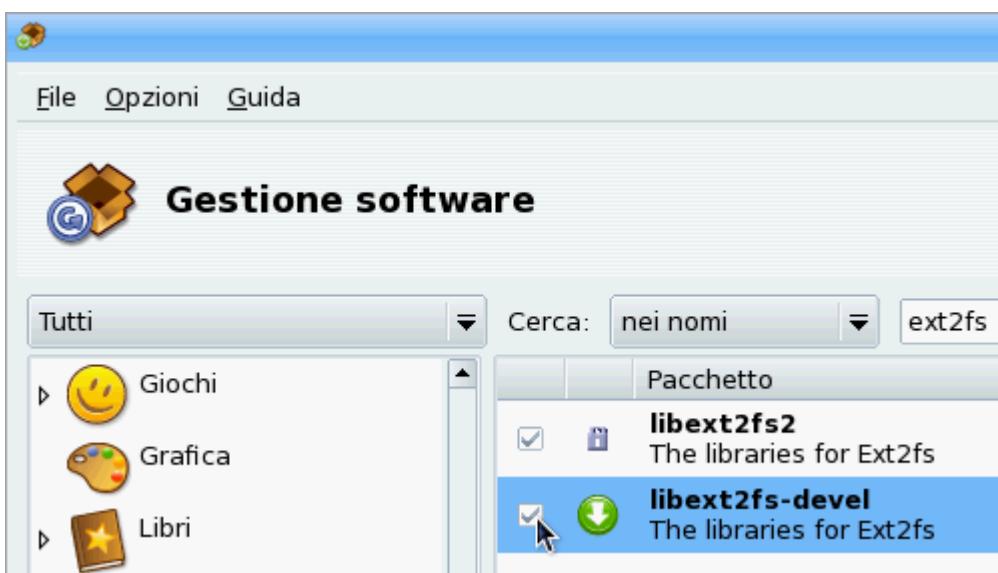


fig 4

Terminata l'installazione ritornate alla shell ridando il comando *make*:

```
make
$ gcc -o zerofree -lext2fs zerofree.c
```

ls

COPYING Makefile zerofree* zerofree.c

Bravissimi!! La compilazione è andata a buon fine, ed il comando ls per vedere il contenuto della directory in cui vi trovate ci dice che esiste un nuovo file (ora sono quattro in tutto), *zerofree*, dove l'asterisco indica si tratta di un file eseguibile e non fa parte del nome. Copiamo adesso il programma nella directory /root , la home dell'amministratore:

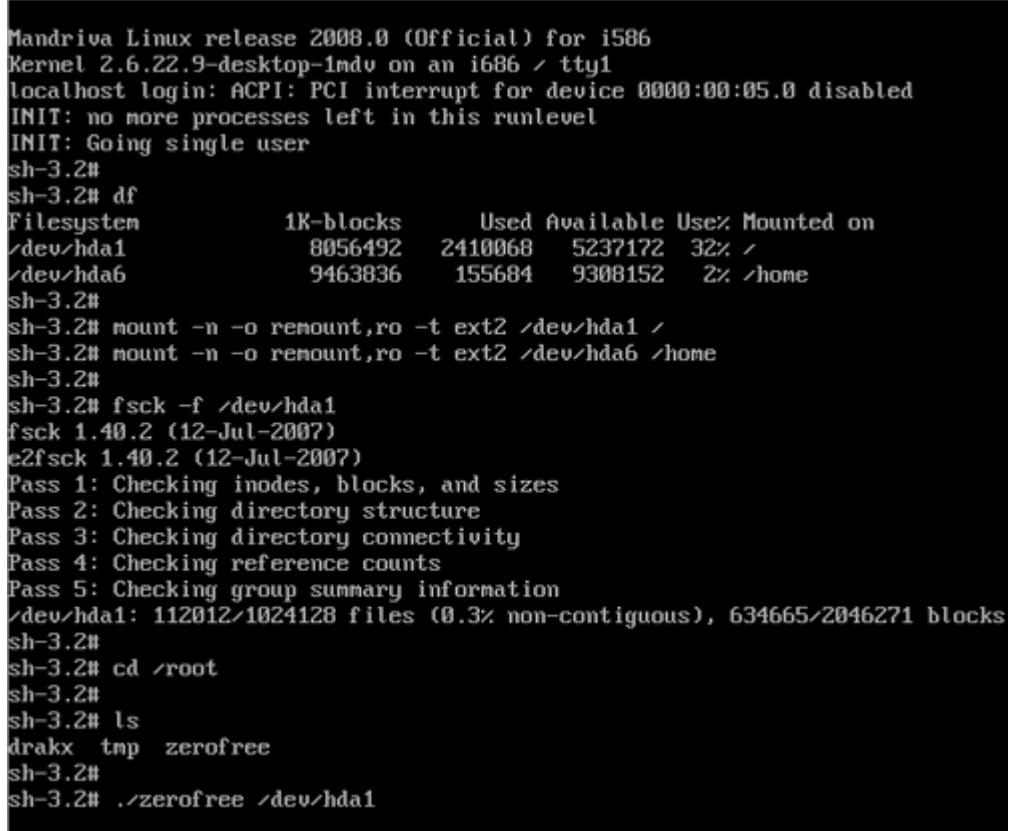
```
sudo cp zerofree /root  
$ Parola d'ordine:
```

chiudete ogni finestra aperta e nella shell date il comando:

```
sudo telinit 1
```

In questo modo portate Linux al runlevel 1. I *runlevel* in Linux specificano cosa deve essere in funzione e cosa no per ognuno di essi; portarsi a runlevel 1 permette di rendere la macchina monoutente, tirar via tutto ciò che non serve per la sua amministrazione (interfaccia grafica, rete...) e lavorarci in tranquillità.

Se tutto è andato bene, vi ritroverete in una schermata simile a quella di **fig. 5**, dove ho dato un invio fra un comando e l'altro per aumentare la leggibilità. Cominciate con il notare che in questo momento siete amministratori della macchina (indicato dall'ultima lettera del prompt, # per l'amministratore e \$ per l'utente normale), perciò attenti a quello che fate, potreste segarvi via l'intera distribuzione (non è un'esagerazione, può accadere con il comando giusto nel posto sbagliato).



Mandriva Linux release 2008.0 (Official) for i586
Kernel 2.6.22.9-desktop-1mdv on an i686 / tty1
localhost login: ACPI: PCI interrupt for device 0000:00:05.0 disabled
INIT: no more processes left in this runlevel
INIT: Going single user
sh-3.2#
sh-3.2# df
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/hda1 8056492 2410068 5237172 32% /
/dev/hda6 9463836 155684 9308152 2% /home
sh-3.2#
sh-3.2# mount -n -o remount,ro -t ext2 /dev/hda1 /
sh-3.2# mount -n -o remount,ro -t ext2 /dev/hda6 /home
sh-3.2#
sh-3.2# fsck -f /dev/hda1
fsck 1.40.2 (12-Jul-2007)
e2fsck 1.40.2 (12-Jul-2007)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hda1: 112012/1024128 files (0.3% non-contiguous), 634665/2046271 blocks
sh-3.2#
sh-3.2# cd /root
sh-3.2#
sh-3.2# ls
drakx tmp zerofree
sh-3.2#
sh-3.2# ./zerofree /dev/hda1

fig. 5

Ora, sempre guardando **fig. 5**, date il comando:

```
df
```

df permette di visualizzare dimensioni, spazio libero ed utilizzato di una data partizione, dato senza nessun argomento come sopra, ci mostra i dati di tutte le partizioni. In **fig. 5** potete vedere come in Mandriva (e in tante altre distribuzioni) le cose siano un po' diverse da Ubuntu (**fig. 8** della 3a parte del post per un confronto): le partizioni qui sono 2 e non una sola.

Nella prima riga della prima colonna infatti leggiamo che esiste un dispositivo (/dev = device) che corrisponde alla prima partizione del vostro disco rigido virtuale (/hda1), dopo un po' di dati sullo spazio occupato o meno, ci dice che è montato

all'inizio del filesystem (estrema destra, la colonna "Mounted on" è sotto "/").

Ma nella seconda riga della prima colonna leggiamo che esiste un altro dispositivo (/dev = device) che corrisponde all'altra partizione del vostro disco rigido virtuale (/hda6, perchè salti da a1 ad a6 lo si capirà in successivi post dedicati a Linux), e sempre dopo un po' di dati sullo spazio occupato o meno, ci dice che è montato su /home (estrema destra, la colonna "Mounted on" è sotto "/home"). In altre parole questa partizione contiene la vostra home.

La maggior parte delle distribuzioni, di default fanno proprio così: mettono su una partizione il sistema operativo, e su un'altra le home degli utenti. E' un po' come se sotto Windows sul disco C: installassimo solo Windows ed i programmi, e su D: invece la cartella Documenti (una prassi molto seguita in ambito aziendale). Che sia Windows o Linux, questa separazione è di grandissima utilità: potete in qualunque momento procedere ad una reinstallazione del sistema operativo senza la preoccupazione di perdere tutti i documenti che sono su una partizione separata (un salvataggio comunque è sempre consigliabile).

Torniamo a noi. Quello che ora faremo è *rimontare le 2 partizioni in sola lettura*, operazione indispensabile per procedere successivamente senza far danni. Date i comandi:

```
mount -n -o remount,ro -t ext2 /dev/hda1  
mount -n -o remount,ro -t ext2 /dev/hda6 /home
```

che tradotto significa: fai il mount, senza aggiornare il file mstab (-n), facendo il remount in sola lettura (remount,ro) come se fosse un filesystem di tipo EXT2 (-t ext2), del dispositivo /dev/hda1, rimontandolo in /; analogamente per il dispositivo /dev/hda6 rimontandolo in /home. Ora date il comando:

```
fsck -f /dev/hda1
```

farete così un bel *FileSystemCheck* sulla prima partizione, un controllo approfondito per assicurarvi che tutto sia a posto prima di iniziare. Se riscontrasse dei problemi e vi propone dei rimedi, accettateli; a meno di essere dei "filesystem guru" non riuscireste a fare meglio di quanto propone. Quando ha terminato ed è tutto ok, date un bel

```
cd /root
```

per portarvi nella home dell'amministratore. Ricordate? E' lì che abbiamo messo zerofree: è arrivato il momento di usarlo.

Per eseguirlo:

```
./zerofree /dev/hda1
```

Il comando dice a Linux di eseguire il programma zerofree cercandolo nella directory corrente (./) e di passare al programma la partizione su cui agire (/dev/hda1). Sembrerà che non accada nulla ma in realtà il programma sta lavorando intensamente; guardate l'icona di attività dei dischi rigidi virtuali, in basso a destra sulla barra di stato della finestra di VirtualBox che contiene il guest, e ve ne accorgerete. Quando zerofree avrà terminato il suo lavoro ricomparirà semplicemente il prompt: ridate il comando

```
fsck -f /dev/hda1
```

per essere sicuri dell'integrità della partizione, e se tutto è ok abbiamo terminato con la prima partizione.

Ripetiamo gli stessi passi per l'altra partizione:

```
fsck -f /dev/hda6
```

Qui fate attenzione: *fsck* si potrebbe lamentare come in **fig. 6**. In realtà il filesystem è montato in sola lettura proprio per evitare danni, per cui date y e proseguite.

```
WARNING!!! Running e2fsck on a mounted filesystem may cause  
SEVERE filesystem damage.  
Do you really want to continue (y/n)?
```

fig. 6

Quando ha terminato:

```
./zerofree /dev/hda6
```

e poi ancora

```
fsck -f /dev/hda6
```

per essere sicuri che tutto sia a posto, infine spegnete la macchina virtuale con il comando:

```
halt
```

Sul guest è tutto.

Ora che le zone libere del disco rigido virtuale sono state riempite di valori zero, sull'host non vi resta che dare il solito comando

```
"C:\Programmi\innotek VirtualBox\VBoxManage" modifyvdi "nomefile".vdi compact
```

ed avete terminato del tutto (per maggiori informazioni sul comando, leggete la parte finale della 1a parte del post). Ricordate solo che i doppi apici che racchiudono nomefile servono solo se nomefile contiene degli spazi, altrimenti potete ometterli.

Beh, direi proprio che potete fermarvi qui soddisfatti di quello che avete fatto. Intanto non avrete più problemi con dischi rigidi ad espansione dinamica che sanno solo espandersi: anche sotto Linux potrete comprimerli quando vi pare e con la massima efficacia. Non solo: avete imparato anche qualcosa in più sul funzionamento di Linux; ma qui, siamo solo all'inizio. ;-)

```
@:\>
```

Pubblicato alle 15.14 Argomenti: guida, Linux, software, tutorial, Virtualbox, virtualizzazione, Windows

venerdì 9 novembre 2007

VirtualBox: installazione di Ubuntu 7.10, suggerimenti

Nel post "VirtualBox e Ubuntu 7.10: matrimonio possibile" vi avevo raccontato delle difficoltà per virtualizzare Ubuntu 7.10 purtroppo condivise anche da alcuni di voi. Augusto segnalava un pò di quelli che ha avuto lui, insieme a qualche utile suggerimenti su come risolverli che riporto qui sotto.

Dunque in merito al noto problema della risoluzione video, le tecniche che sono state esposte nei precedenti post non hanno funzionato per la mia macchina quindi ho escogitato un trucchetto banale: ho spostato i panel a destra e sinistra del desktop e questo consente di vedere i pulsanti quel tanto che basta per usarli. Inoltre mi sono imbattuto nel problema del mirror: l'installazione cerca di connettersi probabilmente alla rete per qualche aggiornamento e se la connessione a internet dell'host non c'è oppure qualcosa ancora non va, bene li rimane (all'82%). Per risolverlo basta disconnettere la rete (durante la fase di installazione quando è bloccato), l'installer va in errore (ovviamente) e poi prosegue normalmente. Spero che questa "avventura" sia utile a qualcuno.

Sicuramente Augusto, grazie. :-)

@:\>

Pubblicato alle 16.00 Argomenti: Linux, suggerimenti, Virtualbox, virtualizzazione