

debianizzati e-zine



UNIX TIME 1264018530

PRODOTTO DA
DEBIANIZZATI.ORG

 creative
commons

NUMERO 3
GENNAIO 2010

Indice

Editoriale	1
La Pagina dei lettori	3
0.1 Ripristinare il supporto samba in Midnight Commander	3
1 Storia e filosofia di Debian	9
1.1 DFSG	10
1.1.1 Storia	10
1.1.2 Il contratto sociale	12
1.1.3 Le DFSG	14
2 Il sistema operativo Debian	17
2.1 Debian server tramite preseed	18
2.1.1 Introduzione	18
2.1.2 Prepariamo il materiale	18
2.1.3 Modifiche finali	30
2.1.4 Installazione e configurazione SSH	31
2.1.5 Installazione del server FTP	32
2.1.6 Considerazioni conclusive	34
3 Debian ports	35
3.1 Debian GNU/kFreeBSD	36
3.1.1 Introduzione	36
3.1.2 Installazione	36
3.1.3 Conclusioni	51

4	Hardware & Debian	53
4.1	Debian su android	54
4.1.1	Introduzione	54
4.1.2	Preparazione	56
4.1.3	Installazione	57
4.1.4	Configurazione	60
4.1.5	Screenshot	69
4.1.6	Conclusioni	72
5	Tips & Tricks	75
5.1	Vivere l'Emacs	76
5.1.1	Una brevissima storia	76
5.1.2	Installare e configurare Emacs 23 in Debian Lenny	77
5.1.3	La faccia di Emacs	79
5.1.4	Uso di emacs come editor	81
5.1.5	Altre utilità di Emacs	85
5.1.6	Quante altre cose è Emacs?!	88
5.2	ALSA & Gnome	90
5.2.1	Un po' di storia	90
5.2.2	Le API audio in linux	90
5.2.3	Installare e configurare ALSA	91
5.2.4	Abilitare i suoni in Gnome	92
5.2.5	Riferimenti	94
5.3	Alias	95
5.3.1	Usare gli alias	95
5.3.2	Note di chiusura	115
5.3.3	Ringraziamenti	115
6	Softwares in analisi	117
6.1	Mai più senza backup	118
6.1.1	Premessa	118
6.2	backup2l	119
6.2.1	Installazione	120
6.2.2	Configurazione	121

6.2.3	Esempio di un report di <i>backup</i>	125
6.2.4	Sintassi	128
6.2.5	Utilizzo	131
6.2.6	Conclusioni	133
6.2.7	Riferimenti	133
6.3	Clonezilla	133
6.3.1	Introduzione	133
6.3.2	Il progetto <i>clonezilla</i>	134
6.3.3	Architettura	135
6.3.4	Funzionalità	135
6.3.5	Avvertenze prima dell'uso	136
6.3.6	Casi d'uso	137
6.3.7	Conclusioni	161
6.3.8	Riferimenti	162
6.4	Iproute2: il futuro del networking è già qui	164
6.4.1	Introduzione	164
6.4.2	Installazione	164
6.4.3	Il comando IP	165
6.4.4	Arp	169
6.4.5	Routing	170
6.4.6	Tunneling	173
6.4.7	Conclusioni	175
7	Storia e filosofia del software libero	177
7.1	Nexenta: quando opensolaris incontra GNU	178
7.1.1	Introduzione	178
7.1.2	GNU/OpenSolaris	179
7.1.3	Nexenta	179
7.1.4	ZFS	179
7.1.5	Debian GNU/OpenSolaris	181
7.1.6	StormOS	181
7.1.7	Conclusioni	182
	Impressum	185

Editoriale

Con la neve caduta copiosa in questi giorni è terminato anche l'autunno, mentre l'inverno, bussando alla porta, ci ha annunciato l'inizio del nuovo anno: BUON 2010 A TUTTI!

... l'idea di riprendere un progetto abbandonato da ca. un anno mi stuzzicava...
Era il gennaio del 2009 e la rinascita del progetto ezine, per altro mai nato, stava per venire alla luce...

Con queste parole, un anno fa è iniziato, dunque, il cammino dell'ezine; mentre il primo numero è uscito solo ad aprile dello scorso anno, è stato proprio nel gennaio del 2009 che sono state consolidate le basi del progetto... e chi dimenticherà mai le ore passate a decidere in che formato uscire con i numeri, con che opzioni, con quale grafica: i grandi dubbi dei mesi freddi dell'anno scorso...

Come ogni anno vissuto, anche nel nostro caso sono state maturate nuove esperienze e provate molte emozioni. Si è cercato di integrare al meglio l'e-zine con il resto del portale (proposito che, peraltro, si intende ulteriormente migliorare nel corso di questo nuovo anno). Il gruppo di lavoro è stato più volte amalgamato nella sua articolazione interna, mentre l'impostazione del progetto, dopo le prime grandi e difficili decisioni iniziali, è rimasta sostanzialmente invariata trovando crescente riscontro nella collaborazione offerta dagli utenti della comunità.

In questo numero 3 abbiamo deciso di effettuare qualche piccolo cambiamento. Come prima cosa, per dar maggior importanza ai ports, tema principale di questo numero, abbiamo rinominato il capitolo Debian GNU/Hurd in Debian Ports, proponendovi subito squeeze (la futura stable) supportata al 100% dal kernel freeBSD. Inoltre, viste le difficoltà incontrate nei numeri scorsi ad inserire almeno un articolo per ogni sezione, volendo proporre comunque materiale di qualità (piuttosto che completare forzatamente un capitolo) e per meglio integrare gli articoli stessi nel tema principale del numero, abbiamo deciso di eliminare (anche se in modo solo temporaneo e dipendentemente dal numero) alcuni capitoli

rinunciando in questo nuovo numero 3 a Interfacce grafiche e Il kernel Linux, argomenti che in ogni caso potrebbero essere riproposti nelle prossime uscite della rivista, in base ai temi e agli articoli che verranno sviluppati.

Tra i propositi per il futuro, infine, anticipiamo (ma non troppo, non possiamo ancora svelarvi tutto ;-)) che auspichiamo di gettare le basi per un'ulteriore apertura verso il mondo esterno attraverso la collaborazione con altre comunità interessate allo sviluppo del software libero... stay tuned !

Non posso che augurarvi ancora una buona lettura e rinnovare l'invito a ritrovarci su www.debianizzati.org!

La Pagina dei lettori

Questo mese pubblichiamo come parte integrante della guida MC inserita nel numero 2 dell'e-zine , questo articolo/segnalazione che ci ha inviato fr4nc3sco .

Vedremo come integrare il supporto a samba in MC , poichè di default non viene abilitato .

Pochi semplici passi spiegati molto bene dal nostro attento lettore , vi apriranno nuove porte nell'utilizzo di MC .

0.1 Ripristinare il supporto samba in Midnight Commander

Midnight Commander, comunemente noto come MC, comprende tra i "file system" virtuali (vfs) supportati anche smbfs per samba, ma nel pacchetto presente su debian tale funzionalità è stata disabilitata, come è possibile verificare impartendo il seguente comando:

```
$ mc -V | grep "File System Virtuale"  
File System Virtuale: tarfs, extfs, cpiofs, ftpfs, fish
```

Come riportato nel **Debian Bug Tracking System**¹, il motivo è dovuto al fatto che il codice inerente a "samba" all'interno del sorgente di MC è ormai obsoleto (risale al 1995) e questo determina problemi di malfunzionamenti e sicurezza: pertanto, gli sviluppatori di Debian, per attenersi agli standard di stabilità della distribuzione, hanno deciso di disabilitarne il supporto. Ciò non di meno, ripristinare in MC tale funzionalità non pregiudica l'usabilità del programma. Pertanto, è possibile ripristinare il supporto samba per MC ricompilandone il codice sorgente originale attivando nella configurazione

¹<http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=520114>

l'opzione `--with-samba`: purtroppo, tale strada è però molto sconveniente in quanto è necessario applicare successivamente all'incirca cinquanta "patches" fornite dai maintainers del pacchetto per Debian. In alternativa, è possibile ricreare il pacchetto alla debian-way. Per prima cosa occorre rimuovere MC dal PC ed installare alcuni tools per la manipolazione dei pacchetti con i seguenti comandi:

```
$ su -c "aptitude purge -y mc"
$ su -c "aptitude install -y devscripts cdb"
```

Si userà `/tmp/mc` come directory di lavoro:

```
$ mkdir /tmp/mc && cd /tmp/mc
$ dget -x -u http://ftp.it.debian.org/debian/pool/main/m/mc/mc_4.6.2~git20080311-4.dsc
```

`dget` è una utility usata dagli sviluppatori Debian che permette di creare velocemente un pacchetto sorgente;

- richiede come parametro obbligatorio il riferimento (URL) al file `.dsc` del pacchetto ("debian source control" ovvero al descrittore del pacchetto sorgente,²) che contiene alcuni metadati sul pacchetto stesso: consultando <http://packages.debian.org/> il riferimento del file `.dsc` sarà nella parte destra della pagina riferita al pacchetto per la propria distribuzione;
- scarica dai repository Debian, in formato compresso, il codice sorgente del programma a cui il file `.dsc` si riferisce;
- esclude la verifica delle firme digitali (opzione `-u`);
- estrae il codice sorgente (opzione `-x`) dal pacchetto del codice sorgente che ha scaricato registrandolo in una "directory" chiamata `nomepacchetto-versione/` (dove, al posto di `nomepacchetto` è riportato il nome originale del pacchetto considerato, cioè `mc-4.6.2 git20080311/` nel caso della versione disponibile al momento della stesura del presente testo).

²vedi <http://wiki.debian.org/dsc>

A questo punto, il codice sorgente del programma è pronto ad essere utilizzato ed è possibile impartire i seguenti comandi:

```
$ cd mc-4.6.2~git20080311/  
$ dch -l custom "Enable vfs smb"
```

dch è un tool che serve a manipolare il *changelog* (l'elenco dei cambiamenti effettuati) del pacchetto; in questo caso con

- l'opzione `-l custom` specifica che si desidera aggiungere un suffisso al numero della versione del pacchetto; in particolare, si indica che si sta creando un “pacchetto locale” la cui versione conterrà la parola *custom*;
- l'argomento `Enable vfs smb` indica che si è desidera aggiungere tale testo al *changelog*;

Modificare il *changelog* non è solo una formalità: se non lo facessimo il gestore di pacchetti (APT), dopo l'installazione del nostro pacchetto modificato, tenterebbe di aggiornare nuovamente il pacchetto con quello ufficiale presente nei repository e privo del supporto per *smb*.

A questo punto, occorre installare i programmi necessari alla compilazione di MC; per individuarli, è sufficiente consultare il *file debian/control* alla riga che inizia con `Build-Depends` impartendo il seguente comando:

```
$ cat debian/control | grep Build-Depends  
Build-Depends: debhelper (> 5), libglib2.0-dev, libgpm-dev |not+linsux-gnu, \  
gettext, libslang2-dev, quilt, cvs, automake, binutils
```

Possiamo quindi installare tali pacchetti necessari a soddisfare le dipendenze per compilare MC impartendo i seguenti comandi:

```
$ su -c "aptitude install -y libglib2.0-dev libgpm-dev gettext libslang2-dev \\  
quilt cvs automake binutils"
```

A questo punto, si deve fare in modo che il sistema per la creazione dei pacchetti (build system) compili MC con il supporto samba. Poiché il pacchetto ufficiale è stato creato impostando nella configurazione il parametro `--without-samba`, lo si dovrà sostituire con `--with-samba`. Fino alla versione 4.2.6 di MC (quella disponibile in Lenny), è stato usato il Colin's Build System (cbs) per la creazione del pacchetto, nel quale le variabili con i parametri da passare al compilatore sono contenute nel file `debian/rocks`; la variabile che ci interessa è `DEB_CONFIGURE_EXTRA_FLAGS`, il cui valore sarà passato allo script `configure` prima della compilazione. Il parametro sopra indicato potrà essere modificato con un editor oppure usando `sed`:

```
$ sed -i "s/--without-samba/--with-samba/g" debian/rocks
```

Attenzione! - le versioni successive di MC usano Common Debian Build System (cdfs) per la creazione del pacchetto, il file da modificare sarà `debian/rules` mentre il nome della variabile è sempre lo stesso.

A questo punto, si può creare il pacchetto impartendo i seguenti comandi:

```
$ debuild clean
$ debuild --no-lintian
```

`debuild` si occuperà di compilare il codice sorgente di MC e di creare il pacchetto `.deb`:

- l'opzione `clean` indica a `debuild` di cancellare i file non necessari dalle *directory* del codice sorgente;
- l'opzione `--no-lintian` disattiverà i messaggi di warning.

Alla fine verrà segnalato un errore in quanto non è stato possibile firmare con la propria chiave gpg il pacchetto; questo non è un problema, in quanto il pacchetto è stato creato ed è integro: se si volesse firmarlo basta aggiungere il parametro `-k <IDGpgKey>` al precedente comando.

Si può procedere ora all'installazione di MC:

```
$ su -c "dpkg -i /tmp/mc/*.deb"
```

Per controllare che MC abbia il supporto samba è possibile impartire il comando:

```
$ mc -V | grep "File System Virtuale"
```

```
File System Virtuale: tarfs, extfs, cpiofs, ftpfs, fish, smbfs
```

Inoltre, avviando MC è quindi possibile verificare l'attivazione per il supporto al protocollo smb come da immagine allegata:

Ringraziamo l'attento fr4nc3sco per la segnalazione

firmato : la redazione dell'e-zine

Capitolo 1

Storia e filosofia di Debian



In questa sezione verranno proposti articoli a riguardo la storia e la filosofia che sta dietro al sistema operativo debian.

1.1 DFSG

Le *linee guida del software Debian* (in inglese *Debian Free Software Guidelines*, ossia *DFSG*) sono un insieme di principi sviluppati e condivisi dai creatori e dai sostenitori del progetto Debian per definire la loro concezione di software libero.

Furono abbozzate da Bruce Perens (da un'idea di Ean Schussler) e vennero poi approvate dalla comunità di sviluppatori Debian il 4 Luglio 1997 ¹ al termine di una conferenza via mail tenuta nel corso del mese di Giugno. Dopo la versione 1.0, definita in quella notte di luglio, l'attuale versione è la 1.1, approvata il 26 aprile 2004. Tra le due versioni non esiste nessuna differenza sostanziale per quanto riguarda i contenuti, nell'ultima revisione sono stati semplicemente modificati alcuni termini per rendere il tutto più chiaro e comprensibile.

1.1.1 Storia

A precedere le DFSG si trova la Definizione del Software Libero ²: un documento scritto da Richard Stallman e pubblicato dalla Free Software Foundation (FSF) nel quale viene definito il concetto di software libero. La prima pubblicazione conosciuta della definizione è l'edizione del febbraio 1986 da parte del bollettino GNU della FSF ³ e sostiene due punti fondamentali (n.d.R. tradotto in italiano):

La parola libero nel nostro nome non si riferisce al prezzo; si riferisce alla libertà. Prima di tutto, la libertà di copiare un programma e ridistribuirlo agli altri cosicché loro possano usarlo come te. Come seconda cosa, la libertà di modificare il programma, così tu puoi controllarlo e lui non può controllarti; per questo, il codice sorgente deve essere accessibile.

¹<http://lists.debian.org/debian-announce/1997/msg00017.html>

²<http://www.gnu.org/philosophy/free-sw.it.html>

³<http://www.gnu.org/bulletins/bull1.txt>

La versione attuale che tutti conosciamo si basa invece su quattro principi di libertà:

- Libertà di eseguire il programma, per qualsiasi scopo (libertà 0).
- Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1). L'accesso al codice sorgente ne è un prerequisito.
- Libertà di ridistribuire copie in modo da aiutare il prossimo (libertà 2).
- Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti da voi apportati (e le vostre versioni modificate in genere), in modo tale che tutta la comunità ne tragga beneficio (libertà 3). L'accesso al codice sorgente ne è un prerequisito.

Una curiosa nota è un riferimento ad un noto discorso del presidente degli stati uniti Franklin D. Roosevelt del 6 gennaio del 1941, nel quale citava quattro libertà fondamentali: libertà di parola e di espressione, libertà di credo, libertà dal bisogno e libertà dalla paura. Chissà se Richard M. Stallman non si sia ispirato proprio a Roosevelt per la stesura delle sue libertà ;-). Ma torniamo alle DSFG...

Nel febbraio del 1998, Bruce Perens ed Eric Raymond fondano la Open Source Initiative (OSI) ⁴. Essi si basarono sugli stessi principi delle DFSG per creare l' Open Source Definition (OSD) ⁵. Per fare ciò è stato sufficiente eliminare tutti riferimenti al sistema specifico Debian dandogli così una valenza più generale. Nonostante la strettissima relazione fra le DFSG e l'OSD, il nuovo concetto di open source viene criticato molto dalla comunità del software libero; in particolare Richard M. Stallman, il quale afferma che il punto centrale prettamente pragmatico dell'iniziativa distrae gli utenti dalle questioni più importanti di ambito morale e dalla libertà offerta dal software libero, offuscando la distinzione tra software semi-libero o totalmente proprietario ⁶.

La storia si complica ulteriormente quando si tratta di associare licenze approvate dall'OSI alle DSFG. Un esempio è dato dalla CDDL ⁷ (v. anche articolo su Nexenta); quest'ultima viene riconosciuta dall'OSI e anche considerata libera dalla FSF; in ogni caso viene

⁴<http://www.opensource.org/>

⁵<http://www.opensource.org/docs/osd>

⁶http://it.wikipedia.org/wiki/Open_Source_Initiative#Relazioni_con_il_movimento_per_il_software_libero

⁷http://en.wikipedia.org/wiki/Common_Development_and_Distribution_License

dichiarata non-compatibile con la GPL (General Public License) ⁸ a causa di alcune restrizioni relative al concetto di proprietà intellettuale. D'altro lato non è chiaro a nessuno se sia effettivamente compatibile con le DFSG ⁹; al momento Debian la considera però quasi per definizione non compatibile, anche se una spiegazione più dettagliata è difficile da trovare e anche il wiki ¹⁰ non dice niente a riguardo. Le calde discussioni delle mailinglist ¹¹, così come gli interventi al debconf del 2006 ¹² non aiutano ancora ad inquadrare meglio la questione. Calcolando infine la derivazione stretta della CDDL con la MPL (Mozilla Public License) ¹³ e che la MPL è considerata compatibile con le DFSG ¹⁴ capirete quando è difficile capire... e scusate il gioco di parole ;-).

Nel novembre del 1998, Ian Jackson, propose delle modifiche per un'eventuale versione 1.4 ¹⁵. Le proposte di Jackson furono alla base della seconda revisione del contratto sociale; in ogni caso, le DFSG non sono apparentemente mai state modificate in modo almeno significativo.

1.1.2 Il contratto sociale

Il contratto sociale ¹⁶ è rappresentato da 5 punti cardine su cui si basa lo sviluppo del sistema Debian. A seguire la versione 1.1 del contratto. Per i più curiosi, la versione originale del 1997, la 1.0, è disponibile al seguente indirizzo:

http://www.debian.org/social_contract.1.0.it.html

1. Debian rimarrà libera al 100%

Forniamo le linee guida che usiamo per determinare se un'opera sia libera nel documento intitolato The Debian Free Software Guidelines. Promettiamo che il sistema Debian e tutte le sue componenti rimarranno liberi in accordo con le citate linee guida. Supporteremo le persone che creino o usino in Debian opere sia libere che non libere. Non renderemo mai il sistema dipendente da un componente non libero.

⁸<http://www.gnu.org/licenses/gpl.html>

⁹<http://soundadvice.id.au/blog/2005/02/04/#cddl>

¹⁰<http://wiki.debian.org/DFSGLicenses>

¹¹<http://lists.debian.org/debian-devel/2006/08/msg00552.html>

¹²[http://meetings-archive.debian.net/pub/debian-meetings/2006/debconf6/theora-small/2006-05-](http://meetings-archive.debian.net/pub/debian-meetings/2006/debconf6/theora-small/2006-05-14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps_Alvaro_Lopez_Ortega.ogg)

[14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps_Alvaro_Lopez_Ortega.ogg](http://meetings-archive.debian.net/pub/debian-meetings/2006/debconf6/theora-small/2006-05-14/tower/OpenSolaris_Java_and_Debian-Simon_Phipps_Alvaro_Lopez_Ortega.ogg)

¹³<http://www.mozilla.org/MPL/MPL-1.1.html>

¹⁴<http://wiki.debian.org/DFSGLicenses#MozillaPublicLicense.28MPL.29>

¹⁵<http://lists.debian.org/debian-devel/1998/11/msg01944.html>

¹⁶http://www.debian.org/social_contract.it.html

2. Renderemo alla Comunità Free Software

Quando scriviamo nuovi componenti del sistema Debian, li rilasceremo con una licenza che rispetti le Debian Free Software Guidelines. Realizzeremo il sistema migliore che potremo, cosicché le opere libere siano usate e distribuite il più possibile. Comunicheremo cose come bug fix, migliorie e richieste degli utenti agli autori - upstream delle opere incluse nel nostro sistema.

3. Non nasconderemo i problemi

Manterremo sempre il nostro intero bug report database aperto alla pubblica lettura. I rapporti che le persone invieranno online saranno prontamente resi visibili a tutti.

4. Le nostre priorità sono gli utenti ed il software libero

Ci faremo guidare dai bisogni dei nostri utenti e della comunità del software libero. Metteremo al primo posto i loro interessi. Supporteremo le necessità dei nostri utenti di operare in molti diversi tipi di ambienti di calcolo. Non ci opporremo alle opere non libere che siano state pensate per l'uso in sistemi Debian e non richiederemo compensi a chi crea o usa queste opere. Permetteremo ad altri di creare distribuzioni contenenti sia il sistema Debian che altre opere, senza richiedere compensi. Per raggiungere questi scopi, forniremo un sistema integrato di materiali di alta qualità senza alcuna restrizione legale che limiti qualsiasi uso del sistema.

5. Opere che non rispettano i nostri standard free software

Ci rendiamo conto che alcuni dei nostri utenti richiedono di usare opere non conformi alle Debian Free Software Guidelines. Abbiamo creato le aree contrib e non-free nel nostro archivio per queste opere. I pacchetti in queste aree non fanno parte del sistema Debian, sebbene siano stati configurati per l'uso con Debian. Invitiamo i realizzatori di CD a leggere le licenze dei pacchetti in queste aree per determinare se possono distribuire i pacchetti sui loro CD. Inoltre, anche se le opere non libere non fanno parte di Debian, supporteremo il loro uso e forniremo infrastrutture per i pacchetti non liberi (come il nostro bug tracking system e le mailing list).

1.1.3 Le DFSG

Infine vi riportiamo le DFSG in versione originale (tradotte in italiano) presenti nella pagina di debian ¹⁷.

1. Libera redistribuzione

La licenza di un componente Debian non può porre restrizioni a nessuno per la vendita o la cessione del software come componente di una distribuzione software aggregata di programmi proveniente da fonti diverse. La licenza non può richiedere royalty o altri pagamenti per la vendita.

2. Codice sorgente

Il programma deve includere il codice sorgente. Inoltre la sua distribuzione deve poter avvenire sia come codice sorgente che in forma compilata.

3. Lavori derivati

La licenza software deve permettere modifiche e lavori derivati e deve permettere la loro distribuzione con i medesimi termini della licenza del software originale.

4. Integrità del codice sorgente dell'autore

La licenza può porre restrizioni sulla distribuzione di codice sorgente modificato solo se permette la distribuzione di file patch insieme al codice sorgente con lo scopo di modificare il programma durante la compilazione. La licenza deve esplicitamente permettere la distribuzione di software compilato con codice sorgente modificato. La licenza può richiedere che i lavori derivati abbiano un nome o un numero di versione diversi da quelli del software originali.

Nota al punto 4: tale punto va visto come compromesso tra la possibilità di modificare il codice e il diritto da parte degli autori ad essere riconosciuti come tali.

5. Nessuna discriminazione di persone o gruppi

La licenza non può discriminare nessuna persona o gruppo di persone.

¹⁷http://www.debian.org/social_contract.it.html#guidelines

6. Nessuna discriminazione nei campi di impiego

La licenza non può porre restrizioni all'utilizzo del programma in uno specifico campo di impiego. Per esempio, non può porre restrizioni all'uso commerciale o nella ricerca genetica.

7. Distribuzione della licenza

I diritti applicati al programma devono essere applicabili a chiunque riceva il programma senza il bisogno di utilizzare licenze aggiuntive di terze parti.

8. La licenza non può essere specifica per Debian

I diritti applicati al programma non possono dipendere dal fatto che esso sia parte di un sistema Debian. Se il programma è estratto da Debian e usato o distribuito senza Debian ma ottemperando ai termini della licenza, tutte le parti dovrebbero avere gli stessi diritti di coloro che lo ricevono con il sistema Debian.

9. La licenza non deve contaminare altro software

La licenza non può porre restrizioni ad altro software che sia distribuito insieme al software concesso in licenza. Per esempio, la licenza non può richiedere che tutti gli altri programmi distribuiti con lo stesso supporto debbano essere software libero.

10. Esempi di licenze

Le licenze GNU General Public License, BSD e Artistic sono esempi di licenze che consideriamo libere.

Capitolo 2

Il sistema operativo Debian



In questa sezione, tutto sul sistema operativo debian GNU/Linux (per debian GNU/hurd si segue la sezione dedicata).

2.1 Debian server tramite preseed

2.1.1 Introduzione

Il preseeding (o preconfigurazione) è un tipo di installazione offerto dall'installer Debian che permette di preconfigurare le risposte alle domande poste dall'installer rendendo così automatizzato il processo. Oltre a questo offre funzionalità che non sono presenti in una normale installazione, quali per esempio la scelta di pacchetti particolari e l'esecuzione di script mirati per la configurazione della macchina.

Per la preconfigurazione si possono usare tre metodi: `initrd`, `file` e `network`. La preconfigurazione `initrd` funziona con tutti i metodi d'installazione e consente la preconfigurazione di parecchie domande tramite un file ma richiede la preparazione più lunga. Le preconfigurazioni `file` e `network` si possono usare con metodi d'installazione diversi. Ci interessiamo al metodo *initrd*, più impegnativo ma decisamente più interessante rispetto agli altri due. Stiamo per fare un'installazione di un server (ma il processo si può configurare anche per un qualsiasi pc) in modo un pò anomalo, non in modo diretto, ma tramite un'immagine iso un pò modificata. L'obiettivo è fare in modo che la macchina, una volta inserito il CD (o il supporto USB), faccia tutto da sola cioè non si renda necessario interagire direttamente con questa. La domanda spontanea è: Perché complicarsi la vita?

I motivi possono essere diversi.

Un caso particolare potrebbe essere quello di un'azienda che compra n computer identici e sui quali si vuole installare il sistema Debian. Un'altra situazione interessante si presenta quando noi in prima persona non possiamo accedere direttamente alla macchina o non possiamo lavorare sulla macchina perchè priva di monitor, tastiera e mouse.

In questo modo, con il solo dispositivo mobile (CD-Rom o USB), è possibile installare tutto il sistema e configurare il server pronto per essere gestito da remoto da dove potremo completare tranquillamente l'installazione dei pacchetti necessari.

Quello che tenteremo di costruire quindi è un server perfettamente funzionante, gestibile tramite protocollo SSH.

2.1.2 Prepariamo il materiale

La prima cosa da fare è procurarsi il sistema Debian per un'installazione minimale. Useremo il ramo Stable, dato che l'obiettivo è quello di ottenere un server perfettamente funzionante, sicuro ed affidabile. Scarichiamo la versione più recente nella forma NetInstall

dal sito ufficiale. Al momento della stesura dell'articolo la versione più aggiornata è la 5.0.3 ¹

Creiamo sul Desktop una cartella che chiamiamo *temp*, rinominiamo l'immagine appena scaricata e spostiamola all'interno di questa cartella:

```
$ mkdir temp
$ mv debian-503-i386-netinst.iso temp/NetInstImage.iso
```

Il file appena scaricato, seppur perfettamente funzionante, non fa esattamente quello che vogliamo noi. Se masterizzassimo l'immagine su un supporto CD, avviando la macchina questa porterebbe ad un'installazione classica: menu dell'installer, con poi tutte le domande a cui rispondere per definirne la configurazione.

Vediamo quindi come fare perchè tutto proceda in modo automatizzato.

La versatilità di Debian (e di linux in generale) da questo punto di vista è incredibile. È qui che entra in gioco il *preseeding* ovvero l'utilizzo di un file contenente le risposte alle domande di configurazione che vengono poste durante l'installazione del sistema.

Per velocizzare il procedimento ho creato uno script apposito, che fa essenzialmente quanto segue:

- monta l'immagine in una cartella temporanea;
- modifica l'initrd inserendo il file per il preseeding;
- modifica l'installer in modo da avviare automaticamente l'installazione;
- aggiorna l'md5sum;
- crea una nuova iso;

Ancora da terminale, ci spostiamo nella cartella *temp* e creiamo il file:

```
$ cd temp
$ gedit ISO_maker.sh
```

e copiamo all'interno lo script riportato qui sotto:

¹<http://cdimage.debian.org/debian-cd/5.0.3/i386/iso-cd/debian-503-i386-netinst.iso>
<http://cdimage.debian.org/debian-cd/5.0.3/i386/iso-cd/debian-503-i386-netinst.iso>

```
#!/bin/bash
#
# Script Bash per l'inserimento del file preseed nell'initrd all'interno
# dell'iso di Debian.
# Necessario:
# - immagine di Debian NetInstall rinominata in NetInstImage.iso
# - file preseed.cfg nella stessa cartella dell'immagine iso
# - pacchetto rsync
# - modulo loop attivo.

mkdir -p loopdir
mount -o loop NetInstImage.iso loopdir
rm -rf cd
mkdir cd
rsync -a -H --exclude=TRANS.TBL loopdir/ cd
umount loopdir

# Modifica all'initrd
mkdir irmod
cd irmod
gzip -d < ../cd/install.386/initrd.gz | \
    cpio --extract --verbose --make-directories --no-absolute-filenames

# Inseriamo il file con la configurazione
cp ../preseed.cfg preseed.cfg

# Imponiamo l'avvio automatico dell'installer ;)
cd ../cd/isolinux
sed -i "s%timeout 0%timeout 1%g" isolinux.cfg
cd ..
cd ../irmod

# Comprimiamo il tutto
find . | cpio -H newc --create --verbose | \
    gzip -9 > ../cd/install.386/initrd.gz
cd ../
rm -fr irmod/
```

```
# Aggiornamento dell'md5sum
cd cd
md5sum `find -follow -type f` > md5sum.txt
cd ..

# Creazione della nuova immagine

# Affinchè il cd sia avviabile da boot è necessario passare i parametri corretti
# a mkisofs.
# Creiamo l'immagine CD usando l'immagine di boot isolinux con il comando:

mkisofs -o test.iso -r -J -no-emul-boot -boot-load-size 4 \
  -boot-info-table -b isolinux/isolinux.bin -c isolinux/boot.cat ./cd
```

Verifichiamo di avere il pacchetto *rsync* (eventualmente installiamolo), quindi proseguiamo. Ci resta da configurare il file di preseeding. Creiamo quindi il file *preseed.cfg* nella cartella temp, dove si trovano anche i file *ISO_maker.sh* e *NetInstImage.iso*. All'interno vi inseriamo il codice seguente:

```
#### Contents of the preconfiguration file (for lenny)
```

```
### Localization
```

```
# Con locale si impostano lingua e paese.
d-i debian-installer/locale string it_IT
```

```
### Selezione della tastiera.
```

```
#d-i console-tools/archs select at
d-i console-keymaps-at/keymap select it
# Esempio con una diversa architettura della tastiera
#d-i console-keymaps-usb/keymap select mac-usb-us
```

```
### Network configuration
```

```
# netcfg sceglierà, se possibile, un'interfaccia che abbia un collegamento.
# Questo evita la visualizzazione di una lista se è presente più di
# un'interfaccia.
d-i netcfg/choose_interface select auto

# Oppure scegliere un'interfaccia ben precisa:
#d-i netcfg/choose_interface select eth1

# Se si ha un server dhcp lento e l'installatore va in timeout in attesa
# di esso, questo potrebbe essere utile.
#d-i netcfg/dhcp_timeout string 60

# Se si preferisce configurare manualmente la rete, togliere il commento
# alla prossima riga e alle righe con la configurazione statica della rete.
#d-i netcfg/disable_dhcp boolean true

# Se si desidera che il file di preconfigurazione funzioni su qualsiasi
# sistema, a prescindere dalla presenza di un server DHCP in rete, si può
# togliere il commento alle righe seguenti e alle righe con la configurazione
# statica della rete.
#d-i netcfg/dhcp_failed note
#d-i netcfg/dhcp_options select Configure network manually

# Configurazione statica della rete.
#d-i netcfg/get_nameservers string 192.168.1.1
#d-i netcfg/get_ipaddress string 192.168.1.42
#d-i netcfg/get_netmask string 255.255.255.0
#d-i netcfg/get_gateway string 192.168.1.1
#d-i netcfg/confirm_static boolean true

# Si noti che gli eventuali nomi di host e dominio assegnati da dhcp hanno
# la precedenza sui valori qui impostati. Naturalmente, l'impostazione dei
# valori previene ancora che le domande siano visualizzate anche se i
# valori arrivano da dhcp.
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain

# Disabilita la noiosa finestra di dialogo della chiave WEP.
```

```
d-i netcfg/wireless_wep string
# Lo stupido nome host dhcp che alcuni ISP usano come una sorta
# di password.
#d-i netcfg/dhcp_hostname string radish

# Se per la scheda di rete o per altro hardware è necessario caricare un
# firmware non-libero, è possibile configurare l'installatore in modo
# che lo carichi senza chiedere conferma. Altrimenti cambiare in false
# per disabilitare la richiesta.
#d-i hw-detect/load_firmware boolean true

### Network console
# Per effettuare un'installazione da remoto tramite SSH con il componente
# network-console, usare le seguenti impostazioni. Questo ha senso se si
# intende effettuare il resto dell'installazione manualmente.
#d-i anna/choose_modules string network-console
#d-i network-console/password password r00tme
#d-i network-console/password-again password r00tme

### Mirror settings
# Quando si seleziona ftp non è necessario impostare mirror/country.
#d-i mirror/protocol string ftp
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.it.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

# Versione da installare.
#d-i mirror/suite string testing
# Versione dei componenti dell'installatore da caricare (facoltativo).
#d-i mirror/udeb/suite string testing

### Clock and time zone setup
# Controlla se l'orologio hardware deve essere impostato o meno su UTC.
d-i clock-setup/utc boolean true
```

```
# Questo può essere impostato con qualsiasi valore valido per $TZ; si
# consulti /usr/share/zoneinfo/ per i valori validi.
d-i time/zone string Europe/Rome

# Determina se usare NTP per impostare l'orologio durante l'installazione
d-i clock-setup/ntp boolean true
# Server NTP da usare. Quasi sempre è possibile usare quello predefinito.
#d-i clock-setup/ntp-server string ntp.example.com

### Partitioning
# Se il sistema ha spazio libero è possibile scegliere di partizionare
# soltanto quello spazio.
#d-i partman-auto/init_automatically_partition select biggest_free

# In alternativa è possibile specificare un disco da partizionare.
# Il nome del dispositivo deve essere dato nel tradizionale formato
# non-devfs.
# Nota: è obbligatorio specificare un disco tranne nel caso che il
# sistema abbia un disco solo.
# Per esempio, per usare il primo disco SCSI/SATA:
#d-i partman-auto/disk string /dev/sda
# Inoltre è necessario specificare il metodo da usare. Quelli attualmente
# disponibili sono "regular", "lvm" e "crypto".
d-i partman-auto/method string lvm

# L'utente riceve un avviso se uno dei dischi che stanno per essere
# partizionati automaticamente contiene già una precedente configurazione
# LVM. Comunque anche questo può essere preconfigurato...
d-i partman-lvm/device_remove_lvm boolean true
# La stessa cosa avviene se già esiste un array in RAID software:
d-i partman-md/device_remove_md boolean true
# E può essere preconfigurata anche la conferma alla scrittura delle
# partizioni lvm.
d-i partman-lvm/confirm boolean true

# È possibile scegliere una qualsiasi delle ricette di partizionamento
```

```
# predefinite:
# - atomic: tutti i file in una partizione
# - home: partizione separata per /home
# - multi: partizioni separate per /home, /usr, /var e /tmp
d-i partman-auto/choose_recipe select atomic

# Oppure creare una ricetta personalizzata...
# Il formato della ricetta è documentato nel file devel/partman-auto-recipe.txt.
# Se si ha modo di importare un file ricetta dentro l'ambiente
# dell'installatore, è possibile semplicemente puntare a questo.
#d-i partman-auto/expert_recipe_file string /hd-media/recipe

# Altrimenti, è possibile mettere un'intera ricetta in una riga. Questo
# esempio crea una piccola partizione /boot, un'area swap appropriata e
# usa il resto dello spazio per la partizione root:
#d-i partman-auto/expert_recipe string \
#     boot-root :: \
#         40 50 100 ext3 \
#             $primary{ } $bootable{ } \
#             method{ format } format{ } \
#             use_filesystem{ } filesystem{ ext3 } \
#             mountpoint{ /boot } \
#         . \
#         500 10000 1000000000 ext3 \
#             method{ format } format{ } \
#             use_filesystem{ } filesystem{ ext3 } \
#             mountpoint{ / } \
#         . \
#         64 512 300% linux-swaps \
#             method{ swap } format{ } \
#         .

# Questo fa eseguire a partman il partizionamento automatico in base a
# quanto indicato con uno dei metodi precedenti, senza chiedere conferma.
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
```

```
### Base system installation
# Selezionare il programma per la creazione dell'initrd per i kernel 2.6.
#d-i base-installer/kernel/linux/initramfs-generators string yaird

# Il (meta)pacchetto con l'immagine del kernel da installare; per non
# installare un kernel inserire "none".
#d-i base-installer/kernel/image string linux-image-2.6-486

### Account setup
# Saltare la creazione dell'account di root (l'account per l'utente
# normale potrà usare sudo).
#d-i passwd/root-login boolean false
# In alternativa, è possibile saltare la creazione dell'account per
# l'utente normale.
#d-i passwd/make-user boolean false

# Password di root, in chiaro entrambe le volte
d-i passwd/root-password password amministratore
d-i passwd/root-password-again password amministratore
# oppure criptata in un hash MD5.
#d-i passwd/root-password-crypted password [MD5 hash]

# Per creare l'account per l'utente normale.
d-i passwd/user-fullname string defaultuser
d-i passwd/username string defaultuser
# La password dell'utente, in chiaro entrambe le volte
d-i passwd/user-password password pass_temporanea
d-i passwd/user-password-again password pass_temporanea
# oppure criptata in un hash MD5.
#d-i passwd/user-password-crypted password [MD5 hash]
# Per creare il primo utente con un UID specifico al posto di quello
# predefinito.
#d-i passwd/user-uid string 1010

# L'account utente viene automaticamente inserito in alcuni gruppi.
# Per modificare questo comportamento usare la seguente preconfigurazione.
```

```
#d-i passwd/user-default-groups string audio cdrom video

### Apt setup
# È possibile scegliere di installare software non-free e contrib.
d-i apt-setup/non-free boolean true
d-i apt-setup/contrib boolean true
# Togliere il commento alla prossima riga se non si vuole usare
# un mirror.
d-i apt-setup/use_mirror boolean false
# Scegliere i servizi di aggiornamento da utilizzare; definire quali
# mirror devono essere usati. I valori riportati in seguito sono quelli
# predefiniti.
#d-i apt-setup/services-select multiselect security, volatile
#d-i apt-setup/security_host string security.debian.org
#d-i apt-setup/volatile_host string volatile.debian.org

# Altri repository, sono disponibili local[0-9]
#d-i apt-setup/local0/repository string \
#     http://local.server/debian stable main
#d-i apt-setup/local0/comment string local server
# Abilita le righe deb-src
#d-i apt-setup/local0/source boolean true
# URL alla chiave pubblica del repository locale; è necessario fornire
# una chiave altrimenti apt si lamenta perché il repository non è
# autenticato e che quindi non dovrebbe essere usato (commentandolo in
# sources.list)
#d-i apt-setup/local0/key string http://local.server/key

# Normalmente l'installatore richiede che i repository siano autenticati
# con una chiave gpg conosciuta. Per disabilitare questa autenticazione si
# può usare questa impostazione. Attenzione: insicuro, non raccomandato.
#d-i debian-installer/allow_unauthenticated string true

### Package selection
tasksel tasksel/first multiselect standard
# Modificare come sotto per installare anche un web-server
```

```
# tasksel tasksel/first multiselect standard, web-server

# Quando è selezionato il task desktop, si può installare kde e xfce
# al posto del desktop predefinito gnome.
#tasksel tasksel/desktop multiselect kde, xfce

# Ulteriori pacchetti da installare
d-i pkgselect/include string ssh openssh-server build-essential
# Se aggiornare gli altri pacchetti dopo l'avvio. I valori ammessi
# sono: none, safe-upgrade, full-upgrade
#d-i pkgselect/upgrade select none

# Alcune versioni dell'installatore possono inviare un resoconto di quali
# programmi sono stati installati e quali sono usati. Questo resoconto, la
# cui ricezione aiuta il progetto a determinare quali sono i programmi più
# popolari per inserirli nei CD, normalmente non viene inviato.
popularity-contest popularity-contest/participate boolean false

### Boot loader installation
# Grub è il boot loader predefinito (per x86). Invece, se si desidera
# che sia installato lilo, scommentare questa riga:
#d-i grub-installer/skip boolean true
# Per evitare anche l'installazione di lilo, e quindi non installare
# nessun boot loader, scommentare questa riga:
#d-i lilo-installer/skip boolean true

# Questa è ragionevolmente sicura da impostare, fa in modo che grub sia
# installato automaticamente nel MBR se nessun altro sistema operativo è
# rilevato nella macchina.
d-i grub-installer/only_debian boolean true

# Questa fa in modo che grub-installer si installi sul MBR se trova anche
# qualche altro sistema operativo, che è un po' meno sicuro poiché potrebbe
# non riuscire ad avviare gli altri sistemi operativi.
#d-i grub-installer/with_other_os boolean true

# In alternativa, se si vuole installare in una posizione differente dal
```

```
# MBR, scommentare ed editare queste linee:
#d-i grub-installer/only_debian boolean false
#d-i grub-installer/with_other_os boolean false
#d-i grub-installer/bootdev string (hd0,0)
# Per installare grub su più dischi:
#d-i grub-installer/bootdev string (hd0,0) (hd1,0) (hd2,0)

# Passowrd opzionale per grub, può essere in chiaro
#d-i grub-installer/password password r00tme
#d-i grub-installer/password-again password r00tme
# oppure criptata in un hash MD5, si veda grub-md5-crypt(8).
#d-i grub-installer/password-crypted password [MD5 hash]

### Finishing up the installation
# Durante le installazioni da console seriale o di gestione, normalmente
# le console virtuali (da VT1 a VT6) vengono disabilitate in /etc/inittab.
# Togliere il commento dalla prossima riga per evitare che questo accada.
#d-i finish-install/keep-consoles boolean true

# Evita l'ultimo messaggio riguardo il completamento dell'installazione.
d-i finish-install/reboot_in_progress note

# Impedisce che il programma d'installazione espelli il CD prima del riavvio,
# in alcuni casi può essere utile.
#d-i cdrom-detect/eject boolean false

# Indica come spegnere quando l'installatore ha finito, senza riavviare
# il sistema appena installato.
#d-i debian-installer/exit/halt boolean true
# Questo spegne la macchina anziché semplicemente arrestarla.
#d-i debian-installer/exit/poweroff boolean true

### Preconfigurazione di altri pacchetti
# A seconda del software che si sceglie di installare, o se le cose vanno
# storte durante il processo di installazione, è possibile che altre domande
# possano essere poste. Si possono preconfigurare anche quelle,
```

```
# naturalmente. Per ottenere una lista di ogni possibile domanda che possa
# essere posta durante un'installazione, fare un'installazione, quindi
# eseguire questi comandi:
# debconf-get-selections --installer > file
# debconf-get-selections >> file

#### Opzioni avanzate
### Running custom commands during the installation
# d-i preseeding is inherently not secure. Nothing in the installer checks
# for attempts at buffer overflows or other exploits of the values of a
# preconfiguration file like this one. Only use preconfiguration files from
# trusted locations! To drive that home, and because it's generally useful,
# here's a way to run any shell command you'd like inside the installer,
# automatically.

# This first command is run as early as possible, just after
# preseeding is read.
#d-i preseed/early_command string anna-install some-udeb

# This command is run just before the install finishes, but when there is
# still a usable /target directory. You can chroot to /target and use it
# directly, or use the apt-install and in-target commands to easily install
# packages and run commands in the target system.
#d-i preseed/late_command string apt-install zsh; in-target chsh -s /bin/zsh
```

Diamo i permessi di esecuzione allo script, quindi eseguiamolo:

```
$ chmod +x ISO_maker.sh
$ sudo ./ISO_maker.sh
```

Creata così la nuova immagine possiamo masterizzare il CD per effettuare l'installazione.

2.1.3 Modifiche finali

Affinchè il server sia effettivamente raggiungibile dall'esterno dobbiamo fare in modo che il suo indirizzo sia univoco e non cambi col tempo. Ci resta quindi da configurare la rete,

per poi poter utilizzare SSH da un qualsiasi computer esterno. Per far questo possiamo fare riferimento a quanto spiegato nell'articolo su `switchconf`² scritto da *mm-barabba* e *pmate* sullo scorso numero dell'ezine. Tuttavia, per evitare malintesi nel proseguimento di quest'articolo, riporto qui il contenuto del file `/etc/network/interfaces` (modificarlo a piacere per ip differenti):

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.1.201
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

il che significa che il server si troverà all'indirizzo ip `'192.168.1.201'`.

Nota: è possibile automatizzare tutto questo tramite script sempre con il nostro preseed ma, onde evitare malintesi, sugli ip ho preferito esplicitare questo passaggio.

2.1.4 Installazione e configurazione SSH

SSH (Secure SHell) è un protocollo che permette di stabilire una sessione remota cifrata ad interfaccia a linea di comando con un altro host. L'intera comunicazione avviene in modo cifrato ed è per questo che SSH è diventato un protocollo standard nella gestione remota di sistemi UNIX. Risulta quindi perfetto per l'utilizzo che ne vogliamo fare.

Abbiamo già installato il demone `ssh` e tutto il necessario tramite `preseed`, ci resta solamente da configurare un paio di parametri per garantire una maggiore sicurezza al sis-

²http://e-zine.debianizzati.org/web-zine/numero_2/?page=61

tema. Apriamo quindi il file `/etc/ssh/sshd_config` da root e cambiamo la porta di ascolto di default del demone.

Fate bene attenzione al nome del file, in quanto ve ne sono altri con nomi simili ma non vanno modificati.

Sostituiamo la riga

```
Port 22
```

con

```
Port 656
```

Attenzione: in questo momento la configurazione non è ancora molto sicura in quanto è abilitato l'accesso all'utente root. Ci sono infatti molti programmi che permettono un bruteforce sulla macchina server e che riescono a trovare in un tempo relativamente breve la password per entrare. Per evitare questo consiglio la seguente procedura: assicurarsi di avere installato il pacchetto `sudo`, creare un nuovo utente nel sistema (che chiamiamo **goofy**), dargli i permessi di root tramite la modifica al file `sudoers` ed infine modificare la voce:

```
PermitROOTLOGIN YES
```

con

```
PermitROOTLOGIN NO
```

nel file `sshd.config`. Rimando alle relative guide presenti nella sezione *Guide* del portale di Debianizzati qualora non fosse chiaro come effettuare le operazioni sopra descritte. D'ora in poi per poter usufruire dei permessi di root sarà necessario autenticarsi con questo nuovo utente e passare a root tramite il comando `sudo su`. Salviamo ora il file ed usciamo.

2.1.5 Installazione del server FTP

Abbiamo terminato l'installazione del demone ssh, per sperimentare quanto appena realizzato procediamo in questo modo: spostiamoci su un altro pc interno alla nostra rete. Su questa macchina il nostro utente è **pippo**:

```
pippo@home $
```

Sulla macchina remota vogliamo però autenticarci come utente goofy (l'utente creato precedentemente con i permessi di root). Da terminale quindi digitiamo

```
pippo@home $ ssh -p 656 -l goofy 192.168.1.201
password:
```

Dopo aver digitato la password, se corretta, siamo nel sistema. Cioè tutto quello che digitiamo da terminale viene eseguito sul server che abbiamo appena configurato! Possiamo quindi immaginare di continuare l'ultima parte di questo articolo dalla postazione in cui ci troviamo.

L'utente *goofy* ha la possibilità di autenticarsi come amministratore, ci basta digitare:

```
$ su
password:
#
```

Ora, nella maggior parte dei server, torna molto utile installare un demone FTP per il trasferimento dei file. Nel nostro caso possiamo installare proftpd. Per far questo è necessario modificare i repository (che fino ad ora non ci interessavano). Apriamo vim e modifichiamo il file *sources.list* andando a commentare le righe relative al CD dell'installazione e inserendovi i repository ufficiali di lenny:

```
## Debian Stable (lenny)
deb http://ftp.it.debian.org/debian/ lenny main contrib non-free
#deb-src http://ftp.it.debian.org/debian/ lenny main contrib non-free

## Aggiornamenti della sicurezza
deb http://security.debian.org/ lenny/updates main contrib
#deb-src http://security.debian.org/ lenny/updates main contrib
```

Ora aggiorniamo l'elenco dei pacchetti e installiamo il demone necessario:

```
# apt-get update
# apt-get install proftpd
```

Una volta terminata l'installazione è possibile personalizzare il demone in maniera precisa. Per effettuare le giuste configurazioni vi rimando quindi all'ottima guida ³ presente sul sito.

³http://guide.debianizzati.org/index.php/Installare_un_server_FTP

2.1.6 Considerazioni conclusive

Allo stato attuale dell'installazione il server non è raggiungibile dall'esterno ma solo dalla rete interna. Con qualche altra modifica anche questo risulta essere un passaggio molto semplice da realizzare.

Consiglio di consultare la sezione guide del portale per questo scopo.

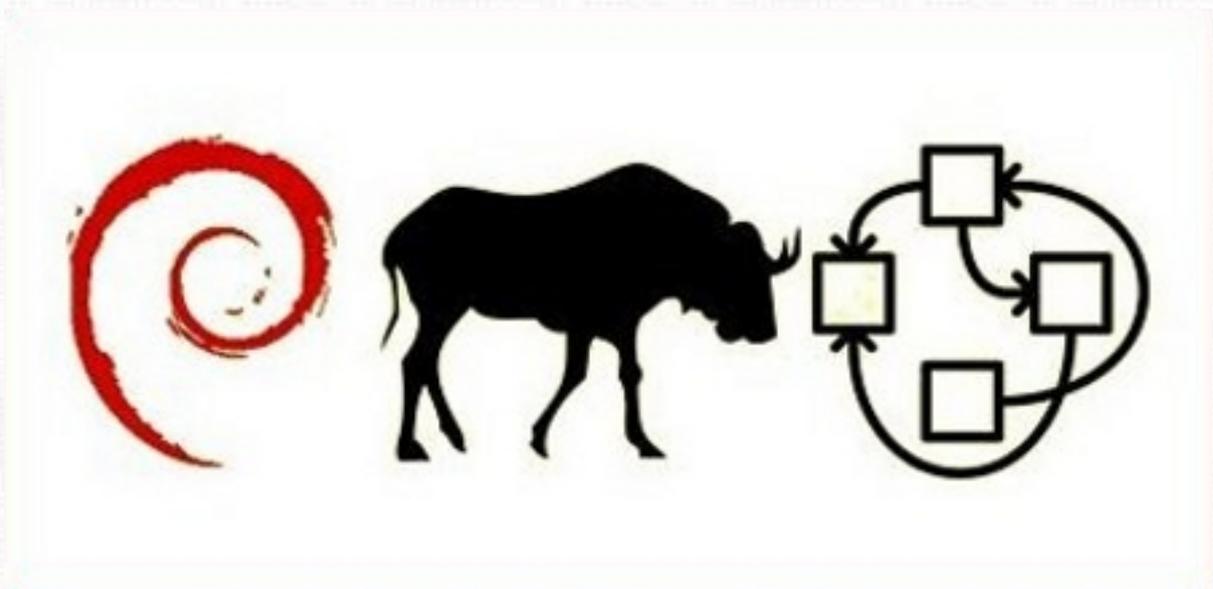
Ancora, nell'installazione tramite preseed è caldamente consigliato che i componenti che interessano questa guida (come per esempio la scheda di rete) siano riconosciuti correttamente dal sistema per evitare problemi nelle configurazioni. E' possibile che i nomi dei dispositivi non corrispondano con quelli indicati quindi è meglio verificare (se possibile) prima di effettuare l'installazione eventualmente con un livecd e modificare dove necessario secondo le proprie esigenze.

Un ringraziamento particolare agli amici del Linux User Group di Verona (LugVR) che hanno inserito sul loro sito⁴ ottimo materiale da cui ho preso spunto per la stesura dell'articolo.

⁴<http://www.verona.linux.it/index.php/Ssh>

Capitolo 3

Debian ports



Conoscevatelo solo Debian GNU/Linux? Vi daremo la possibilità di studiare il vostro sistema operativo preferito sulla base di altri kernel.

Dopo i primi articoli su Debian GNU/Hurd una prima installazione di Debian GNU/kfreeBSD.

3.1 Debian GNU/kFreeBSD

3.1.1 Introduzione

Come annunciato dal team Debian <http://www.debian.org/News/2009/20091007>, Squeeze, la prossima Debian 6.0 stable, sarà la prima ad usare, oltre al kernel Linux anche il kernel FreeBSD.

Utilizzare il kernel FreeBSD consente di godere di alcuni benefici legati alle funzionalità garantite da tale kernel, quale ad esempio l'OpenBSD Packet Filter, il supporto al file system ZFS di Sun, alla Network Driver Interface Specification (NDIS), una minore vulnerabilità, un supporto per hardware che non è supportato da Linux o che ha un supporto migliore e migliori prestazioni o maggiore stabilità per dischi e filesystem.

Attualmente Debian GNU/kFreeBSD è disponibile per architettura amd64 ed i386 anche se è ancora molto unstable.

L'installazione di Debian GNU/kFreeBSD può avvenire seguendo due strade differenti: la prima (obsoleta) è in stile FreeBSD, ossia si utilizza una ISO di circa 110 MB (il cui ultimo aggiornamento però risale allo scorso luglio) reperibile da <http://glibc-bsd.alioth.debian.org/install-cd> che si basa su una versione modificata di FreeBSD sysinstall per la quale è possibile seguire la documentazione presente su <http://glibc-bsd.alioth.debian.org/doc>; la seconda è in stile Debian, si utilizza mini ISO di 12 MB creata giornalmente presente su <http://d-i.debian.org/daily-images/>.

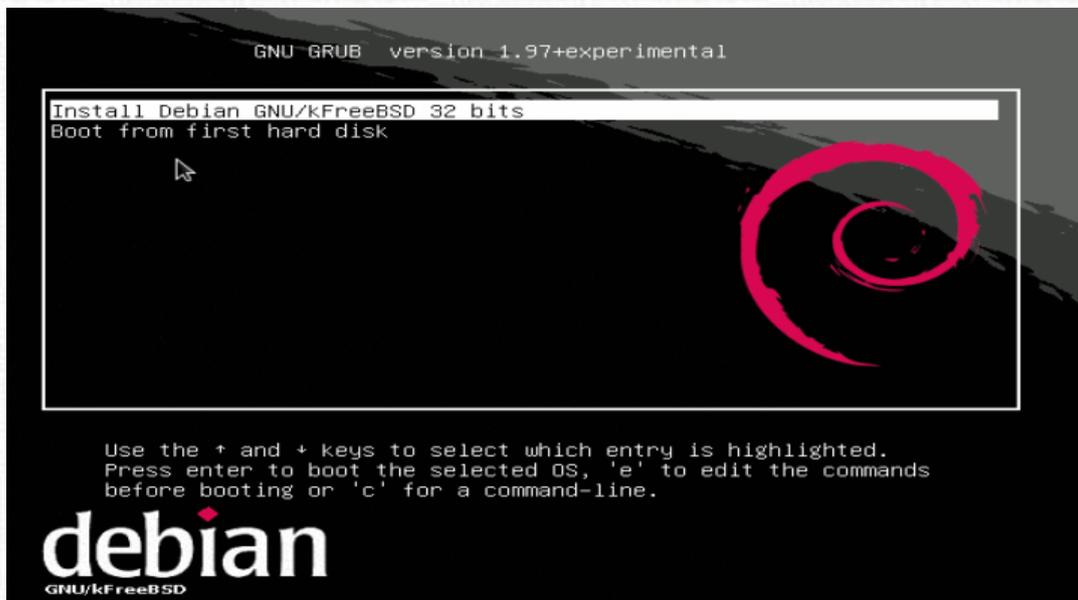
Abbiamo voluto provare ad installare Debian GNU/kFreeBSD, di seguito un mini how-to su come procedere.

3.1.2 Installazione

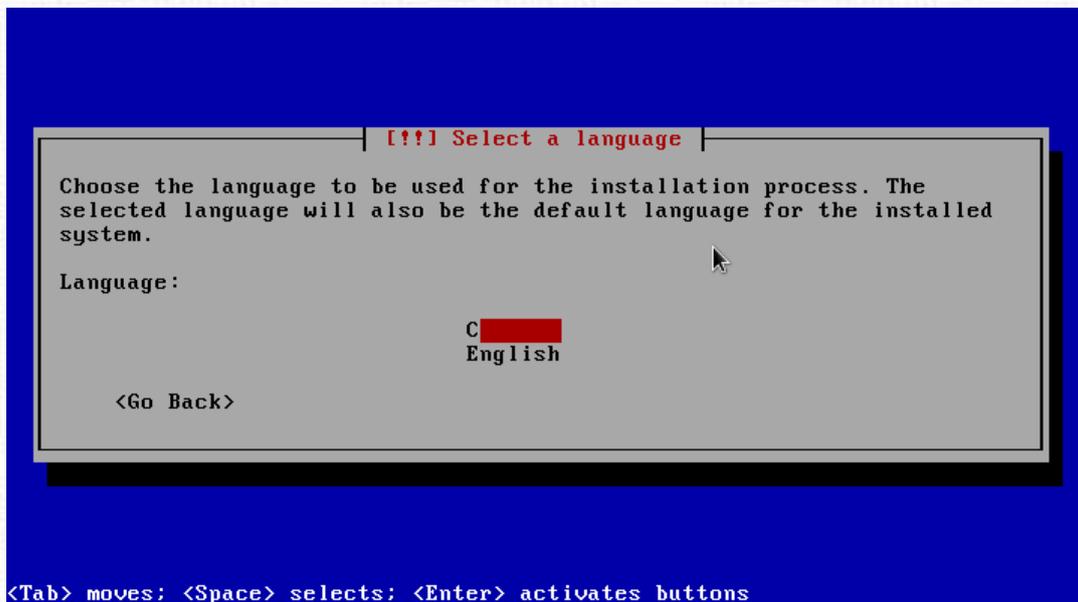
Warningbox - Debian GNU/kFreeBSD è in continuo sviluppo e quindi alcuni programmi potrebbero non funzionare, perciò in questa fase è da usare solamente a livello sperimentale.

Fatta la dovuta premessa passiamo ad analizzare l'installazione di Debian GNU/kFreeBSD su una macchina virtuale di 5G e 256M di ram.

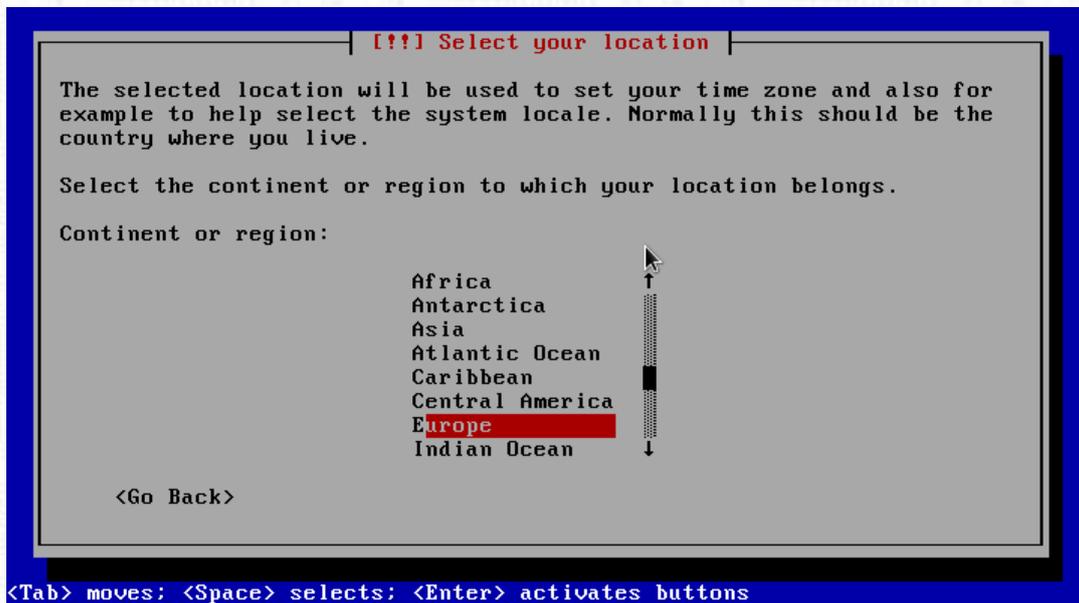
Dopo aver scaricato la mini ISO per la nostra architettura, nel nostro caso kfreebsd-i386, procediamo con l'installazione. (Attenzione alcune ISO possono essere difettose motivo per cui l'installazione non va in porto; in tal caso è possibile provare a scaricare altre ISO).



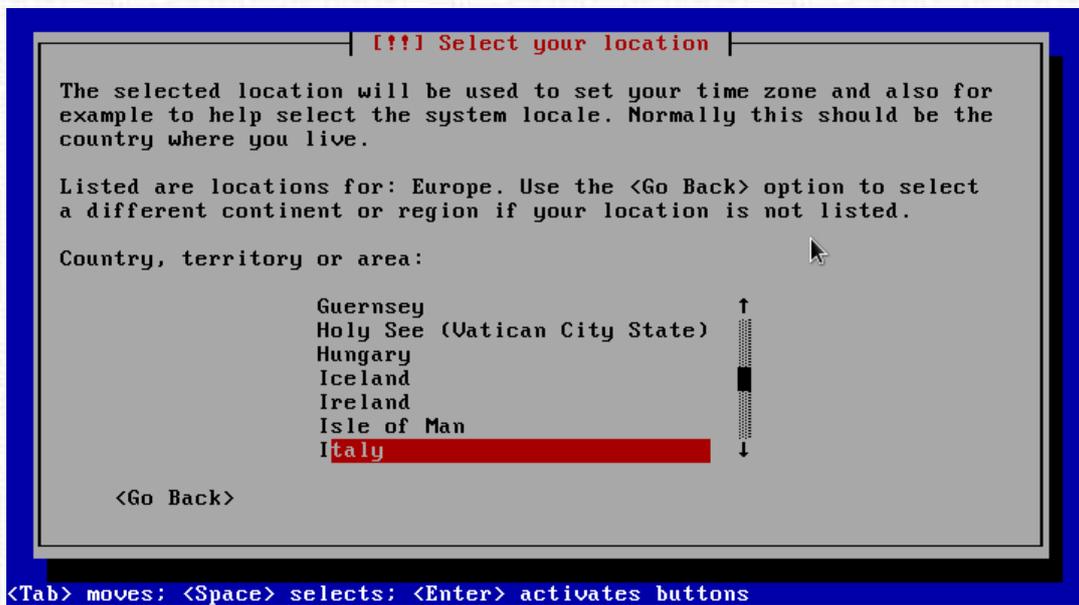
Questa è la schermata che ci si presenta al boot, selezionare ovviamente la prima voce Install.....



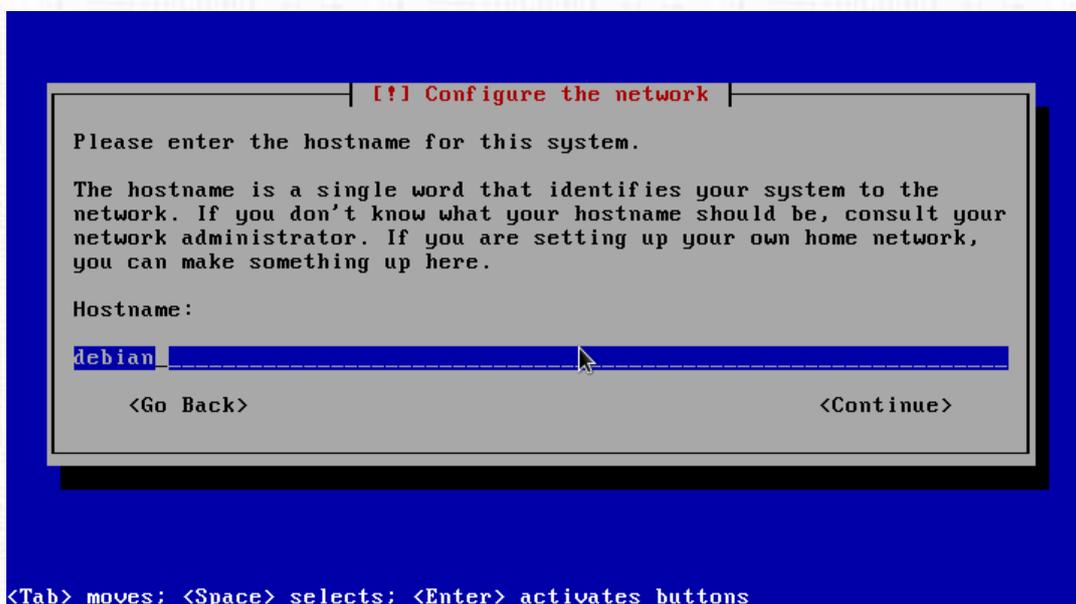
Scegliamo la lingua, per altre lingue oltre l'inglese selezionare C.....



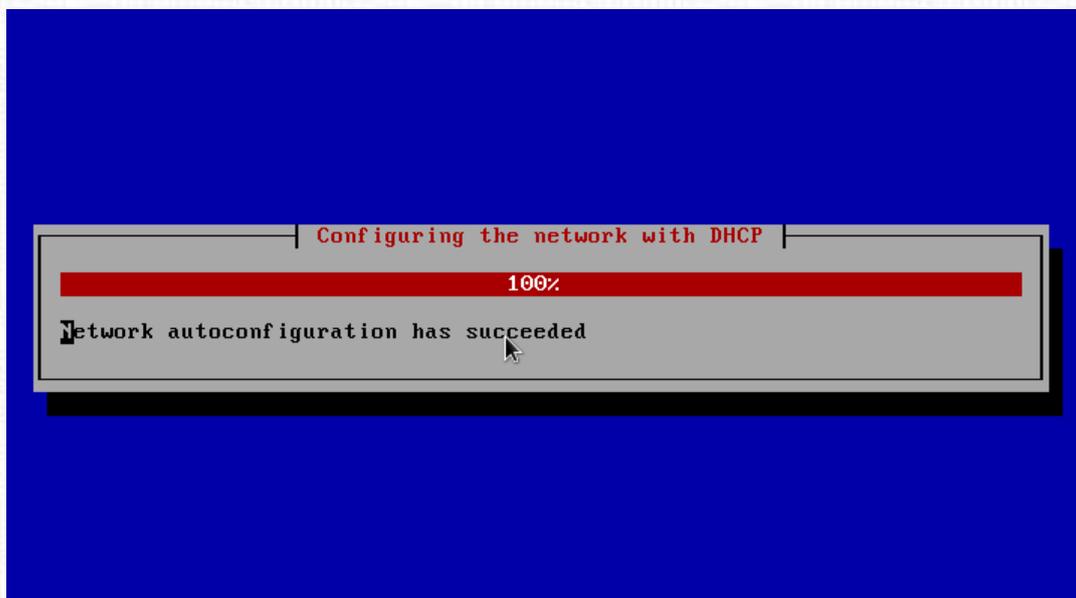
per l'italiano selezionare Europe....



e di seguito Italy....



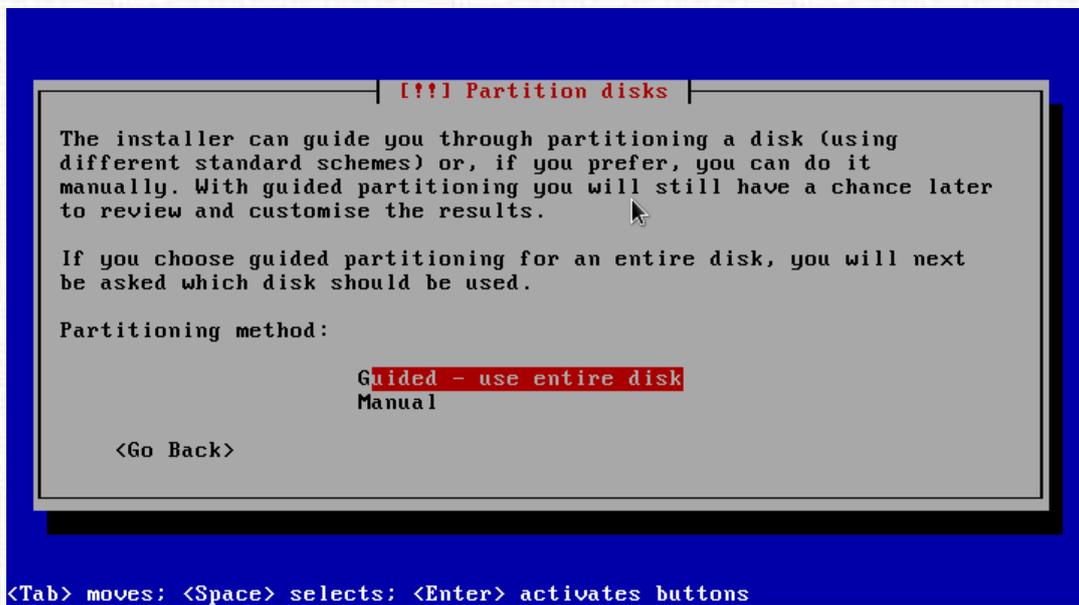
ora diamo un nome alla nostra macchina...



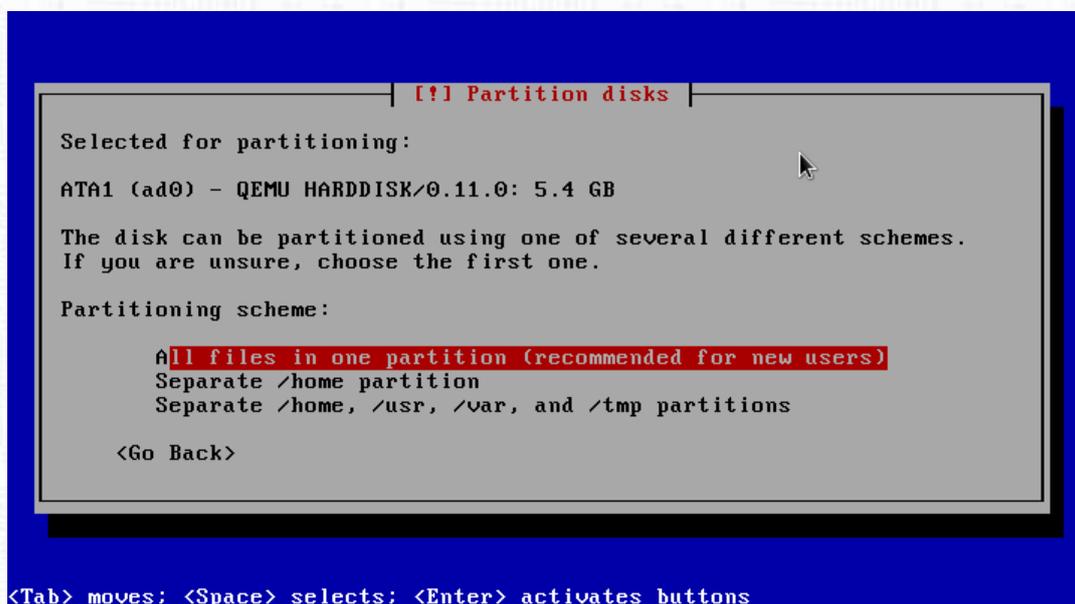
subito dopo parte il riconoscimento automatico della rete con dhcp, se si vuol fare una configurazione manuale: selezionare annulla e scegliere configura la rete manualmente



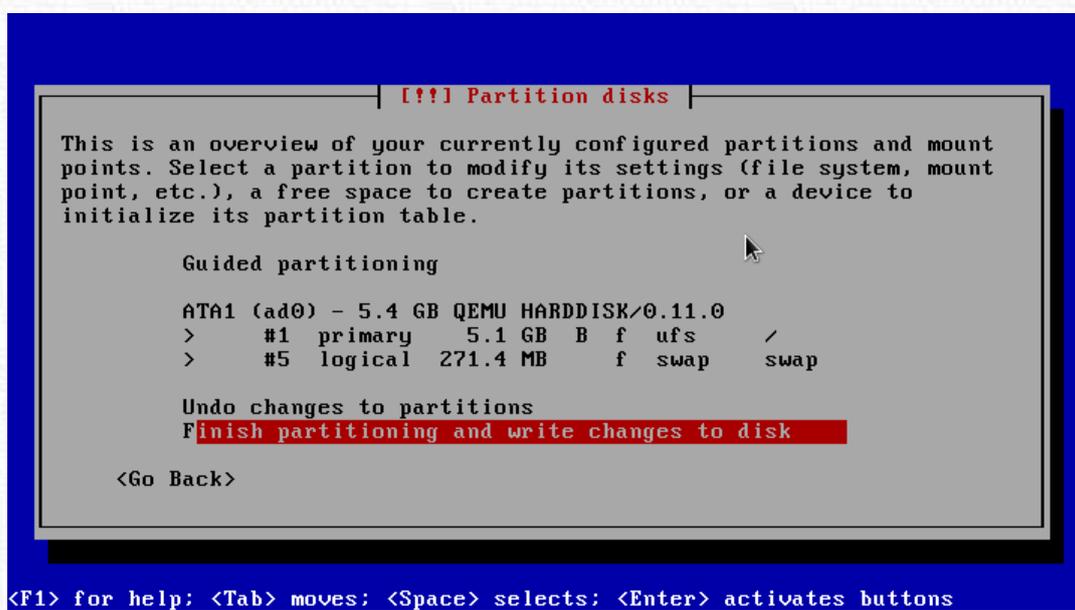
dopo aver configurato la rete scegliamo il mirror da cui scaricare il sistema base ed il resto



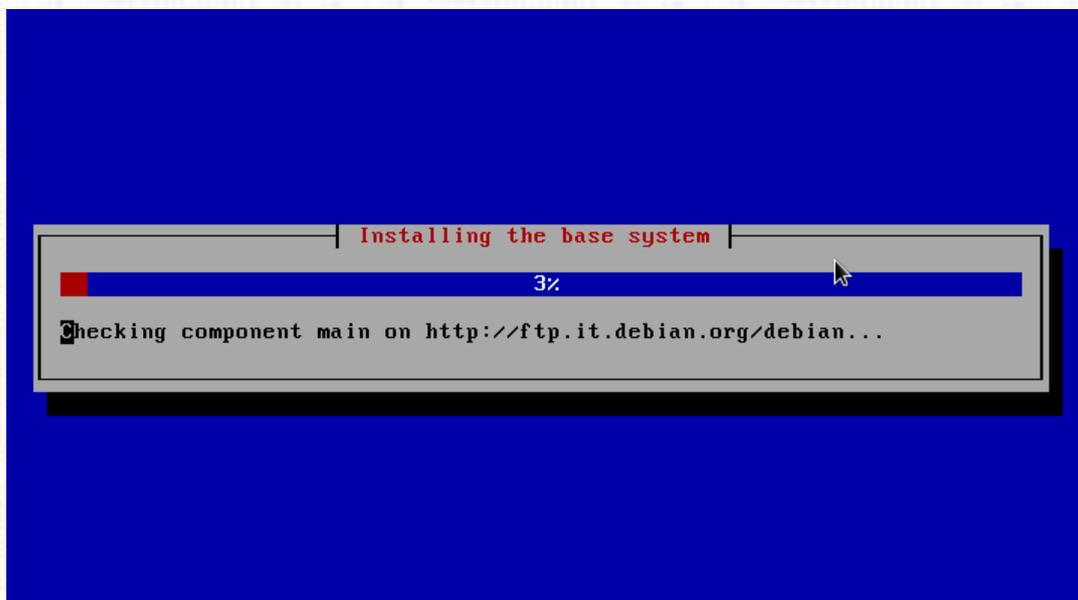
siamo arrivati al partizionamento del disco, ci affidiamo al partizionamento automatico (ovviamente se si vuole si può proseguire manualmente)



scegliamo di installare tutto in una singola partizione



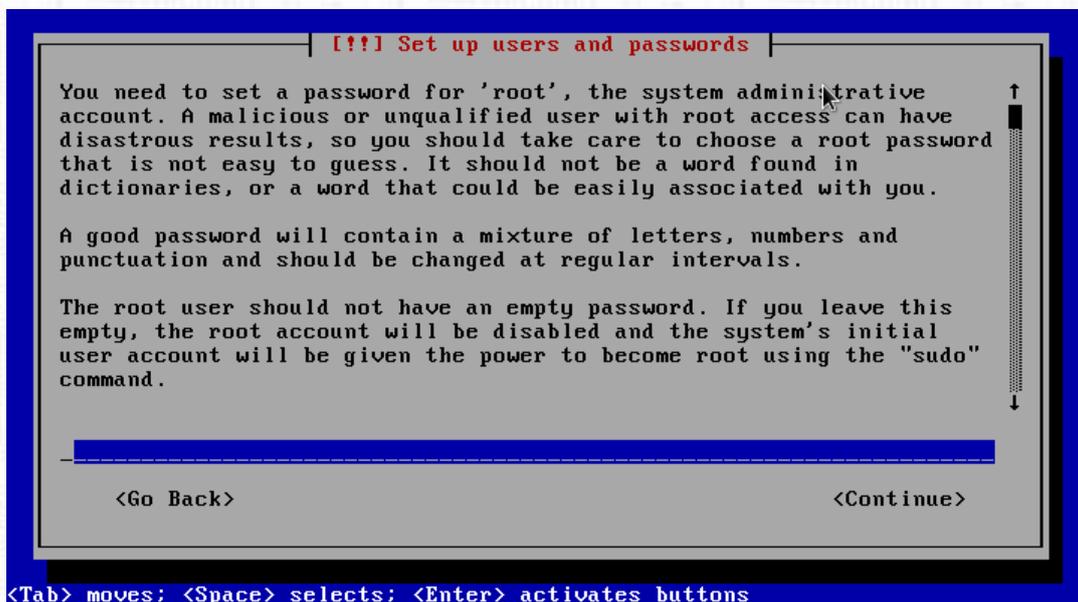
diamo invio e scriviamo i cambiamenti sul disco



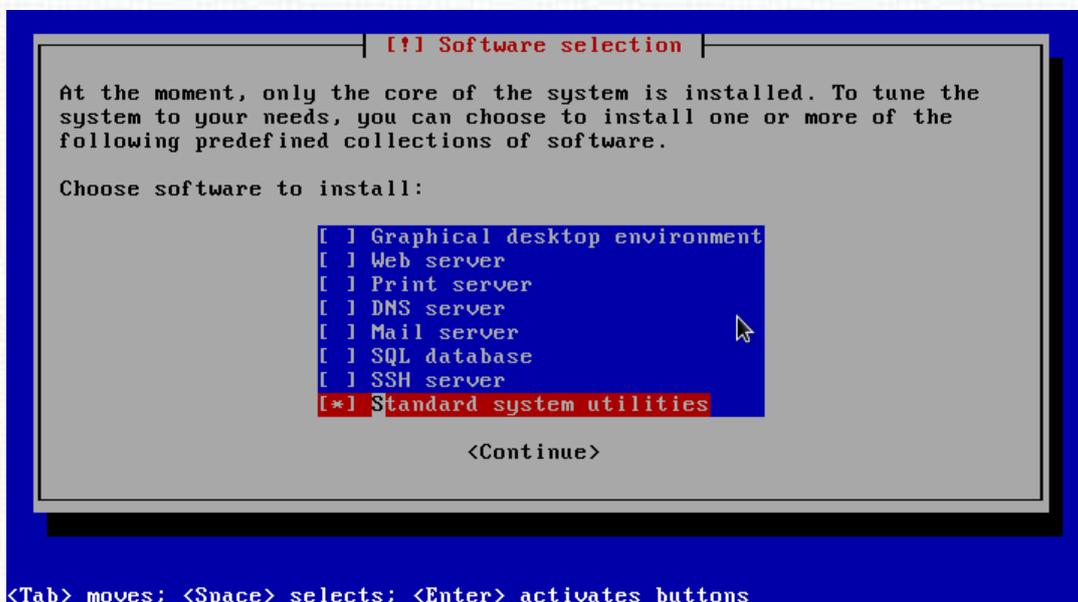
ora viene scaricato il sistema base queste è una fase delicata dove si può incorrere in qualche problema



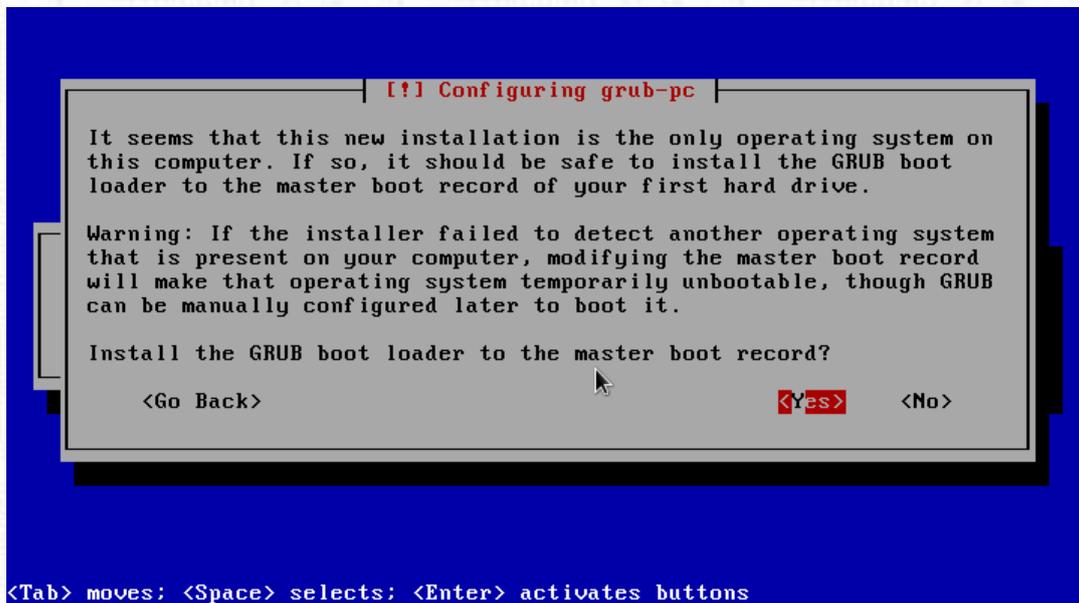
eccolo! A questo punto proviamo a scaricare un'altra mini ISO 8-)



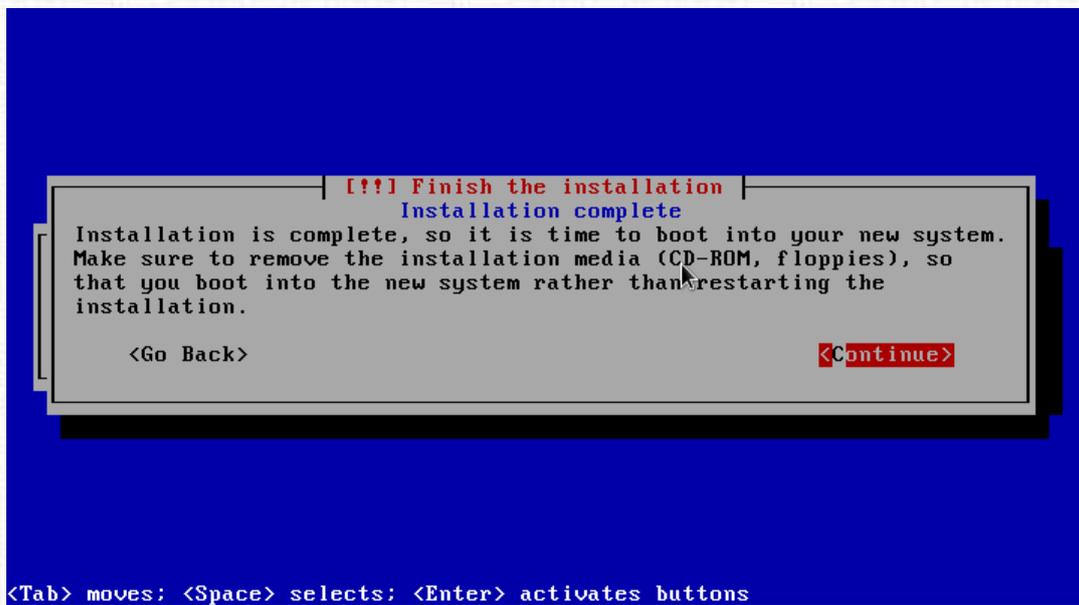
se invece tutto va in porto, avremo questa schermata inseriamo la password di root e procediamo inserendo il nostro utente e relativa password



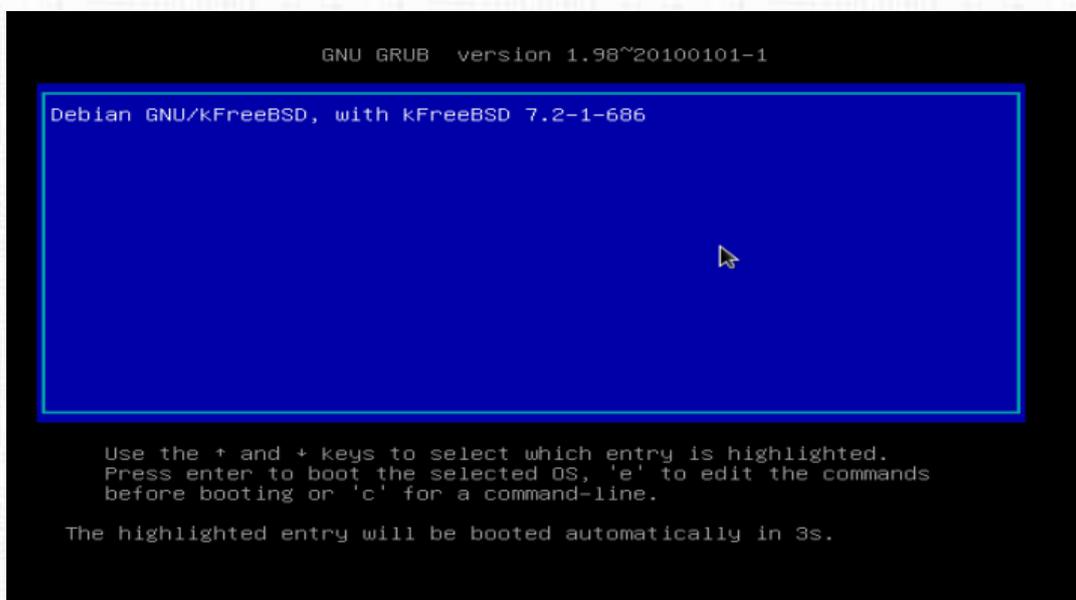
tasksel ci propone cosa installare, in questa fase scegliamo solo il sistema base



infine installiamo grub2 sul MBR.



Finalmente abbiamo finito l'installazione del sistema base e di grub togliamo il disco d'installazione e riavviamo la macchina.



Ecco il boot di grub2

```
Listening on BPF/ed0/52:54:00:12:34:56
Sending on BPF/ed0/52:54:00:12:34:56
Sending on Socket/fallback
DHCPDISCOVER on ed0 to 255.255.255.255 port 67 interval 8
DHCPOFFER from 10.0.2.2
DHCPREQUEST on ed0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 10.0.2.15 -- renewal in 37328 seconds.
done.
Starting portmap daemon...
Cleaning up temporary files...
Loading console keymap...done.
INIT: Entering runlevel: 2
Starting portmap daemon...Already running..
Starting enhanced syslogd: rsyslogdmknod: '/dev/xconsole': Operation not support
ed
chown: cannot access '/dev/xconsole': No such file or directory
.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: cron.
Starting MTA: exim4.

Debian GNU/kFreeBSD squeeze/sid debsd ttyv0
debsd login: █
```

facciamo il login come root

```

Cleaning up temporary files....
Loading console keymap...done.
INIT: Entering runlevel: 2
Starting portmap daemon...Already running..
Starting enhanced syslogd: rsyslogdmknod: `/dev/xconsole': Operation not supported
chown: cannot access `/dev/xconsole': No such file or directory
.
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: cron.
Starting MTA: exim4.

Debian GNU/kFreeBSD squeeze/sid debsd ttyv0

debsd login: root
Password:
GNU/kFreeBSD debsd 7.2-1-686 #0 Tue Oct 6 16:51:07 CEST 2009 i686

The programs included with the Debian GNU/kFreeBSD system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/kFreeBSD comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
debsd:~# apt-get install gnome-desktop-environment

```

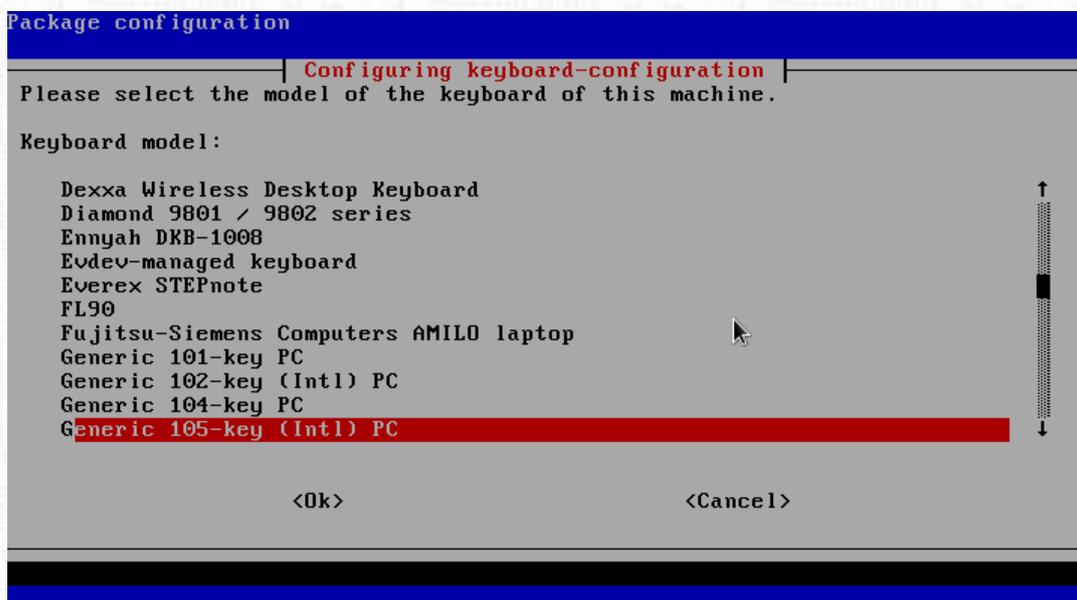
ora siamo pronti per installare il server grafico e il DE ” Ricorda: in questa fase abbiamo la tastiera americana” diamo il comando come da immagine

```

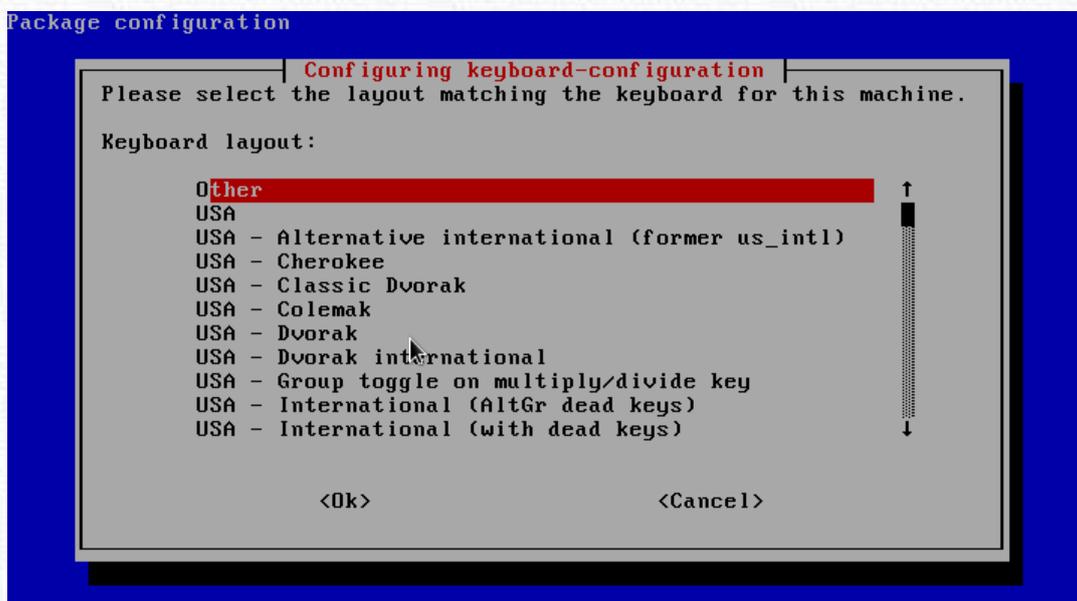
python-twisted-web python-utidglib python-wnck python-xdg
python-zope.interface rarian-compat rpm-common rpm2cpio rss-glx scrollkeeper
seahorse seahorse-plugins sgml-data shared-mime-info sound-juicer sudo
swfdec-gnome synaptic system-tools-backends tcptraceroute
telepathy-mission-control-5 telepathy-salut totem totem-coherence
totem-common totem-mozilla totem-plugins ttf-dejavu ttf-dejavu-core
ttf-dejavu-extra usbutils vinagre vino wodim x-ttcidfont-conf x11-apps
x11-session-utils x11-utils x11-xfs-utils x11-xkb-utils x11-xserver-utils
xbase-clients xfonts-encodings xfonts-utils xinit xkb-data xserver-common
xserver-xephyr xserver-xorg xserver-xorg-core xserver-xorg-input-all
xserver-xorg-input-kbd xserver-xorg-input-mouse xserver-xorg-video-apm
xserver-xorg-video-ark xserver-xorg-video-ati xserver-xorg-video-chips
xserver-xorg-video-cirrus xserver-xorg-video-i128 xserver-xorg-video-i740
xserver-xorg-video-intel xserver-xorg-video-mach64 xserver-xorg-video-mga
xserver-xorg-video-neomagic xserver-xorg-video-nv
xserver-xorg-video-openchrome xserver-xorg-video-r128
xserver-xorg-video-radeon xserver-xorg-video-rendition xserver-xorg-video-s3
xserver-xorg-video-s3virge xserver-xorg-video-savage
xserver-xorg-video-siliconmotion xserver-xorg-video-sis
xserver-xorg-video-tdfx xserver-xorg-video-trident xserver-xorg-video-tseng
xserver-xorg-video-vesa xserver-xorg-video-voodoo xsltproc yelp zenity
0 upgraded, 672 newly installed, 0 to remove and 0 not upgraded.
Need to get 470MB of archives.
After this operation, 1223MB of additional disk space will be used.
Do you want to continue [Y/n]?

```

accettiamo quello che ci viene proposto e proseguiamo e dopo alcuni minuti...



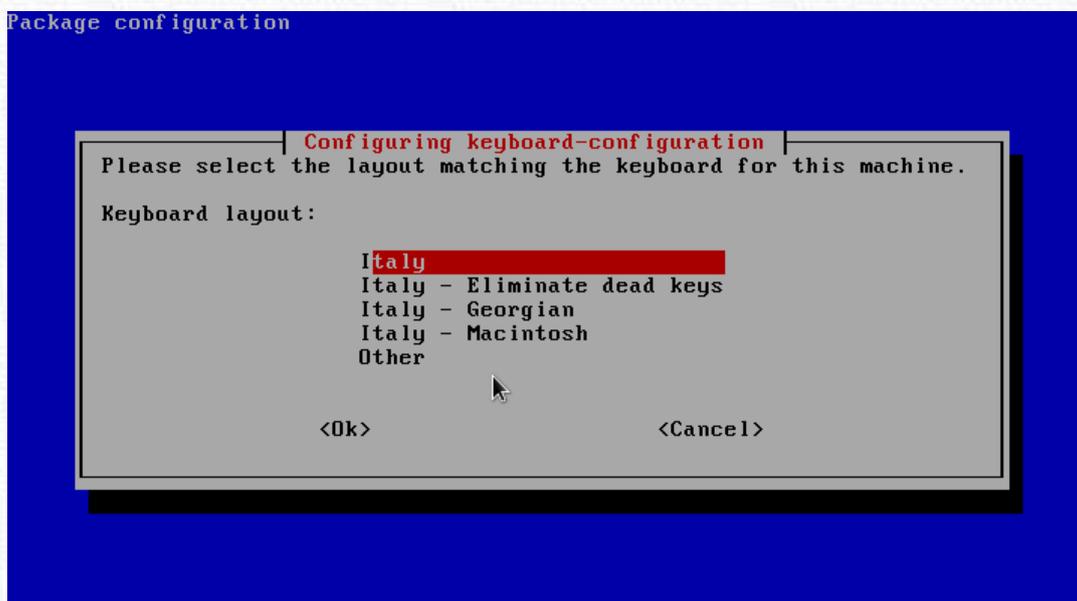
xorg ci permette di configurare la tastiera...



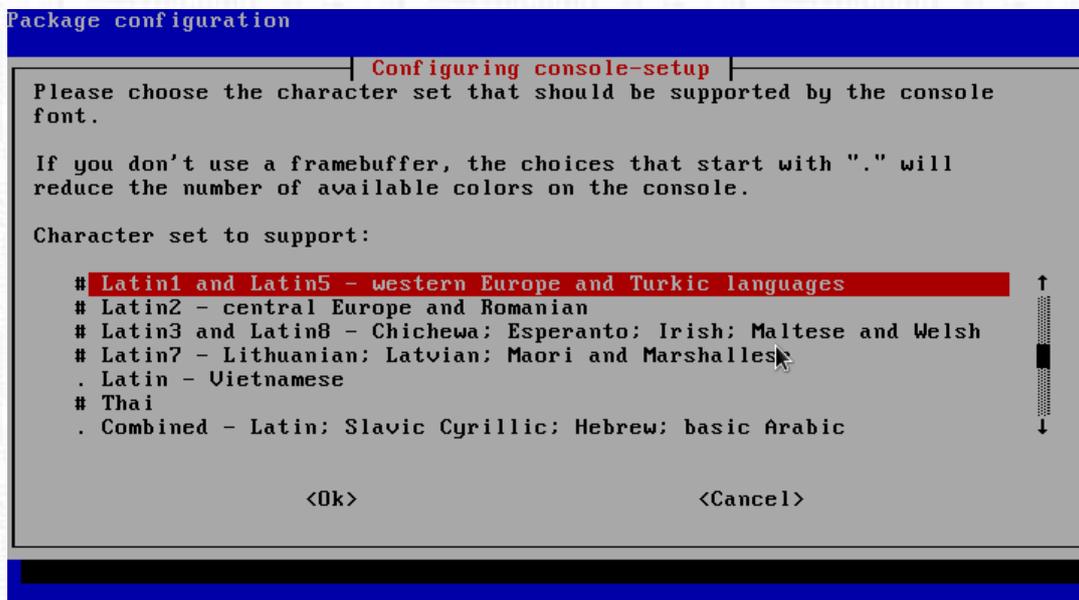
seguiamo le indicazioni come da foto...



selezioniamo Italy..



anche il layout..



e quindi il set dei caratteri...e siamo pronti ad entrare nel nostro DE.

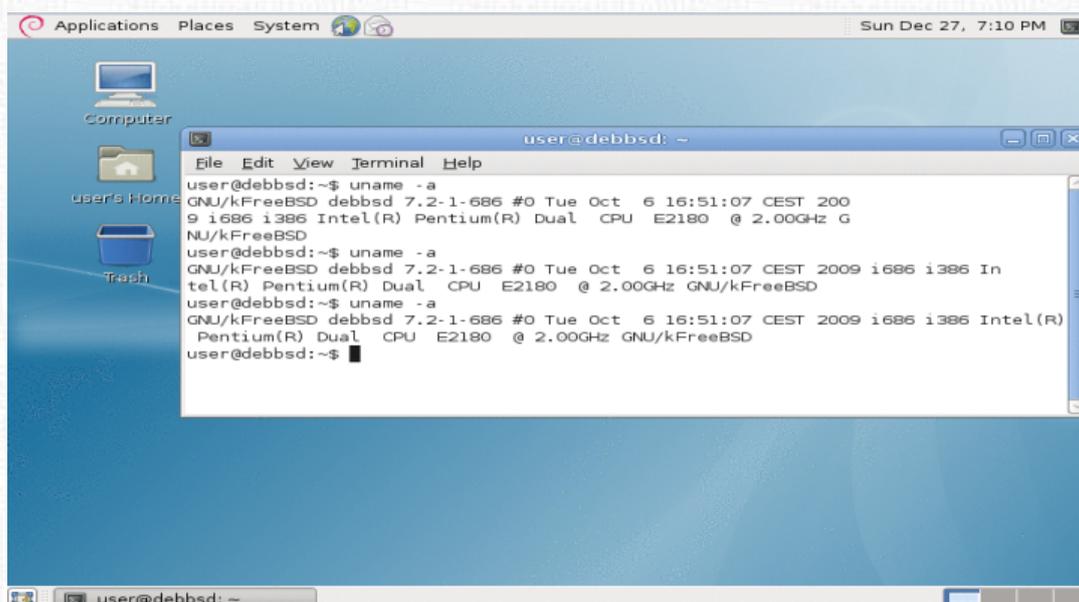
Gdm può dar qualche problema quindi meglio rimuoverlo

```
apt-get remove gdm
```

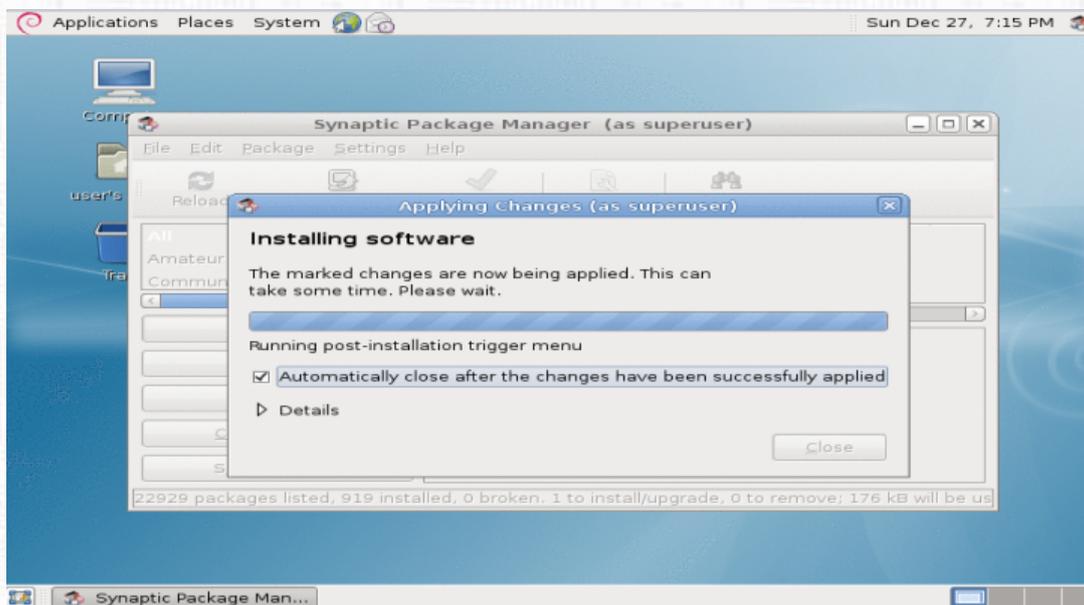
siamo pronti per far partire il server grafico

diamo il comando

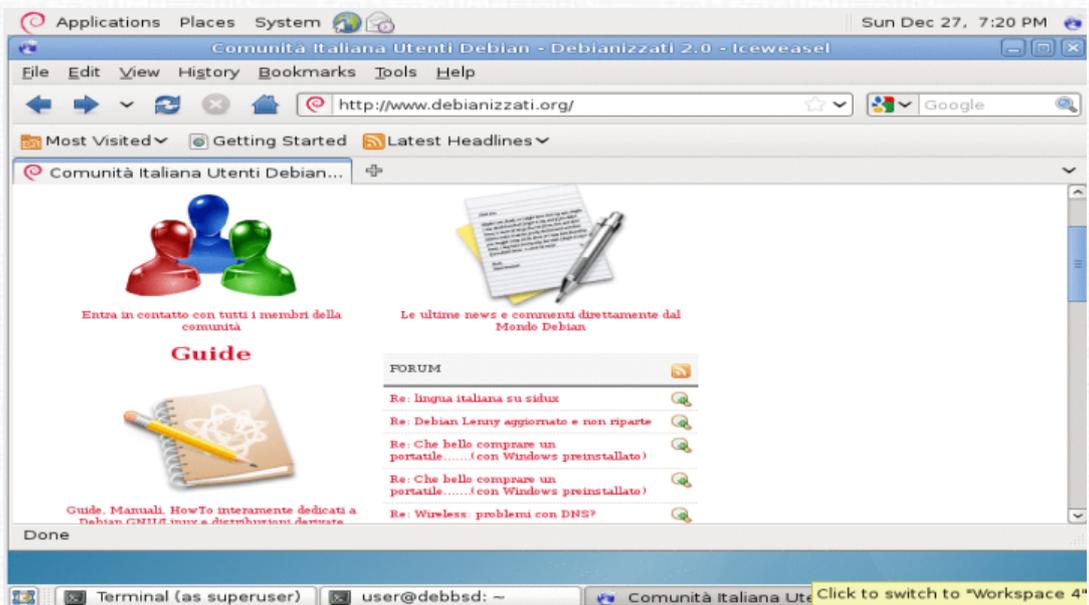
```
startx
```



..ed eccoci, finalmente, in Debian GNU/kfreebsd.



Synaptic al lavoro....



Iceweasel e debianizzati.org

3.1.3 Conclusioni

Debian GNU/kfreebsd, certo, in questo momento non è un sistema usabile, ma, confidando nell'ottimo lavoro del team che si sta occupando del port, siamo sicuri che il prodotto finale sarà certamente all'altezza dello standard Debian. Questo articolo vuol essere un'anteprima per coloro che vogliono cimentarsi nel testare questa nuova proposta.

Maggiori informazioni sono reperibili sul wiki

http://wiki.debian.org/Debian_GNU/kFreeBSD

in cui sono illustrati anche i perché

http://wiki.debian.org/Debian_GNU/kfreebsd_why

inoltre si possono seguire gli sviluppi e segnalare i vari bug iscrivendosi alla mailing list

<http://lists.debian.org/debian-bsd/>

Se non riuscite a trovare una iso funzionante, da qui ¹ potete scaricare la iso di cui ci siamo avvalsi per questo breve how-to.

Se poi siete pigri e volete partire subito, potete scaricare una immagine di un disco da usare con virtualbox

<http://tinyurl.com/ylemu3b>

le password sono **root** -- > **root** , **user** -- > **user** per far partire l'ambiente grafico dare il comando *startx* dopo aver fatto il login.

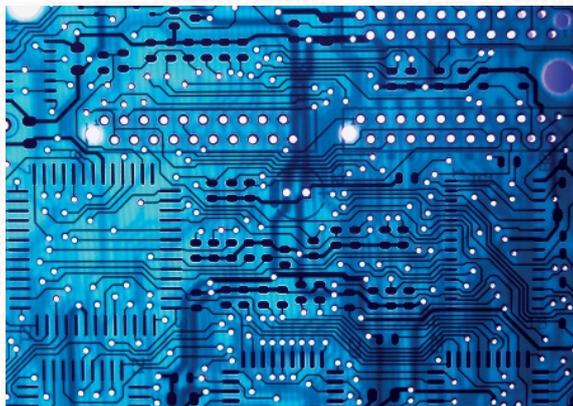
Debian il sistema operativo universale

<http://www.debian.org/>

¹<http://dl.dropbox.com/u/3247077/mini19dic.iso>

Capitolo 4

Hardware & Debian



Tutte le informazioni, esperienze, tuning sull'hardware e debian.

Debian, così come Linux, è una delle distribuzioni più versatili adattandosi a quasi ogni tipo di interfaccia possibile.

Ogni tanto è però necessario pensare un momento prima di riuscire a configurare il nostro sistema operativo per un certo tipo di hardware.

In questa sezione troverete i nostri esperimenti, così come consigli e hack.

4.1 Debian su android

4.1.1 Introduzione

Dopo i port su vari kernel, l'installazione sulle macchine più disparate, non è stato necessario aspettare molto tempo prima di vedere la nostra debian arrivare a conquistare il mondo dei dispositivi mobili quali telefoni cellulari e smartphone. E anche nel mondo dei cellulari l'installazione di debian è possibile sempre su più dispositivi, anche se appoggiandosi ai sistemi operativi con cui nascono tali dispositivi. Il sogno di debian come unico sistema operativo di un apparecchio resta al momento ancora negato, ma con il passare del tempo e la flessibilità di debian, quel giorno, forse, non sarà ancora per molto lontano (v. maemo).

Un grosso problema è però quello legato alla parte telefono, dove i chip sono spesso pieni di codice binario senza sorgenti, interrompendo la libertà del sistema.

Sistemi operativi liberi

Attualmente, come detto in precedenza, debian si lascia installare su vari sistemi operativi liberi (non prenderemo in considerazione sistemi proprietari, nonostante fra alcuni di essi è comunque possibile installare debian). In sostanza si tratta di port su architettura ARM EABI (armel) ¹ ² Qui di seguito una piccola panoramica sugli attuali sistemi operativi liberi che consentono l'installazione di debian:

- Maemo ³ ⁴: nato nel 2005, questo sistema ha come casa madre la multinazionale Nokia. Il sistema operativo è basato su debian GNU/Linux ed è in collaborazione con altri progetti opensource quali ad esempio GNOME, SQLite, gstreamer, Mozilla, ecc.. Utilizza le librerie GTK+ e da poco anche le librerie Qt. La licenza, nonostante la grande base di software libero, ha delle componenti proprietarie. L'intera debian ARM è installabile come applicazione sul sistema operativo. A differenza di altri sistemi operativi basati su (GNU)Linux, maemo ha il grande vantaggio di poter usufruire di molti pacchetti precompilati non modificati.

¹<http://www.debian.org/ports/arm/index.it.html>

² <http://wiki.debian.org/ArmEabiPort>

³<http://maemo.org/intro/>

⁴<http://en.wikipedia.org/wiki/Maemo>

- Openmoko ⁵: nato nel 2006, il progetto openmoko vuole offrire un sistema operativo per telefonia mobile completamente libero: la licenza non può che essere la GPL. "First International Computer (FIC)" è la casa madre. Dopo gli sviluppi iniziali il progetto sembrerebbe essere attualmente in crisi, in parte dovuto alla crisi stessa della casa madre, con importanti licenziamenti e tagli ai fondi. Il successore del noto "Neo FreeRunner", il "GTA03", è stato cancellato dai piani, mentre la ditta ha puntato su "WikiReader", una specie di enciclopedia elettronica basata su wikipedia. Su openmoko parrebbe ancora sia possibile installare l'intera debian ARM. Il progetto hackable1⁶ sembra si stia muovendo in questa direzione.
- Android ⁷: inizialmente concepito nel 2005, è venuto alla luce solo nel 2008 con il primo cellulare che lo sopporta in modo nativo. La casa madre è Google. Android è un sistema molto particolare e nonostante nel cuore ci sia Linux, non si può dire che ci assomigli molto; o per lo meno, non si può dire che assomigli molto a un sistema GNU/Linux, a partire ad esempio dall'assenza di X, senza tralasciare il legame stretto con determinate applicazioni di casa Google. Apparentemente la licenza è doppia: Apache 2.0 e GPLv2. Ciò nonostante, alcune parti del sistema sono più oscure di altre, così come la relazione di Google fra la comunità opensource e i loro interessi personali... In ogni caso, è possibile installare almeno una versione di debian ARM con qualche accorgimento e sfruttando un server VNC per utilizzare un'interfaccia grafica indipendente.

Debian ARM

Incominciamo a capire che cosa sia un'architettura di tipo ARM. Tratto da wikipedia:

L'architettura **ARM** (precedentemente **Advanced RISC Machine**, prima ancora **Acorn RISC Machine**) indica una famiglia di microprocessori **RISC** ⁸ a 32-bit utilizzata in una moltitudine di sistemi embedded. Grazie alle sue caratteristiche di basso consumo (rapportato alle prestazioni) l'architettura ARM domina il settore dei dispositivi mobili dove il risparmio energetico delle batterie è fondamentale.⁹

⁵http://wiki.openmoko.org/wiki/Main_Page/it

⁶http://www.hackable1.org/wiki/Main_Page

⁷<http://www.android.com/>

⁸http://it.wikipedia.org/wiki/Reduced_instruction_set_computer

⁹http://it.wikipedia.org/wiki/Architettura_ARM

Chi continuasse la lettura su wikipedia non si stupirebbe dunque di certo quando leggerebbe che la famiglia ARM copre il 75% mondiale dei processori a 32-bit in sistemi embedded. Sempre sulla stessa pagina troverete ancora una lista con i prodotti che utilizzano questo tipo di tecnologia; senza citarne alcuni, non faticherete a trovare una serie di ditte molto conosciute che si avvalgono dei processori ARM.

E debian? Non poteva di certo stare a guardare... Così, dopo il port del kernel ARM creato nel 1998, due anni più tardi nasceva in casa debian il port su ARM. Nel 2003 si passerà a ARM EABI e la vecchia ARM è considerata oggi deprecata.

Essendo - come detto in precedenza - l'architettura ARM presente sul 75% degli attuali apparecchi mobili, grazie a questo port è possibile installare debian su svariati dispositivi. Nonostante l'apparente facilità, non è però sempre operazione semplice adattare il sistema operativo ad un hardware specifico. In questo articolo vedremo come è possibile installare una debian ARM sfruttando il kernel di android, il sistema operativo mobile della Google. Le due distribuzioni si potranno dunque utilizzare in parallelo.

4.1.2 Preparazione

Per poter installare debian dobbiamo disporre di:

- l'*SDK (Software Development Kit)* di android
- una partizione `ext3` sulla scheda microSD
- un dispositivo sbloccato e *rooted* (alias, con la possibilità di utilizzare un utente "root")
- busybox, il coltellino svizzero del Linux *embedded*
- A2SD, per poter eseguire applicazioni dalla scheda microSD

Il SDK di android si potrà scaricare direttamente dal sito:<http://developer.android.com/sdk/index.html>.

È possibile scegliere l'SDK per sistemi operativi Windows, Mac OSx e Linux, per cui ci procuriamo quello per sistemi Linux.

Per creare una partizione `ext3` sulla scheda microSD possiamo utilizzare un tool qualsiasi, come ad esempio il comodissimo "gparted".

ATTENZIONE: la partizione `ext3` dovrà seguire la partizione standard in `fat32`; se già siete al lavoro, una piccola partizione di swap alla fine della scheda potrà essere utilizzata con dei firmware non-ufficiali (v. sotto) aumentando le prestazioni del sistema.

Per rendere idoneo il vostro dispositivo all'installazione di debian (e dunque sottostare ai prerequisiti sopra elencati), la cosa più semplice è quella di installare un firmware non-ufficiale che già include i prerequisiti richiesti. A seconda del modello del vostro cellulare potrei consigliarvi il firmware *CyanogenMod* o *MoDaCo*.

ATTENZIONE: i firmware non ufficiali sono specifici per modello; prima di installare qualcosa, oltre ad un sano backup, informatevi sull'hardware supportato dal firmware.

Per installare questi firmware vi rimando alla documentazione ufficiale presente in rete. In generale la procedura da eseguire consiste nel copiare l'immagine del nuovo firmware sulla scheda microSD, installare un bootloader con il quale avviare l'installer del firmware che provvederà a caricare nella ROM (*Read-Only Memory*) del dispositivo il firmware non-ufficiale scelto.

4.1.3 Installazione

Per installare la debian ARM utilizziamo il tool "debootstrap" presente nel nostro sistema operativo. Se non lo abbiamo ancora installato, lo potremo fare dai repository principali utilizzando il nostro gestore dei pacchetti preferito. Una volta fatto ciò montiamo la partizione ext3 della microSD sulla nostra macchina. Ad esempio:

```
# mount /dev/sdb2 /media/sdcard2
```

Dopo aver creato una directory "debian" in questa partizione possiamo scaricarvi l'immagine debian utilizzando debootstrap:

```
# debootstrap --verbose --arch armel --foreign lenny /media/sdcard2/debian \
    http://ftp.ch.debian.org/debian
```

dove con `--verbose` visualizziamo qualche dato sull'andamento dell'installazione (oramai siamo curiosi cronici ;-)); con `--arch armel` definiamo l'architettura di debian che vogliamo installare; con `--foreign` diciamo che l'architettura da cui partiamo (nei miei esperimenti amd64) non è uguale a quella d'arrivo (armel). Infine con `lenny` decidiamo che versione di debian installare; con `/media/sdcard2/debian` selezioniamo la directory dove verranno installati i pacchetti; per ultimo definiamo il mirror dal quale scaricare il tutto, in questo caso quello svizzero.

A questo punto possiamo smontare la partizione della scheda microSD e aprire una shell dell'adb (*Android Debug Bridge*) dalla SDK. Se abbiamo installato uno dei due firmware non-ufficiali sopracitati, la partizione in ext3 verrà automaticamente montata in `/system/sd`. Altrimenti dovremo farlo noi manualmente con:

```
# busybox mount /dev/block/mmcblk0p2 /system/sd
```

Come abbiamo potuto osservare si fa l'uso di busybox¹⁰, chiamato il coltellino svizzero del Linux *embedded*. BusyBox comprende varie utility UNIX in un unico eseguibile di piccole dimensioni, come ad esempio `ls`, `cat`, `chroot`, `mount`,... e tante altre ancora. Per la lista completa dei comandi supportati si consulti il *man page*. Quest'ultimi hanno generalmente meno opzioni rispetto agli originali; tuttavia, le opzioni che sono incluse sono del tutto fedeli a quelle originali. BusyBox risulta dunque essere un ambiente abbastanza completo per qualsiasi sistema di piccole dimensioni o *embedded*, il tutto come detto racchiuso in un unico eseguibile.

Tornando alla nostra shell dell'adb, esportiamo ora le variabili relative al *path* (riferite a debian), al terminale e alla home di root:

```
export PATH=/usr/bin:/usr/sbin:/bin:$PATH
export TERM=linux
export HOME=/root
```

A questo punto possiamo terminare l'installazione con:

```
# busybox chroot /system/sd/debian /debootstrap/debootstrap --second-stage
```

Come abbiamo potuto notare, si tratta di una *chroot jail*: una specie di spazio (in inglese *jail*, prigione) nel quale si definisce la *root* / del sistema in `/system/sd/debian` (tramite `chroot`) e dalla quale si lancia *debootstrap* per concludere l'installazione. Si noti che una volta definita la *root*, la directory *debootstrap* di `/system/sd/debian` sarà presente nella *root* / del sistema, la *chroot jail*, e verrà chiamata */debootstrap*.

Per lanciare debian daremo ora:

```
# busybox chroot /system/sd/debian /bin/bash
```

¹⁰<http://www.busybox.net/>

Il prompt:

```
localhost:/#
```

ci testimonierà di trovarci nell'ambiente della nostra debian.

Piccola pausa di riflessione

Per rendere ulteriormente utilizzabile la nostra debian dovremo ancora eseguire qualche operazione. Prima di tutto è però necessaria una riflessione su quanto abbiamo effettuato finora. L'installazione mediante *debootstrap*, tool molto pratico proprio per sistemi in *chroot jail*, non prevede l'installazione né di un kernel né di un bootloader. Non ci credete? Incominciamo a scrutare la directory di *boot*:

```
localhost:/# ls boot
localhost:/#
```

alias, un grande vuoto: né kernel né bootloader. Di conseguenza, niente moduli del kernel...

```
localhost:/# ls lib/modules
localhost:/#
```

Ci credete ora? Io sì... ma come può la nostra debian funzionare senza un kernel? Mistero... Ma diamo ancora un'occhiata... forse qualche strumento per scrutare il kernel c'è... ma certo, *uname*!

```
localhost:/# uname -r
2.6.27-mck-teknologist-1.9
```

Ebbene sì, il kernel che è utilizzato in questo caso da debian è il kernel di android, nel mio caso un kernel Linux 2.6.27 versione 1.9 sviluppato da teknologist. Altrimenti detto, è il kernel che viene associato al firmware non-ufficiale MoDaCo. Ma basta questo per far funzionare la nostra debian? Beh... penso di sì... quasi quasi adesso che è tutto a posto, grazie a *top* do una bella occhiata a quanto sta macinando il mio sistema:

```
localhost:/# top
Error: /proc must be mounted
  To mount /proc at boot you need an /etc/fstab line like:
    /proc /proc proc defaults
  In the meantime, run "mount /proc /proc -t proc"
```

??? Proc non è montato? Va beh, proviamo a seguire le istruzioni e montare il file system:

```
mount /proc /proc -t proc
```

L'immagine non ve la illustro, ma vi posso garantire che ora funziona :-). Chissà che ci sarà ancora da fare...

4.1.4 Configurazione

Dopo una piccola pausa, forse un po' teatrale (sperando però di aver illustrato i concetti di base della *chroot jail*), torniamo alla configurazione del sistema. Come abbiamo già visto, per poter utilizzare appieno il nostro sistema, oltre al file system *proc* dovremo ancora montare *devpts* e *sysfs*:

```
mount /proc /proc -t proc
mount /devpts /dev/pts -t devpts
mount /sysfs /sys -t sysfs
```

(nonostante con *mount* sia consuetudine definire il tipo di file system subito dopo il comando stesso [`mount -t <filesystem> ...`] ho seguito l'esempio di sopra suggerito dal sistema).

Partizioni

Sappiamo che le partizioni sono montate nel sistema in fase di boot dal kernel, ma poiché il kernel utilizzato è già avviato (è quello di android) dobbiamo eseguire queste operazioni in modo manuale. Il noto */etc/fstab* non è però utilizzabile per i motivi appena citati ed in effetti lo troviamo vuoto. Inoltre sarebbe interessante utilizzare, oltre ai file system *proc*,

sysfs e *devpts*, anche le partizioni già montate da android. Per fare ciò utilizzeremo un piccolo trucco: nel file */etc/mtab* sono presenti le partizioni riconosciute dal sistema come montate ed in che modo. Inoltre in */proc/mounts* sono presenti le partizioni effettivamente montate. Possiamo dunque eliminare il *etc/mtab* originale e sostituirlo con un link che punta a */proc/mount*:

```
localhost:/# rm etc/mtab
localhost:/# ln -s /proc/mounts /etc/mtab
```

Ora basterà creare delle directory per montare le partizioni principali di android su debian:

```
localhost:/# mkdir /media/sdcard
localhost:/# mkdir /media/system
localhost:/# mkdir /media/data
localhost:/# mkdir /media/dev
```

Per effettuare ora un *bind* alle partizioni relative è però necessario uscire da debian (semplicemente con un *exit*) ed eseguire i comandi da una shell di android, ad esempio dall'*adb* (proprio con l'*adb* assicuriamoci che la partizione */sdcard* sia montata altrimenti avremo chiaramente un errore):

```
busybox mount --bind /sdcard /system/sd/debian/media/sdcard
busybox mount --bind /system /system/sd/debian/media/system
busybox mount --bind /data /system/sd/debian/media/data
busybox mount --bind /dev /system/sd/debian/media/dev
```

Rete e Repository

Basandoci sulla configurazione della rete di android (vi ricordo che il kernel risiede lì) bisogna ancora configurare i DNS e i repository. Per quanto riguarda il primo, possiamo semplicemente inserire i DNS del nostro provider; in ogni caso, per quanto riguarda l'UMTS ho avuto qualche problema e mi sono trovato meglio con gli openDNS. Inseriamo dunque quest'ultimi nel file, ad esempio con:

```
echo "nameserver 208.67.222.222" > /etc/resolv.conf
echo "nameserver 208.67.220.220" >> /etc/resolv.conf
```

Allo stesso modo inseriamo l'unico repository in */etc/apt/sources.list*:

```
echo "deb http://ftp.ch.debian.org/debian lenny main" > /etc/apt/sources.list
```

scegliendo, eventualmente, un mirror più vicino a voi. A questo punto possiamo subito tastare la rete con un bel `apt-get update` (nonostante `aptitude`, il mio preferito, funzioni benissimo, `apt-get` è, in questo ambiente, un po' più veloce).

boot

Abbiamo già visto tutto ciò che ci serve per avviare il sistema. Un comodo script riassumerà quanto svolto fino ad ora:

```
#!/system/bin/sh

# esportiamo le variabili di sistema
export PATH=/usr/bin:/usr/local/bin:/usr/sbin:/bin:$PATH
export TERM=linux
export HOME=/root

# montiamo i file system necessari
busybox mount /proc /system/sd/debian/proc -t proc
busybox mount /devpts /system/sd/debian/dev/pts -t devpts
busybox mount /sysfs /system/sd/debian/sys -t sysfs

# binding delle partizioni d'android a debian
busybox mount --bind /sdcard /system/sd/debian/media/sdcard
busybox mount --bind /system /system/sd/debian/media/system
busybox mount --bind /data /system/sd/debian/media/data
busybox mount --bind /dev /system/sd/debian/media/dev

# prima di avviare, un bel logo è quello che ci vuole
clear
```

```

echo " "
echo "      .':::;:'''. "
echo "      'lxxxxxxxxxxxxdol,. "
echo "      .okkkkl,... ..,lkkko' "
echo "      okkkd;          ,xkkko "
echo "      .dkx;          ckkxd "
echo "      'kk:          ','''. ckd,. "
echo "      .xkl          .c;.      .      kk; "
echo "      ckd          .o          xkl "
echo "      :k:          d'          kk, "
echo "      :k,          o:          .      ,kc. "
echo "      :k;          'x,          .      :k; "
echo "      'kx          .'ol,.      .,cd;. "
echo "      dk;          .'colc:c:,. "
echo "      .kkk.          . "
echo "      .kkl "
echo "      .dkc "
echo "      ;kx. "
echo "      ;dd, "
echo "      .:ll, "
echo "      .... "
echo " "
echo "      BENVENUTO IN DEBIAN ! "
echo " "

# avviamo debian, alias apriamo una bash nella chroot jail
busybox chroot /system/sd/debian /bin/bash

```

Si noti come la shell ufficiale di android sia in `/system/bin/sh` (e non la bash). Possiamo salvare ora lo script: lo nominiamo, ad esempio, `ds` (per *debian start* o qualcosa di simile, in ogni caso niente di troppo noioso da scrivere sulla tastiera virtuale), lo memorizziamo in `/system/bin` (o dove preferite, comunque dove sia possibile lanciare un eseguibile; ricordatevi che dall'adb dovete eseguire un remount per avere i permessi di scrittura in `/system`) ed infine assicuriamoci di avere i permessi d'esecuzione ottenendoli eventualmente con `chmod`.

A questo punto, per avviare debian basterà digitare (da root) *ds* da un qualsiasi terminale in android. Per uscire, come già detto basterà un *exit*. Avendo montato i file system *proc*, *devpts* e *sysfs*, se dovessimo riaprire una sessione con il nostro script avremmo degli errori in quanto quest'ultimi non sono stati smontati. Per ovviare al problema possiamo utilizzare ancora un semplice script:

```
#!/bin/bash
umount /devpts
umount /sysfs
umount /proc
echo "Arresto il sistema..."
```

notare che */proc* viene smontato per ultimo in quanto serve agli altri due. Possiamo salvare lo script nominandolo, ad esempio, *dk* (*debian kill*) e dopo averlo reso eseguibile lo possiamo spostare in */usr/local/bin*. A questo punto per uscire dalla sessione possiamo associare il nostro script ad *exit* aggiungendo alla fine del file */etc/bash.bashrc* le seguenti linee:

```
# associamo lo script dk ad exit
alias exit='dk && exit'
```

Per uscire, digitando *exit* verrà eseguito lo script *dk* smontando i file system montati all'avvio.

ATTENZIONE: se volete installare un server grafico (v. sotto) lo script *dk* si può modificare per implementare la chiusura di quest'ultimo all'uscita dal sistema. Lo script andrà dunque modificato; per i dettagli si veda la fine del prossimo capitolo.

Server grafico (X)

Android purtroppo non si basa sul server X. Questo significa che non è possibile installare il server con la nostra debian. Possiamo però avviare una sessione grafica tramite un server VNC (su debian) al quale possiamo poi collegarci con un client da android stesso.

Dapprima ci procuriamo il client dall'android market: nel mio caso ho utilizzato *android-vnc-viewer*. Per il lato server ho utilizzato *tightvncserver* (preferito al più classico *vncserver* per l'ottimizzazione per banda corta). Una volta installato, sempre con il fedelissimo Apt, possiamo avviare il server con:

```
# export USER=root
# vncserver -geometry 480x320
```

Al primo avvio ci verrà richiesta una password per accedere al server; dopo averla impostata ci sarà ancora richiesto se vogliamo impostare una password *view-only*. Lascio a voi la scelta della vostra configurazione; personalmente mi sono accontentato di una password unica ;-). Sempre con il primo avvio verrà, inoltre, creata la directory */root/.vnc/xstartup* con le impostazioni relative al server grafico. Possiamo ora killare la sessione grafica con:

```
# vncserver -kill :1
```

per installare un *window manager* da utilizzare come desktop. Il kill non è fondamentale, ma a seconda del *window-manager* installato dovremo modificare la configurazione di */root/.vnc/xstartup*, creato, come detto, al primo avvio del server. Dopo aver effettuato qualche test, la mia scelta è caduta su *icewm* in quanto molto leggero e piuttosto indipendente dal tasto destro del mouse, difficile da simulare con il client vnc. Installiamo dunque questo pacchetto utilizzando sempre Apt. Fatto ciò possiamo ora modificare il file */root/.vnc/xstartup* adattandolo alla nostra scelta. Commentiamo, dunque, la linea relativa a *xsession* di *X11* e aggiungiamo il nostro *icewm*. Il risultato sarà qualcosa di simile:

```
#!/bin/sh
xrdb $HOME/.Xresources
xsetroot -solid grey
#x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
#x-window-manager &
#/etc/X11/Xsession
icewm &
```

Fatto ciò possiamo ora lanciare la nostra sessione grafica. Per facilitare l'operazione ci aiutiamo ancora con un piccolo script:

```
#!/bin/bash
export USER=root
vncserver -geometry 480x320
```

lo chiamiamo ad esempio *sx* (*start x*) e lo salviamo ad esempio in */usr/local/bin*, e gli attribuiamo i permessi d'esecuzione come sempre con *chmod*. A questo punto lanciamo la sessione grafica digitando semplicemente *sx*. I più attenti noteranno subito un errore all'avvio del server:

```
xauth: (argv):1: bad display name "localhost:1" in "add" command
```

Questo perché non abbiamo un host definito nel sistema associato a localhost. Creiamo dunque il file */etc/hosts* ed inseriamo la seguente riga:

```
127.0.0.1    localhost
```

Salviamo il file e riavviamo il server. L'errore non dovrebbe essere più presente.

Per avviare il client su android, usciamo dal terminale (lasciandolo in *background*, a seconda del dispositivo che stiamo utilizzando, nel mio caso premendo il pulsante *home*) ed avviamo la connessione con le seguenti impostazioni:

- *Nickname*: root
- *Password*: la password che abbiamo impostato all'installazione di *tightvncserver* su debian
- *Indirizzo*: localhost
- *Porta*: 5901
- *Colori*: 24-bit

Per chiudere la sessione grafica torniamo nel terminale di debian (sempre nel mio caso, premendo il pulsante *home* e riselezionando il terminale dal quale abbiamo avviato debian) e digitiamo:

```
# vncserver -kill :1
```

accertandoci, comunque, di aver esportato la variabile `USER` associata a `root`. Ancora una volta, usando un comodo script avremo:

```
#!/bin/bash
export USER=root
vncserver -kill :1
echo "Chiusura della sessione grafica..."
```

Possiamo salvare tale script come `kx` (*kill x*) in `/usr/local/bin`, e attribuirgli i permessi d'esecuzione. Per fermare la sessione grafica possiamo dunque semplicemente digitare `kx`.

'*NOTA BENE*': se ci dimenticassimo di chiudere la sessione, al prossimo avvio del sistema e soprattutto al prossimo avvio della sezione grafica verrà avviato un nuovo display definito come `localhost:2` (al posto di `localhost:1`); oltre a rallentare tutto il sistema (ci saranno due sessioni aperte contemporaneamente) lo script per terminare la sessione sarà valido solo per il primo *display*. Per chiudere il secondo dovremo dare dunque i comandi:

```
# export USER=root
# vncserver -kill :2
```

Per ovviare a tale problema possiamo ora aggiungere un controllo al momento della chiusura del sistema. Il server vnc crea un file `localhost:1.pid` all'apertura di una sessione. A questo punto basterà verificare se il file esiste e nel caso, chiudere la sessione con lo script `kx`. Lo script finale `dk` risulterà dunque:

```
#!/bin/bash

# controlla se una sessione grafica e' attiva; in caso affermativo chiudila
if [ -e /root/.vnc/localhost:1.pid ]; then
    export USER=root
    echo "Chiusura della sessione grafica..."
```

```
vncserver -kill :1
fi

umount /devpts
umount /sysfs
umount /proc

echo "Arresto il sistema..."
```

Allo stesso modo possiamo controllare se esiste già una sessione aperta al momento di lanciare *sx* e in caso affermativo evitare di aprire una nuova sessione. Lo script *sx* finale risulterebbe come il seguente:

```
#!/bin/bash

# controlla se una sessione grafica e' gia' attiva; in caso affermativo evita di \
#                                       avviarne un'altra uscendo dallo script
if [ -e /root/.vnc/localhost:1.pid ]; then
    echo "Una sessione grafica sul display localhost:1 e' al momento gia' attiva! \
        (avvia semplicemente il client vnc da android)"

    exit 0
fi

# nel caso non esiste nessuna sessione grafica attiva, avviane una
export USER=root
vncserver -geometry 480x320
```

Il client VNC di android

Senza entrare troppo nei dettagli ecco un paio di consigli su come muoversi nell'ambiente grafico tramite il client in questione (*android-vnc-viewer versione 0.3.1*):

- Muovere il mouse con il *touchscreen*: dal menu principale dapprima selezioniamo lo Scaling e lo impostiamo come 1:1. Fatto ciò, scegliamo Input Mode e selezioniamo Mouse Pointer Control Mode. A questo punto possiamo muovere il mouse con un

dito sul display. Il click si farà con un *tapping* (ossia battendo un colpetto con un dito sul monitor). Il *trackball* si potrà utilizzare come *scroll*, ma non come click (quest'ultimo vi farà passare al *Desktop Panning Mode*, praticamente inutilizzabile).

- Per inserire del testo tramite una *soft keyboard* (tastiera virtuale): utilizzate l'opzione *Send Text* dal menu principale dopo aver posizionato il cursore nel campo desiderato.

4.1.5 Screenshot

Dopo un articolo sterile d'immagini, ecco infine una raccolta su alcune parti del sistema sul mio dispositivo (la forma arrotondata delle foto è dovuta all'effetto macro, fotografando il monitor da vicino):



Figura 4.1: Il boot di debian dopo un update con Apt

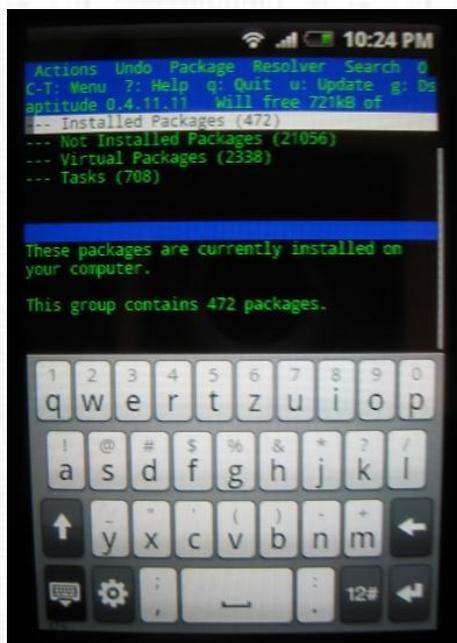


Figura 4.2: Anche le *ncurses* fanno la propria bella figura! Qui l'interfaccia di *aptitude* con la *soft keyboard* - la tastiera virtuale - di android

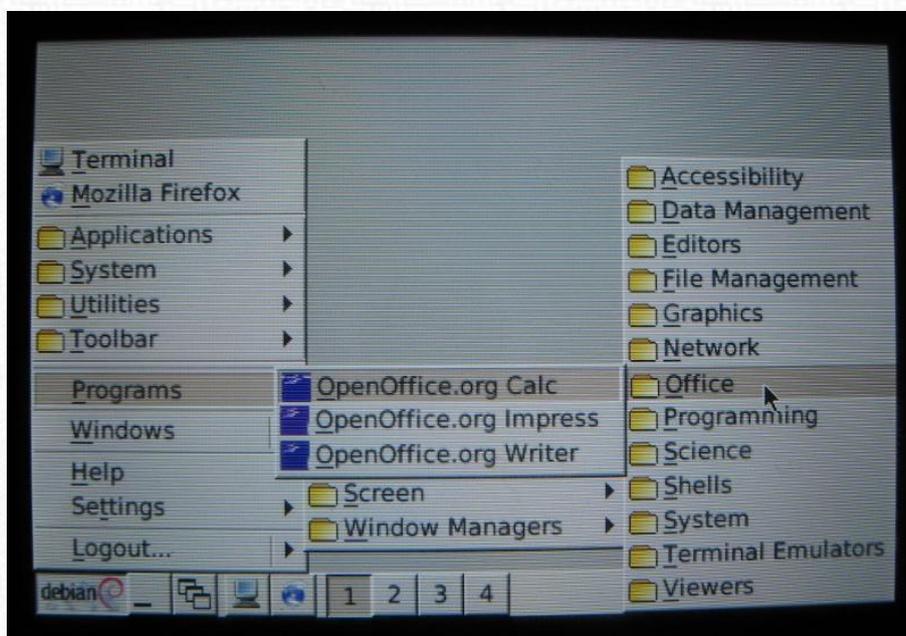


Figura 4.3: Passati all'interfaccia grafica vera e propria, ecco come visualizzerete *icewm* con il menu a tendina

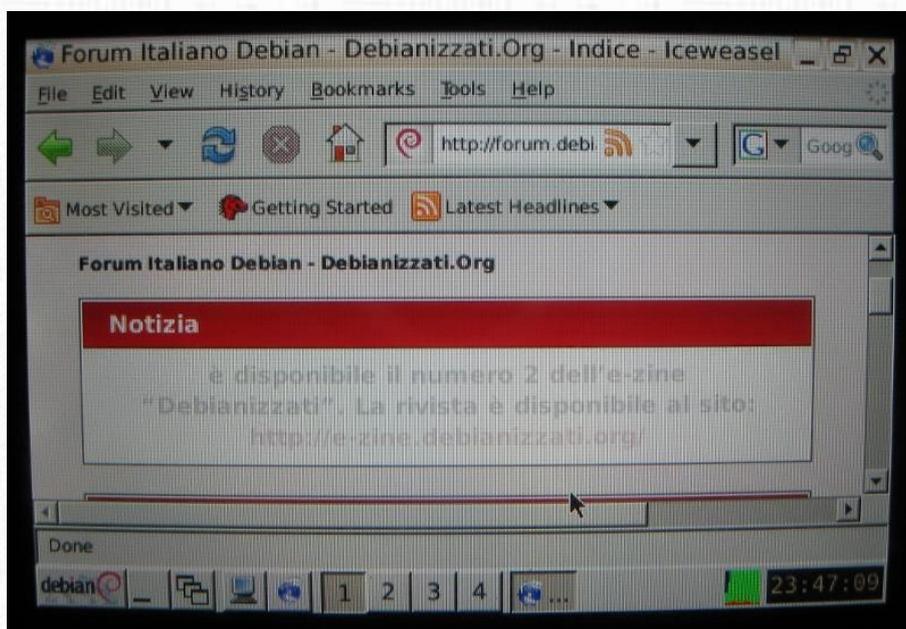


Figura 4.4: Seppure molto più lento e meno pratico del web browser di android, iceweasel in debian funziona alla perfezione

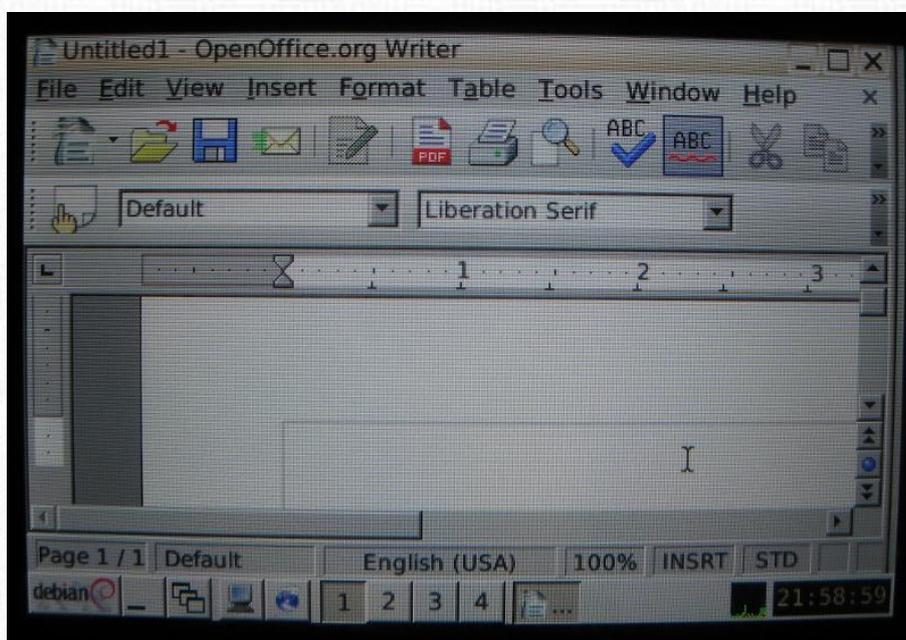


Figura 4.5: Ancora a un livello più complesso, la suite completa di openoffice si lascia installare ed utilizzare. Qualche problema si incontra soprattutto ai limiti di visualizzazione delle finestre relativi al client vnc; un'ottimizzazione per display ridotti farebbe comodo, ma per qualche piccola operazione non ci sono problemi. Nell'esempio, openoffice writer

4.1.6 Conclusioni

Come abbiamo potuto osservare, la debian armel si lascia installare abbastanza bene a fianco di android. Il *wrap* per utilizzare un ambiente grafico tramite un server vnc amplia ancora di più le possibilità di sfruttamento del sistema.

La parallelizzazione con android tramite una *chroot jail* permette di sfruttare appieno le funzionalità di debian, non perdendo però l'aspetto telefonico del cellulare; alla fine serve per questo o no ;-)?

In futuro, per rendere debian indipendente da android servirà in principio solo un kernel che si adatti all'hardware in questione e un'interfaccia grafica che si possa appoggiare a un server X; anche questo, strettamente legato all'hardware. Infine, non dimentichiamoci di trovare un bootloader che possa avviare il tutto (per quest'ultimo ci sono sicuramente buoni propositi già pensando alle immagini di recupero quali ad esempio la *Cyanogen Recovery Image*).

Ma è così complicato quest'hardware? Certo che un processore non è che una piccola parte appunto; ci accorgeremo ancora di più della situazione quando un lspci non ci darà nessun risultato. Grazie comunque a *lshw* possiamo farci un'idea:

```
localhost
  description: Computer
  width: 32 bits
*-core
  description: Motherboard
  physical id: 0
*-memory
  description: System memory
  physical id: 0
  size: 191MiB
*-cpu
  physical id: 1
  bus info: cpu@0
  size: 528MHz
  capacity: 528MHz
  capabilities: cpufreq
*-network:0 DISABLED
  description: Ethernet interface
```

```
physical id: 1
logical name: dummy0
serial: 92:7d:97:e2:7e:e1
capabilities: ethernet physical
configuration: broadcast=yes
*-network:1 DISABLED
description: Ethernet interface
physical id: 2
logical name: rmnet0
serial: ee:92:6a:ef:09:85
capabilities: ethernet physical
configuration: broadcast=yes ip=10.162.125.129 multicast=yes
*-network:2 DISABLED
description: Ethernet interface
physical id: 3
logical name: rmnet1
serial: 1e:d9:f5:d6:c2:f8
capabilities: ethernet physical
configuration: broadcast=yes multicast=yes
*-network:3 DISABLED
description: Ethernet interface
physical id: 4
logical name: rmnet2
serial: 92:1b:89:6e:9a:ae
capabilities: ethernet physical
configuration: broadcast=yes multicast=yes
*-network:4 DISABLED
description: Ethernet interface
physical id: 5
logical name: usb0
serial: 9a:00:77:11:2b:7d
capabilities: ethernet physical
configuration: broadcast=yes multicast=yes
*-network:5
description: Ethernet interface
physical id: 6
logical name: tiwlan0
serial: 00:23:76:25:5a:a5
```

```
capabilities: ethernet physical
configuration: broadcast=yes ip=192.168.1.20 multicast=yes
```

Sarà da subito evidente l'assenza di interfacce: a parte la CPU, la RAM e le interfacce di rete fatichiamo a trovare qualcosa. Probabilmente il tutto è racchiuso in quella *Motherboard* iniziale, ma non sarà facile lavorare con un sistema di questo tipo.

In ogni caso, i test con questo cellulare continueranno e non escludo che in un prossimo numero possano uscire ulteriori articoli, magari proprio su un'installazione nativa di debian su questo tipo di dispositivi ;-). Al tempo l'ardua sentenza...

...al momento, Happy Hacking!

Webografia

<http://www.saurik.com/id/1>

http://bayleshanks.com/wiki.pl?tips-computer-android-g1_debian_cyanogenMod

Capitolo 5

Tips & Tricks



Che aggiungere, soffiare e trucchi ;-)

Tutto ciò di utile che non rientra nelle altre categorie.

In questa sezione troverete articoli che riguardano questioni settoriali e/o molto specifiche su di un argomento in particolare.

5.1 Vivere l'Emacs

Emacs è l'editor in tempo reale, estensibile, personalizzabile, auto-documentante e visuale.

manuale di GNU Emacs 23.1

5.1.1 Una brevissima storia

La prima versione di Emacs fu scritta 1976 da un certo 23enne di nome Richard M. Stallman presso il MIT Computer Science and Artificial Intelligence Laboratory. Inizialmente consisteva in set di macro per TECO (Tape Editor and COrrector che divenne in seguito Text Editor and COrrector) un potente editor di testo che girava al MIT su macchine PDP-6 e PDP-10 dotate del sistema operativo ITS (Incompatible Timesharing System). TECO aveva modalità diverse a seconda che si volesse inserire, modificare o visualizzare il testo e per effettuare ognuna di queste operazioni era necessario scrivere una serie di istruzioni nel suo linguaggio di programmazione (di fatto dei veri e propri programmi). Questo comportava il fatto che non era possibile visualizzare il testo nello stesso momento in cui questo veniva inserito.

Una delle prime funzionalità aggiunte da Stallman fu la modifica alla modalità modifica-visualizzazione di TECO che permetteva all'utente di definire delle scorciatoie da tastiera per eseguire determinati programmi TECO. Di fatto tramite questa macro, diventava possibile aggiornare il video mediante una semplice combinazione di tasti!

Questa e tutte le altre macro che i vari utenti di TECO scrissero in quel periodo furono organizzate e documentate da Stallman ed andarono a formare il primissimo EMACS, sigla che stava per Editing MACroS.

Negli anni vennero fatti port di EMACS su vari sistemi operativi fino a sbarcare su Unix nel 1981 per mano di James Gosling: stiamo parlando di Gosling Emacs la prima versione di emacs scritta in C ¹. Nel 1984 Stallman iniziò a lavorare su GNU Emacs proprio per produrre una versione libera, alternativa al Gosling Emacs, che era software proprietario. Stallman riuscì a breve nel suo intento e già nel 1985 rilasciò la prima versione del suo GNU Emacs. Contemporaneamente prendeva vita il progetto GNU.

¹http://en.wikipedia.org/wiki/Gosling_Emacs

GNU Emacs 23.1

Di passi avanti ne sono stati fatti per arrivare a questa versione rilasciata il 29 giugno 2009. Nonostante tutti questi cambiamenti il cuore di GNU Emacs (da ora in poi detto più semplicemente Emacs) è rimasto lo stesso ed è l'interprete Emacs Lisp. Emacs usa quindi un dialetto del linguaggio di programmazione LISP² e in questo linguaggio sono scritte anche le sue innumerevoli estensioni.

Grazie all'elevato numero di estensioni esistenti e alla relativa semplicità con cui un utente ben documentato possa scriverne di nuove, Emacs diviene estremamente versatile e si può adattare facilmente alle proprie esigenze. Naturalmente tutta la documentazione completa è resa disponibile dal progetto GNU e rappresenta la lettura obbligatoria per chiunque intenda cimentarsi nella programmazione in Emacs Lisp³.

La tendenza generale dei sistemi GNU/Linux degli ultimi anni è stata quella di offrire all'utente un ambiente di lavoro sempre più confortevole e eye-candy. Neppure Emacs è sfuggito a questa tendenza. Le novità più interessanti della versione 23 riguardano proprio l'integrazione nell'ambiente X.

Emacs oggi ha un'interfaccia grafica scritta in GTK+ che si integra perfettamente nei nostri desktop e, novità tra le novità, offre il rendering dei font tramite fontconfig e Xft (X FreeType interface library). Quest'ultima caratteristica ci permette quindi di usare i font FreeType e di godere dell'antialiasing.

Se tutto ciò non ci interessasse possiamo comunque continuare ad usare la versione classica, che non ha bisogno di X per poter funzionare.

5.1.2 Installare e configurare Emacs 23 in Debian Lenny

Dato che questa versione di Emacs è stata rilasciata dopo il freeze di Debian Lenny, non la ritroviamo tra i pacchetti ufficiali. Fortunatamente è già stata introdotta nei backports per Lenny, pertanto la sua installazione è facile e veloce. Basta abilitare i repository, come spiegato sul sito ufficiale⁴ ed installare Emacs dotato dell'interfaccia grafica GTK+ via aptitude:

²<http://it.wikipedia.org/wiki/Lisp>

³<http://www.gnu.org/software/emacs/manual/elisp.html>

⁴<http://backports.org/dokuwiki/doku.php?id=instructions>

```
# aptitude -t lenny-backports install emacs23
```

Rendering & antialiasing

Nella versione prelevata dai backports il rendering **Xft** viene abilitato di default. Se si usassero altre versioni potrebbe essere necessario aggiungere al file `~/.Xresources` la seguente riga:

```
Emacs.FontBackend: xft
```

Il file di configurazione di emacs è `~/.emacs`. Eventualmente si possono inserire più file di configurazione nella directory `~/.emacs.d` che funziona con lo stesso principio di `/etc/apt/apt.conf.d` per APT.

(L'argomento personalizzare Emacs è davvero troppo ampio per essere trattato in questo articolo. Per chi fosse interessato, può leggere il capitolo 32, Customization del manuale)

Possiamo specificare in `~/.emacs` il carattere che Emacs userà in modo predefinito. Possiamo usare uno qualsiasi di quelli installati nel nostro sistema. Per esempio se volessimo utilizzare il carattere *Dejavu Sans Mono*, dimensione 12, dovremmo specificare:

```
(if (display-graphic-p)
  (add-hook 'window-setup-hook
    (lambda nil
      (set-default-font "Dejavu Sans Mono-12")
      (set-fontset-font (frame-parameter nil 'font)
        'han '("cwTeXHeiBold" . "unicode-bmp"))
      )))
```

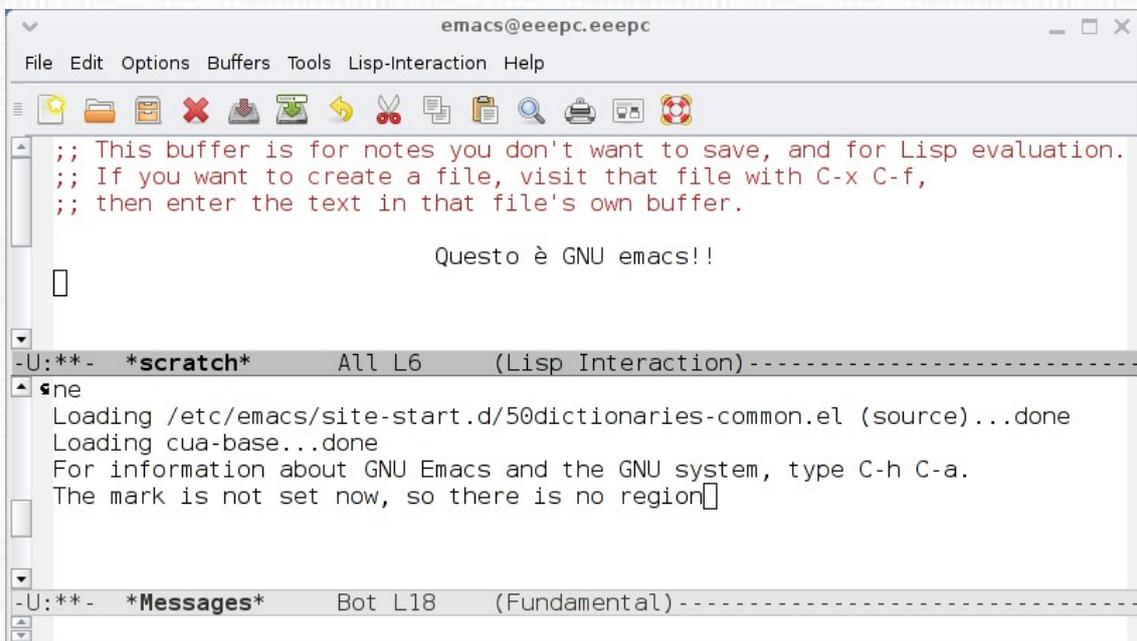
Le magie dell'interfaccia grafica ci permettono di selezionare il carattere preferito (tipo e dimensione) anche da una sessione di Emacs: *Option* → *Set default Font...*

Adesso Emacs è pronto per essere usato.

5.1.3 La faccia di Emacs

Cerchiamo di fare chiarezza sull'organizzazione della finestra che ci si presenta all'apertura di una nuova istanza di Emacs. Questo è il primo passo per entrare nel mondo di questo programma.

Innanzitutto precisiamo che il testo visualizzato viene chiamato buffer. Questo non è il file che abbiamo aperto ma una sua copia residente in RAM. Ogni modifica effettuata sul buffer non viene scritta sul file originale fino a quando non decidiamo di salvare su disco.



1. La parte principale è dedicata alla visualizzazione del testo inserito: questa zona viene detta **point area**. Il cursore lampeggiante ci indica la posizione dove andremo ad inserire il testo.
2. **mode line**: ci dà informazioni riguardo al file aperto e non solo. Si presenta nel seguente modo:

```
-U:**- emacs.tex 41% (108,55) (PDFTeX Fly)-----
```

```
-cs:ch dir buf pos line (major minor)-----
```

Passando il mouse sopra i vari elementi o cliccandoci sopra si ottengono maggiori informazioni.

- **cs** - rappresenta il set di caratteri impiegati per il buffer (di fatto il file) visualizzato (esempio UTF8), nonché il carattere utilizzato come endline. Cliccando sopra con il mouse, vengono date maggiori informazioni
 - **ch** - questa stringa di due caratteri ci fornisce diverse informazioni ed assume i seguenti valori:
 - – il buffer è modificabile (ad esempio perché si hanno i premessi per modificarlo) ma ancora è identico alla copia su disco.
 - ****** il buffer è modificabile (ad esempio perché si hanno i premessi per modificarlo) ed è diverso dalla copia su disco (ovvero è stato modificato).
 - **%%** il buffer è in modalità *read-only* ed è ancora identico alla copia su disco.
 - **%*** il buffer in modalità *read-only* ed è stato modificato: se non vogliamo perdere le modifiche dovremo obbligatoriamente salvare il buffer su un altro file.
 - **dir** - viene visualizzato normalmente come un segno “-”, passando sopra il cursore del mouse ci viene indicata la directory di lavoro.
 - **buf** - il nome del file che stiamo visualizzando
 - **pos** - indica la % di linee di testo che sono presenti prima della schermata che stiamo visualizzando.
 - **line** - indica il numero di linea nella quale si trova il cursore. Con un click si può impostare che venga visualizzato solo il numero di colonna o entrambi nel fomato (*lin,col*)
- major minor** - indica il modo di editing (*major*: ad esempio PDFtex, FORTRAN, C, Text, ecc..) che stiamo utilizzando (indicato con major) ed eventualmente delle features opzionali che abbiamo attivate (*minor*: ad esempio il modo *flyspell* di correzioni del testo).

3. **Echo area** - L'ultima linea in basso sulla finestra. Vengono riportate in questa area diversi tipi di informazioni, come ad esempio la sequenza di comandi da tastiera che stiamo inserendo e gli eventuali autocompletamenti. Se ci clicchiamo sopra visualizziamo il buffer ***Messages*** che rappresenta la cronologia dei messaggi informativi visualizzati nell'echo area.

5.1.4 Uso di emacs come editor

Per prima cosa dobbiamo decidere cosa vogliamo fare... Come abbiamo detto Emacs è anche un editor.

Iniziamo ad esplorarne le funzioni.

Ad oggi gran parte delle cose può essere fatta tramite una comoda interfaccia grafica, molto intuitiva. Emacs ci presenta una serie di menù e una serie di icone associate ognuna ad un'azione.



La vera potenza di Emacs però è rappresentata dalle scorciatoie da tastiera. Ogni azione può essere richiamata da una scorciatoia oppure da un comando. Emacs ci fornisce:

- la possibilità di usare scorciatoie predefinite;
- la possibilità di personalizzare scorciatoie esistenti;
- la possibilità di associare comandi a nuove scorciatoie.

Tutto questo semplicemente agendo sul file di configurazione `~/.emacs`. Per fare questo rimando al manuale e in questa sede ci occuperemo soltanto delle scorciatoie predefinite. Le potenzialità di emacs sono infinite pertanto vedremo soltanto pochi esempi.

Innanzitutto vediamo come si inseriscono comandi e scorciatoie.

Gran parte delle scorciatoie comincia con la combinazione di tasti **Ctrl+C** che nel linguaggio di Emacs si rappresenta con **C-x**. Viceversa ogni comando viene immesso previo inserimento della combinazione **Alt+x** indicato con **M-x**. Quando inseriamo questa combinazione il cursore si sposta nella echo area e ci è permesso di immettere il comando.

The screenshot shows the Emacs editor interface. At the top, the mode line indicates the file is named 'medie.f', the buffer is 'All L1', and the major mode is '(Fortran)'. Below the mode line, a completion menu is displayed, showing 'M-x find-file' with a cursor at the end of the text.

Emacs fornisce l'autocompletamento dei comandi analogamente a bash. Ad esempio è sufficiente la combinazione *M-x* seguita dal TAB per avere accesso al frame **Completions** dove sono riportati tutti i comandi che possiamo inserire.

Aprire un file di testo

Detto *visit new file*. La scorciatoia è *C-x C-f* (questo significa semplicemente che dobbiamo premere **Ctrl+x** quindi **Ctrl+f**). Ovviamente possiamo utilizzare il comando esteso che è *M-x find-file* (ovvero si preme la combinazione **Alt+x** quindi si inserisce il comando *find-file*).

Nella echo area apparirà la dicitura

```
Find file $\sim$/
```

...e non dovremo fare altro che inserire il percorso del file da aprire (aiutati dall'autocompletamento) o del nuovo file da creare.

Per semplificarci la vita possiamo usare la modalità file manager di emacs che richiamiamo con il comando *M-x dired* o la scorciatoia *C-x d*: in un frame verrà mostrato il filesystem a partire dalla directory di lavoro, navigabile in modo analogo a midnight commander.

Uscire da Emacs

La cosa più importante quando si avvia un programma è sapere come uscirne! Quindi quando abbiamo fatto le nostre cose, usciamo con *C-x C-c* oppure con il relativo comando *M-x save-buffers-kill-emacs*.

Salvare un file su disco

Come al solito, abbiamo la possibilità di dare:

- salva - *C-x C-s* oppure *M-x savebuffer*
- salva con nome - *C-x C-w* oppure *M-x write-file*

In entrambi i casi la echo area ci informa di cosa stiamo facendo; solo nel secondo caso ci verrà chiesto di specificare un nuovo nome per il file che andremo a creare.

Ricerca e sostituzioni nel testo

Emacs ci dà la possibilità effettuare ricerche e sostituzioni.

Le ricerche possono essere fatte sia avanti che a ritroso nel testo. Sia nel caso della ricerca che della sostituzione, possiamo utilizzare la modalità classica oppure impiegare le versatissime regular expressions (regexp);

- cerca avanti: *C-x s* oppure *M-x search-forward*
- cerca avanti tramite regexp: *C-u C-s* oppure *M-x search-forward-regexp*
- cerca indietro: *C-x r* oppure *M-x search-backward*
- cerca indietro tramite regexp: *M-x search-backward-regexp*
- sostituisci: *M-%* oppure *M-x query-replace*
- sostituisci tramite regexp: *M-C-%* oppure *M-x query-replace-regexp*

Comparazione di testi

Emacs ci permette di comparare due versioni di un file (o buffer) per evidenziarne le differenze. È sufficiente impartire il comando *M-x ediff* e seguire la procedura guidata nella echo area per comparare due file. Per cose molto più sopraffini rimando al manuale di ediff ⁵.

Ambienti di editing (major mode)

Una delle utilissime funzionalità che emacs offre ai programmatori (ma anche a chi semplicemente usa \LaTeX) è la possibilità di caricare l'ambiente di programmazione adatto al linguaggio che stiamo utilizzando. Ad esempio, se usiamo il buon vecchio FORTRAN, quando apriamo il sorgente .f del nostro progetto, Emacs riconosce automaticamente l'ambiente da caricare e ci offre il suo aiuto. Questo può andare dalla semplice evidenziazione della sintassi fino ad un vero e proprio autocompletamento dei comandi o alla possibilità di inserimento di testo in stile MACRO.

La mode line indicherà il modo che stiamo utilizzando e una nuova voce nel menù verrà visualizzata con nuove interessanti opzioni e utilità specifiche per quel linguaggio di programmazione.

⁵http://www.gnu.org/software/emacs/manual/html_mono/ediff.html

Se emacs dovesse fallire possiamo settare il *major mode* a mano tramite un comando, come ad esempio:

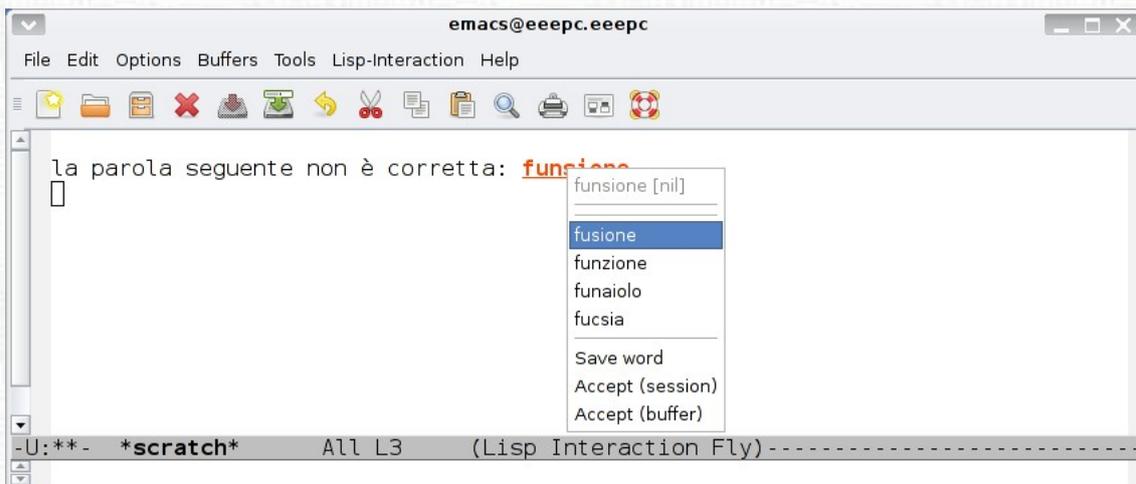
- *M-x c-mode* carica l'ambiente di programmazione C;
- *M-x fortran-mode* carica l'ambiente di programmazione FORTRAN;

Correttore ortografico

Ovviamente questa funzione non può mancare in un editor che si rispetti. Emacs offre la possibilità di utilizzare gli *spell-checker* **ispell** e **aspell**. *GNU aspell* è la scelta predefinita essendo il software pensato da GNU per sostituire il buon vecchio *ispell*. Oltretutto è noto che *aspell* è più veloce nonché presenta vocabolari (anche in italiano) molto più sostanziosi ed aggiornati. Come prima cosa quindi installiamo *aspell* e il relativo vocabolario italiano:

```
aptitude install aspell aspell-it
```

Possiamo utilizzare la funzionalità di *flyspell* che effettua lo *spell-check* durante la digitazione del testo (similmente al correttore ortografico di openoffice) evidenziando le parole che non corrispondono con il vocabolario. Per effettuare la correzione è sufficiente fare click con il tasto centrale del mouse sulla parola evidenziata e scegliere la parola corretta tra i suggerimenti.



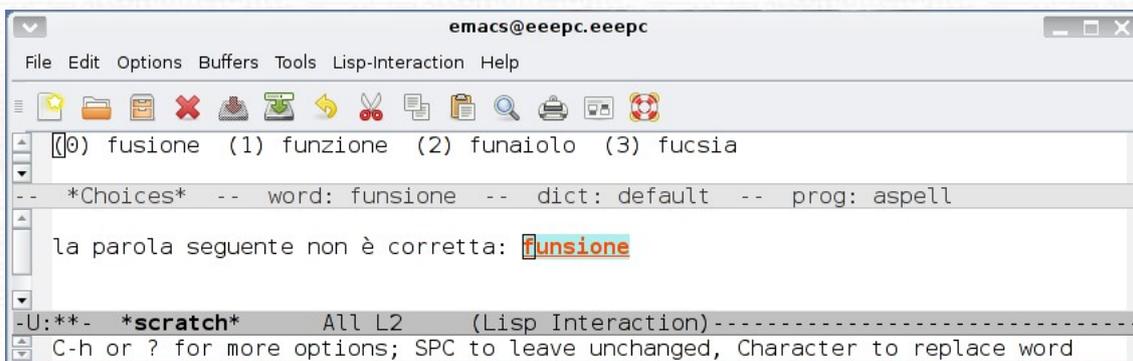
Flyspell può essere attivato dal menu:

Tools --> Spell Checking --> Automatic spell checking (flyspell)

Dallo stesso menù si possono configurare i dizionari. Il comando di avvio rapido è *M-x flyspell-mode*.

Per richiamare una sessione di aspell possiamo invece utilizzare il comando *M-x ispell* che effettuerà una correzione nell'intero buffer oppure nella sola eventuale regione di testo selezionata.

Per ogni errore riscontrato aspell ci proporrà dei suggerimenti: la sostituzione può essere fatta premendo il tasto corrispondente:



5.1.5 Altre utilità di Emacs

Come abbiamo detto Emacs ha tantissime funzionalità, oltre a quella di editor.

Facciamo una carrellata su quelle più interessanti.

Funzionalità di manuale

Come dice il manuale stesso, Emacs è *auto-documentante*.

Cerchiamo di capire come chiedere ad emacs le informazioni che ci servono. Per questo usiamo la funzione data dalla scorciatoia *C-h* facendo seguire **k** se vogliamo informazioni su una scorciatoia oppure **f** per un comando.

Ad esempio, cosa comporta la combinazione di tasti *M-%* ? Possiamo ottenere risposta mediante: *C-h k M-%*:

M-% runs the command `query-replace`, which is an interactive compiled Lisp function.

It is bound to M-%, <menu-bar> <edit> <replace> <query-replace>.

```
(query-replace from-string to-string &optional delimited start end)
```

Replace some occurrences of `from-string` with `to-string`.

As each match is found,

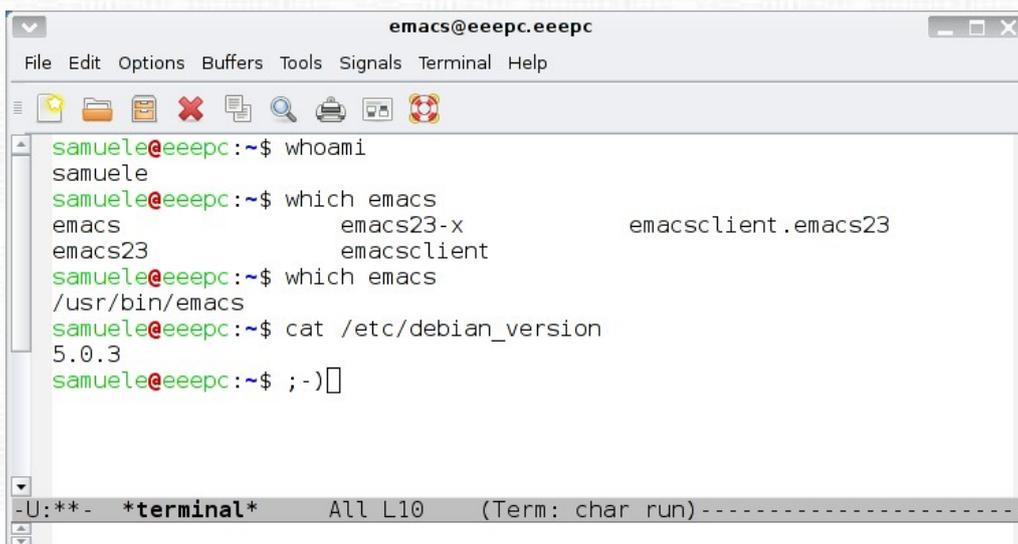
....

....

Analogamente volendo informazioni sul comando associato alla scorciatoia precedente: *C-h f query-replace-regexp*.

Funzionalità di shell

Mediante il comando *M-x shell-command* possiamo impartire un qualsiasi comando da shell, mentre tramite *M-x shell* apriamo una vera e propria shell nel frame inferiore di Emacs. Tramite *M-x term* invece, trasformiamo Emacs in un vero e proprio emulatore di terminale.



The screenshot shows the Emacs terminal window titled "emacs@eeepc.eeepc". The terminal displays a shell session with the following commands and output:

```

samuele@eeepc:~$ whoami
samuele
samuele@eeepc:~$ which emacs
emacs                emacs23-x           emacsclient.emacs23
emacs23              emacsclient
samuele@eeepc:~$ which emacs
/usr/bin/emacs
samuele@eeepc:~$ cat /etc/debian_version
5.0.3
samuele@eeepc:~$ ;-)
```

The terminal window also shows the Emacs menu bar (File, Edit, Options, Buffers, Tools, Signals, Terminal, Help) and a toolbar with various icons. At the bottom, the terminal status bar shows "-U:**- *terminal* All L10 (Term: char run)-----".

Funzionalità di visualizzatore di documenti

Emacs può visualizzare diversi tipi di file: pdf, eps, postscript, dvi, nonché diversi formati di file immagine. Sono tutti visualizzabili grazie alla funzionalità data da *doc-view*. Alcuni di essi, come i postscript (files contenenti istruzioni descrittive), vengono aperti normalmente in modalità testo: per passare alla modalità *documento* è sufficiente *M-x doc-view-mode*. Tutte le funzioni di un visualizzatore di documenti sono assicurate.

Funzionalità di news-reader

Emacs sfrutta gnus per divenire un perfetto news reader. Possiamo specificare un server dal quale vogliamo leggere le notizie, inserendolo in *~/newsrc*. Il comando per avviare gnus è *M-x gnus*.

Funzionalità di vi

Infine, una chicca per chi viene dal mondo di vi(m) e vuole iniziare ad apprezzare Emacs: **viper-mode**.

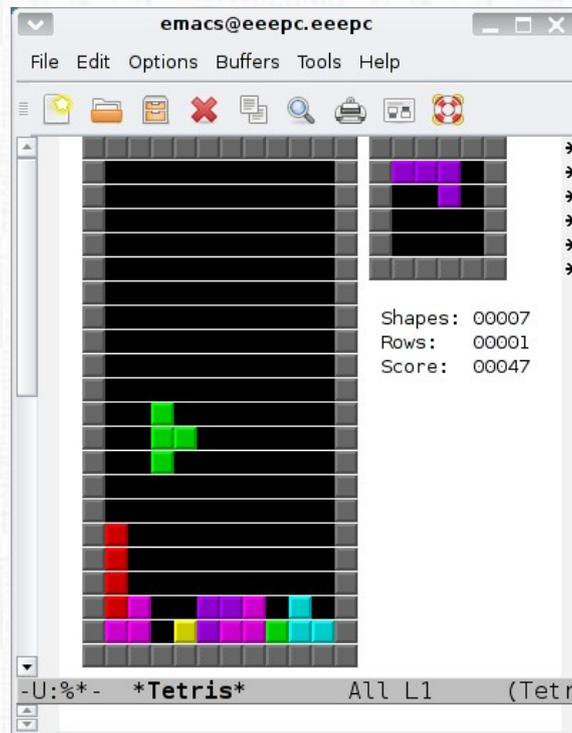
Questo pacchetto offre una piena e completa emulazione dell'editor vi, dentro Emacs. Si avvia semplicemente mediante il comando *M-x viper-mode*:

```
"Viper Is a Package for Emacs Rebels"
```

Anche i membri dell'altra fazione della *guerra degli editor* possono utilizzare Emacs ;-)

Giochi

È noto che ogni buon editor di testo che si rispetti contiene dei giochi al suo interno ;-). Emacs ci dà la possibilità di giocare a tetris con *M-x tetris*, a pong con *M-x pong* ed altri tutti da provare come *M-x blackbox*, *M-x mpuz* and *M-x 5x5*. Da non dimenticare l'imbattibile gioco di scacchi fornito dal pacchetto Lisp **chess.el**.



5.1.6 Quante altre cose è Emacs?!

Leggendo il manuale, possiamo scoprire quante cose è Emacs!

Seguono alcuni esempi, che non verranno approfonditi per motivi di spazio... Ogni funzione è comunque sempre ben documentata e ci si potrà stupire di come alcune di esse siano davvero raffinate!

In conclusione, Emacs è anche:

- Un calendario avanzato: *M-x calendar*
- Un calendario maya, che ci ricorda la fine del mondo ⁶
- Un mail-client completo: invio *M-x mail*, ricezione *M-x rmail*
- Un browser (grazie al pacchetto Lisp opzionale *w3*): *M-x w3-mode (obsoleto)*
- Un sistema di sorting per stringhe di testo: *M-x sort-**

⁶http://it.wikipedia.org/wiki/Calendario_maya#Calcolo_del_calendario_maya

- Un editor per file binari: *M-x hexl-find-file*
- Uno stupido ammazzatempo: *M-x hanoi-unix*
- Un imbattibile giocatore di gomunku: *M-x gomonku*
- Un solitario cinese: *M-x solitaire*
- Un convertitore/traduttore in codice morse: *M-x morse*, *M-x unmorse*
- Un gran confusionario... da far girare la testa! *M-x zone*
- Un gioco d'avventura molto cervelotico...: *M-x dunnet*
- Una forma di vita: *M-x life*
- *Una religione.*

Tutto questo è Emacs...

Link utili

Manuale di Emacs in diversi formati

<http://www.gnu.org/software/emacs/manual/emacs.html>

Una lista dei comandi e scorciatoie più utili

http://sean.wenzel.net/docs/emacs/quick_reference/

GNU Emacs su wikipedia

<http://en.wikipedia.org/wiki/Emacs>

Emacs per multics

<http://www.multicians.org/mepap.html>

TECO

http://en.wikipedia.org/wiki/Text_Editor_and_Corrector

<http://linuxfinances.info/info/teco.html>

Redattore: borlongioffi

5.2 ALSA & Gnome

5.2.1 Un po' di storia

Il supporto per l'audio in Linux ha avuto origine all'inizio degli anni '90 con il driver Creative SoundBlaster, da cui è nata la fondazione per l'Open Sound System (*OSS*). Poiché con il tempo è cresciuta l'insoddisfazione per tale sistema, a partire dal 1998 Jaroslav Kysela ha dato origine, insieme ad altri sviluppatori, all'*Advanced Linux Sound System* (*ALSA*).

Lo sviluppo di tale progetto non è stato lineare e, tra il 1999 ed il 2001, *ALSA* è stato ridisegnato più volte, richiedendo la modifica delle applicazioni audio perché non più compatibili. In questo periodo sono stati, comunque, aggiunti il *sequencer ALSA* ed un router MIDI in kernel-space.

Dalla fine del 2001 *ALSA* è diventato il sistema audio ufficiale di Linux al posto di *OSS*, anche se quest'ultimo non è scomparso del tutto ed è ancora in uso sia su alcuni sistemi Linux che nei sistemi Unix.

Per superare alcuni limiti di *ALSA*, tra i quali la possibilità di connettere tra loro applicazioni audio, Paul Davis nel 2000 ha iniziato a lavorare sulla workstation di audio digitale *Ardour* e nel 2002 ha sviluppato *JACK*, server audio idoneo più all'uso professionale che all'uso su sistemi desktop.

Nel 2004 Lennart Poettering ha iniziato lo sviluppo di *PulseAudio* (introdotto nei sistemi desktop per la prima volta in Fedora 8), un server audio che permette operazioni avanzate quale l'acquisizione dell'audio da più sorgenti e il suo inoltro ad applicazioni o componenti hardware differenti.

5.2.2 Le API audio in linux

È possibile individuare differenti API audio che soddisfano differenti necessità:

ALSA: è l'API di Linux per la riproduzione e la registrazione. Anche se la versione estesa di questa API può apparire ampia e complessa, garantisce il supporto a quasi tutti i moderni componenti hardware audio. Alcune delle funzionalità delle API *ALSA* sono limitate nel loro utilizzo ai dispositivi hardware supportati dal kernel Linux. È consigliato l'uso della versione estesa quando si desidera fare uso di caratteristiche hardware molto specifiche che potrebbero essere necessarie ai fini della produzione audio, altrimenti è sufficiente l'utilizzo della versione ridotta, un'astrazione adatta per la riproduzione e la registrazione di base.

Gstreamer: è lo standard *de-facto* tra i riproduttori per i desktop Linux. Permette la codifica e la decodifica di flussi audio e video grazie al supporto di una vasta gamma di codec audio e backend tra cui *ALSA*, *OSS* e *PulseAudio*. Non è particolarmente adatto ad applicazioni a bassa latenza o realtime.

JACK: è un sistema audio per collegare applicazioni professionali di produzione audio con dispositivi hardware. È indicato principalmente per sistemi a bassa latenza e dove è richiesta l'interconnessione tra applicazioni. Non è idoneo per l'uso desktop o in sistemi embedded.

Libcanberra: è una semplice interfaccia astratta che permette di eseguire i suoni legati agli eventi di sistema. Sui sistemi Unix/Linux il suono da emettere in relazione ad un evento è individuato mediante un meccanismo definito dalle specifiche *XDG Sound Theming/Naming*. *Libcanberra* si basa su tali specifiche. *Gnome* dipende da tale API ma *libcanberra* non ha alcuna dipendenza da *Gnome/Gtk/Glib*, per cui può essere usato anche su altri ambienti desktop.

OSS: è un API di basso livello supportato da diversi sistemi *nix compreso Linux. È stato il primo standard audio in Linux, e nei nuovi kernel è supportato nella versione API 3 come *OSS3*. *OSS3* è considerato obsoleto ed è stato completamente sostituito da *ALSA*. Il successore di *OSS3*, chiamato *OSS4*, è disponibile ma non ha praticamente nessun ruolo su Linux e non è supportato nel kernel standard o da una alcuna delle distribuzioni maggiori.

Phonon e *KNotify*: *Phonon* è un'astrazione di alto livello per sistemi di streaming, mentre *Knotify* è un sistema per le notifiche che però non supporta le specifiche *XDG Sound Theming/Naming*. Entrambe le API sono specifiche per *KDE/Qt*, per cui non possono essere usate al di fuori di applicazioni *KDE/Qt*.

PulseAudio: è un sistema audio per i desktop Linux e gli ambienti embedded che gira in user-space e, di solito, al di sopra di *ALSA*. *PulseAudio* ha una API nativa ma supporta anche la versione ridotta di *ALSA*. Non è utile in ambienti di produzione audio professionale, anche se fornisce la connettività a *JACK*.

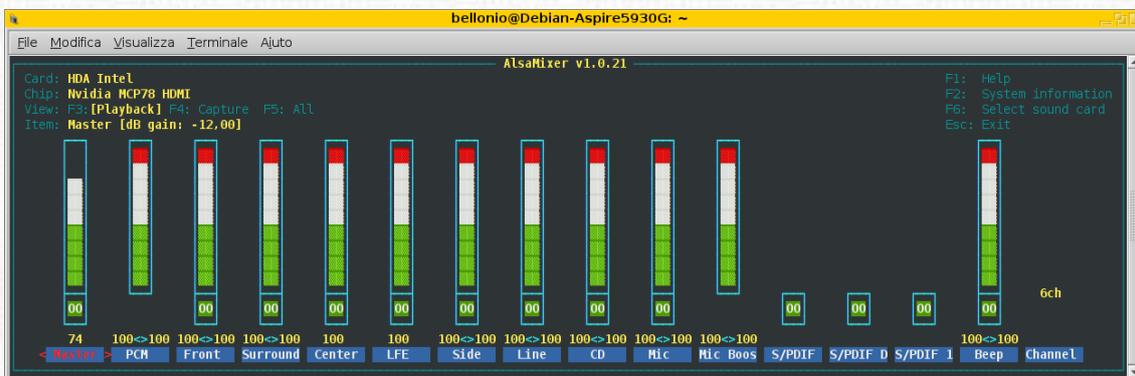
5.2.3 Installare e configurare ALSA

ALSA è parte integrante del kernel 2.6 e per eseguire l'installazione è sufficiente, da root, eseguire:

```
# apt-get install alsa-base alsa-utils
```

per eseguire la regolazione dei volumi è possibile eseguire

```
$ alsamixer
```



il mixer riporta, nella parte alta, i dati della scheda audio e del chip nonché il volume dei singoli canali audio. Alla base di ciascun canale è riportato “00” se il canale è attivo, “MM” se è inattivo (mute-mode); per attivarlo è sufficiente spostarsi con i tasti freccia (sinistra e destra) sul canale e digitare “m”, mentre con i tasti freccia su – freccia giù è possibile aumentare o abbassare il volume.

Terminate le regolazioni è possibile memorizzarle mediante il comando

```
$ alsactl store
```

5.2.4 Abilitare i suoni in Gnome

Per poter abilitare i suoni di Gnome è necessario installare ulteriori pacchetti:

```
# apt-get install gnome-audio libesd-alsa0 esound
```

In Debian Lenny, che ha la versione 2.22.2 5 di Gnome, i suoni di sistema sono già abilitati, mentre in Squeeze, che ha la versione 2.28.1-1 di Gnome, è necessario scaricare ed installare il pacchetto *freedesktop-sound-theme.git* dal sito <http://git.debian.org>. Tale pacchetto inserisce la cartella dei suoni freedesktop in */usr/share/sounds* utilizzata dalle applicazioni, Gnome incluso.

Scaricato il pacchetto in formato tar.gz si scompatta mediante il comando

```
$ tar xvzf freedesktop-sound-theme.tar.gz
```

verrà creata la directory "freedesktop-sound-theme".

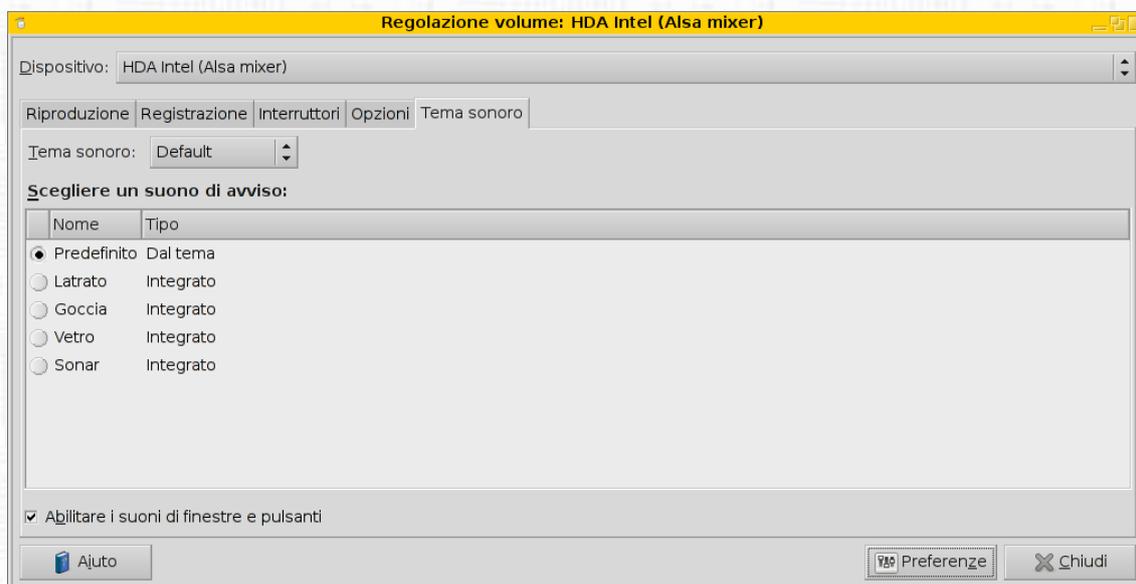
Con la sequenza dei comandi

```
$ cd freedesktop-sound-theme/  
$ ./configure  
$ make
```

si provvede a configurare e a compilare il pacchetto e si può procedere, da root, all'installazione mediante il classico comando

```
# make install
```

A questo punto i suoni di Gnome sono abilitati e accedendo a Sistema → Preferenze → Audio è possibile abilitare il tema sonoro.



5.2.5 Riferimenti

<http://it.emcelettronica.com/audio-linux-passato-presente-e-futuro-discussi-all-lpc-1%C2%B0-parte>

<http://it.emcelettronica.com/audio-linux-passato-presente-e-futuro-discussi-all-lpc-2%C2%B0-parte>

<http://it.emcelettronica.com/audio-linux-passato-presente-e-futuro-discussi-all-lpc-3%C2%B0-parte>

<http://lwn.net/Articles/355542/>

<http://lwn.net/Articles/299211/>

<http://linuxplumbersconf.org/2009/program/>

<http://0pointer.de/blog/projects/guide-to-sound-apis>

<http://forum.debianizzati.org/multimedia/debian-squeeze-e-suoni-sistema-t36489.15.html>

<http://git.debian.org/>

http://www.debianclan.org/index.php?option=com_content&task=view&id=27&Itemid=3

Redattore: bel.gio

5.3 Alias

5.3.1 Usare gli alias

In un sistema GNU/Linux è possibile personalizzare la shell impostando degli alias, cioè delle scorciatoie per i comandi utilizzati frequentemente. Ad esempio, digitando il comando.

```
ls -a --color=auto
```

in un terminale otteniamo l'elenco delle cartelle e dei file (`ls`), compresi i file nascosti (`-a`), il tutto reso mediante una colorazione che automaticamente differenzia gli elementi (`-color=auto`). Con la definizione di un alias per il comando `ls` si può fare in modo che venga eseguito completo dei parametri che ci occorrono senza bisogno di specificarli.

Analogamente è possibile associare *film* come alias al suo percorso:

```
cd ~/archivio/film
```

oppure

```
cd /home/nome_utente/archivio/film
```

In tal modo la semplice digitazione dell'alias 'film' posiziona il terminale nella directory specificata.

La creazione di alias consente di impostare il proprio terminale per l'esecuzione di comandi complessi richiamabili in maniera semplice e veloce, rendendoli molto utili nella definizione di procedure personalizzate.

L'installazione base di un sistema Debian prevede la presenza del pacchetto `bash`, come interprete dei comandi, e del pacchetto `bash-completion` che estende la capacità di completamento dell'interprete su righe di comando complesse. Il pacchetto `bash` in particolare, oltre ad installare in `/etc` il file `bash.bashrc` che provvede alla configurazione del sistema, installa anche la directory `/etc/skel` al cui interno si trovano i file nascosti `.bash_logout`, `.bashrc` e `.profile` che servono, insieme ad altri, al comando `useradd` per creare la configurazione di default nella home directory dei nuovi utenti. Ognuno di questi file esprime una configurazione di base e può essere personalizzato da parte dell'utente così come, se non è presente, può essere facilmente ricreato mediante un editor di testo o ancor più semplicemente ricopiandolo dall'originale mediante il comando:

```
cp /etc/skel/.bashrc /home/nome_utente/.bashrc
```

ed eseguendo successivamente il comando:

```
$ source .bashrc
```

per rendere immediatamente disponibili le impostazioni. Lo stesso comando va impartito ogni qual volta, modificate le impostazioni, le si voglia rendere effettive.

Il file `.bashrc` del pacchetto `bash_4.0-7` in Debian Sid è il seguente:

```
01 # ~/.bashrc: executed by bash(1) for non-login shells.
02 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
03 # for examples
04
05 # If not running interactively, don't do anything
06 [ -z "$PS1" ] && return
07
08 # don't put duplicate lines in the history. See bash(1) for more options
09 # don't overwrite GNU Midnight Commander's setting of 'ignorespace'.
10 HISTCONTROL=$HISTCONTROL${HISTCONTROL+,}ignoredups
11 # ... or force ignoredups and ignorespace
12 HISTCONTROL=ignoreboth
13
14 # append to the history file, don't overwrite it
15 shopt -s histappend
16
17 # for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
18
19 # check the window size after each command and, if necessary,
20 # update the values of LINES and COLUMNS.
21 shopt -s checkwinsize
22
23 # make less more friendly for non-text input files, see lesspipe(1)
24 #[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
25
26 # set variable identifying the chroot you work in (used in the prompt below)
27 if [ -z "$debian_chroot" ] && [ -r /etc/debian_chroot ]; then
28 debian_chroot=$(cat /etc/debian_chroot)
29 fi
```

```
30
31 # set a fancy prompt (non-color, unless we know we "want" color)
32 case "$TERM" in
33 xterm-color) color_prompt=yes;;
34 esac
35
36 # uncomment for a colored prompt, if the terminal has the capability; turned
37 # off by default to not distract the user: the focus in a terminal window
38 # should be on the output of commands, not on the prompt
39 #force_color_prompt=yes
40
41 if [ -n "$force_color_prompt" ]; then
42 if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
43 # We have color support; assume it's compliant with Ecma-48
44 # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
45 # a case would tend to support setf rather than setaf.)
46 color_prompt=yes
47 else
48 color_prompt=
49 fi
50 fi
51
52 if [ "$color_prompt" = yes ]; then
53 PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033
[01;34m\]\w\[\033[00m\]\$ '
54 else
55 PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
56 fi
57 unset color_prompt force_color_prompt
58
59 # If this is an xterm set the title to user@host:dir
60 case "$TERM" in
61 xterm*|rxvt*)
62 PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
63 ;;
64 *)
65 ;;
66 esac
```

```
67
68 # Alias definitions.
69 # You may want to put all your additions into a separate file like
70 # ~/.bash_aliases, instead of adding them here directly.
71 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
72
73 #if [ -f ~/.bash_aliases ]; then
74 #. ~/.bash_aliases
75 #fi
76
77 # enable color support of ls and also add handy aliases
78 if [ -x /usr/bin/dircolors ]; then
79 test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval \
    "$(dircolors -b)"
80 alias ls='ls --color=auto'
81 #alias dir='dir --color=auto'
82 #alias vdir='vdir --color=auto'
83
84 #alias grep='grep --color=auto'
85 #alias fgrep='fgrep --color=auto'
86 #alias egrep='egrep --color=auto'
87 fi
88
89 # some more ls aliases
90 #alias ll='ls -l'
91 #alias la='ls -A'
92 #alias l='ls -CF'
93
94 # enable programmable completion features (you don't need to enable
95 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
96 # sources /etc/bash.bashrc).
97 if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
98 . /etc/bash_completion
99 fi
```

Il file ".bashrc" dalla riga 68 introduce l'utente alla creazione dei propri alias mediante la creazione di un file di testo personale chiamato ".bash_aliases" anch'esso facilmente ricavabile mediante un editor di testo o, ancor più semplicemente, utilizzando il comando:

```
touch .bash_aliases
```

assumendo il fatto che si sta lavorando all'interno della propria directory home. Da notare che la creazione di `.bash_aliases` per l'impostazione degli alias non è tecnicamente necessaria in quanto la definizione degli alias può avvenire direttamente all'interno dello stesso `.bashrc` come si nota poco più in basso alla riga 80 con la creazione di un alias per la colorazione dell'output del comando `ls`. Tuttavia, sebbene non necessaria, la creazione di questo secondo file è consigliabile per una gestione ordinata dei nostri alias che, è bene ricordarlo, pur non trattandosi in generale di procedure che possono nuocere alla stabilità del sistema, sono pur sempre definizioni di comandi che possono intervenire anche sui nostri file ed è dunque consigliata la massima attenzione nella loro composizione. Va inoltre precisato che è possibile impostare alias su comandi che nella loro esecuzione prevedono diversi privilegi utente, compresi i privilegi di root, e che anche l'utente root può avere il suo `.bash_aliases`.

Per utilizzare il file `.bash_aliases` occorre togliere il commento `#` alle righe 73, 74 e 75 di `.bashrc`; si è così pronti ad inserire finalmente i nostri alias.

Passando in rassegna i possibili utilizzi degli alias cerchiamo di specificarne la struttura partendo, ad esempio, dai più semplici visti precedentemente ed inserendoli nel file `.bash_aliases`.

```
alias film='cd ~/archivio/film'
```

```
alias ls='ls -a --color=auto'
```

In ognuna delle righe basta inserire il termine `alias` seguito da uno spazio e dal nome che si desidera assegnare all'alias, quindi il simbolo `=` ed il comando inserito tra apici. Rispettando questa struttura molto semplice è già possibile inserire una grande varietà di alias molto comodi per navigare nel sistema

```
alias cd..="cd .."
alias ..="cd .."
alias ...="cd ../../"
alias ....="cd ../../.."
alias p='cd -'
alias back='cd $OLDPWD'
alias src='cd /usr/src' # change to source dir
alias log='cd /var/log' # change to log dir
```

```
alias down='cd ~/downloads'  
alias film='cd ~/downloads/films'  
alias foto='cd ~/documenti/immagini'  
alias donk='cd ~/.amule/incoming/files'  
alias torr='cd ~/deluge/incoming'
```

o anche alias per ottenere output specifici sul comando ls

```
alias d='ls'  
alias ll='ls -l'  
alias la='ls -A'  
alias l='ls -CF'  
alias ls="ls -s -F -T 0 -b -H -1 -C --color=auto"
```

o ancora alias per la gestione file che possono prevenire alcuni problemi

```
alias cp="cp -i" # chiedi conferma prima di sovrascrivere  
alias mv="mv -i" # chiedi conferma prima di sovrascrivere  
alias rm="rm -i" # chiedi conferma prima di eliminare
```

Continuando con gli esempi è possibile prevedere una serie di alias che ci consentono di acquisire immediatamente informazioni sullo stato del sistema

```
#spazio dischi con tipo di filesystem, conversione byte in KiB/MiB/GiB  
#ed esclusione dei filesystem virtuali (tmpfs, udev)  
alias df='df -Thx tmpfs'
```

```
#ram usata con totale e conversione byte in KiB/MiB/GiB  
alias free='free -tm'
```

```
#albero dei processi con pid  
alias pst='pstree -p|less'
```

```
#veloce consultazione di errori  
alias xs='tail $HOME/.xsession-errors'
```

```
#spazio utilizzato da file o directory  
#conversione byte in KiB/MiB/GiB, somma singolo elemento e totale di tutti  
alias dut='du -hsc'
```

```
#spazio utilizzato da file o directory
#conversione byte in KiB/MiB/GiB, un solo livello di sottodirectory
alias dud='du -h --max-depth=1'
```

```
#i 5 programmi piu' usati nella sessione corrente
alias mostused='hash|grep -v hits|sort -nr|head -n5'
```

```
#elenca gli 11 file modificati piu' di recente
alias lastfile='ls -alt|head -11'
```

```
#notifica i cambiamenti apportati
alias chown='chown -c'
alias chmod='chmod -c'
```

```
#controllo connessioni
alias netm='netstat -putan'
```

Un `.bash_aliases` ben strutturato consente di gestire un sistema in maniera completa prevedendo, ad esempio, la gestione dei supporti in accordo con `fstab`, l'avvio e spegnimento del PC, la gestione del software o di alcuni tipi di connessione in accordo con i privilegi dell'utente

```
alias upenin='mount /media/usbpen;cd /media/usbpen;ls'
alias upenout='cd;umount /media/usbpen'
alias esterin='mount /media/esterno;cd /media/esterno;ls'
alias esterout='cd;umount /media/esterno'
alias datiin='mount /media/dati;cd /media/dati;ls'
alias datiout='cd;umount /media/dati'
alias camin='mount /media/digicam;cd /media/digicam;ls'
alias camout='cd;umount /media/digicam'
alias cdin='mount /media/cdrom;cd /media/cdrom;ls'
alias cdout='cd;umount /media/cdrom;eject;sleep 11;eject -t'
```

```
#spegne il pc
alias halt='su -c halt'
```

```
#riavvia il pc
```

```
alias reboot='su -c reboot'
```

```
#riavvia il pc con controllo forzato del disco (fsck)
```

```
alias rebfsck='su -c "shutdown -rF now"'
```

```
#connessione remota ad altro pc su porta diversa dal default
```

```
#la porta necessita di essere impostata nella configurazione di sshd
```

```
alias remss='ssh -p 54321 -X remote@192.168.0.14'
```

```
#idem per scp
```

```
alias scp='scp -P 54321'
```

Analogamente è possibile utilizzare gli alias per automatizzare il comportamento di comandi che richiederebbero l'intervento dell'utente come nel caso del comando ping e del comando pwgen

```
#ferma il ping dopo 5 invii e ricezioni
```

```
alias ping='ping -c 5'
```

```
#genera password di 11 caratteri, senza numeri
```

```
alias pwgen='pwgen 11 -0 -N 1'
```

così come di applicazioni da terminale come wget o aria2c

```
#wget con directory specifica di destinazione
```

```
alias wget='wget -P /media/dwnld/'
```

```
#aria2c con directory di destinazione, file di input e numero
```

```
#di file scaricati contemporaneamente
```

```
#basta specificare gli url di input nel file list uno per riga
```

```
alias aria='aria2c -d ~/downloads -i ~/downloads/list -l ~/downloads/log -j 3'
```

E' anche possibile predisporre gli alias ad accettare input (\$1) da parte dell'utente mediante l'utilizzo di variabili come ad esempio una ricerca da effettuare in questo caso all'interno del file .bash.history che contiene un elenco di comandi utilizzati nel terminale

```
#cerca qualcosa nella history bash
```

```
alias his='history|grep $1'
```

o anche in molti altri casi

```
#cerca chiave gpg
gpg --list-keys|grep -B 1 $1
```

```
#cerca nei pacchetti installati
#es: pkg mplayer
alias pkg='dpkg -l|grep $1'
```

```
#varie utility per cercare o controllare nei repository
#es: cache mplayer
alias cache='apt-cache search $1'
alias show='aptitude show $1'
alias policy='apt-cache policy $1'
alias dep='apt-cache depends $1'
alias rdep='apt-cache rdepends $1'
```

```
#why e' un'opzione di aptitude che spiega il motivo per cui un
#pacchetto e' installato, guardando le dipendenze piu' prossime
alias why='aptitude why $1'
```

Continuando sui possibili utilizzi degli alias si nota come questi possano via via diventare più completi ed efficaci in diverse situazioni di utilizzo con l'aumentare delle conoscenze del linguaggio bash ed un conseguente aumento dell'utilizzo di variabili. Un caso concreto può essere ad esempio rappresentato dal comando `rm` sul quale abbiamo precedentemente illustrato un alias per chiedere conferma della rimozione di un file

```
alias rm="rm -i" # chiedi conferma prima di eliminare
```

è possibile infatti replicare il comportamento di un ambiente desktop semplicemente pre-disponendo una directory che funge da cestino ed un alias per il comando `rm` che diventa un semplice `mv`, cioè lo spostamento del file nella target directory,

```
#cestino, anziche' cancellare sposta in una directory
alias rm='mv "$@" --target-directory=$HOME/cestino'
```

allo stesso modo è possibile impostare lo svuotamento del cestino magari aggiungendo anche lo svuotamento di altre directory

```
#svuota il cestino piu' la cache di anteprima delle immagini
alias emptrash='/bin/rm -f $HOME/.thumbnails/normal*/;/bin/rm -rf $HOME/cestino/*'
```

Si può far rientrare nella categoria degli alias la costruzione di vere e proprie funzioni che si possono inserire nel nostro `.bash_aliases` e di cui possiamo vedere di seguito una piccola carrellata con tanto di spiegazione della funzione svolta

```
#estrae un pacchetto .deb ricreando
#la directory col nome dello stesso
jexdeb() {
dpkg -x "$@" $PWD/$(echo $@|sed 's/.deb//')
}

#crea il file LISTA-$DIR con un listato di tutti
#i file contenuti nel path in cui lo si usa
jmkls() {
ls -aR > $PWD/LISTA-$(basename $PWD)
}

#gestione demoni, un po' scomodo se non si ricorda il nome
#quindi senza argomenti si ottiene un semplice listato della directory
### daemons ###
dstart() {
    if [ "$1" = "" ]; then
        echo -e "\033[32mstart a daemon, choose one from the list below"
        echo -e "\033[32musage: dstart <daemon>\e[0m"
        /bin/ls -A /etc/init.d/
    else
su -c "/etc/init.d/$1 start"
    fi
}

dstop() {
    if [ "$1" = "" ]; then
        echo -e "\033[32mstop a daemon, choose one from the list below"
        echo -e "\033[32musage: dstop <daemon>\e[0m"
        /bin/ls -A /etc/init.d/
    else
su -c "/etc/init.d/$1 stop"
    fi
}
```

```
drest(){
  if [ "$1" = "" ]; then
    echo -e "\033[32mrestart a daemon, choose one from the list below"
    echo -e "\033[32musage: drest <daemon>\e[0m"
    /bin/ls -A /etc/init.d/
  else
    su -c "/etc/init.d/$1 restart"
  fi
}

#cerca processo attivo con ps
#es: psx mplayer
### psgrep ###
psx() {
ps aux|grep -v grep|egrep "USER|$1"
}

#cerca processo attivo con top
#es: topg mplayer
### topgrep ###
topg() {
top -b -n 1|egrep "($1)|PID"
}

#un surrogato di sudo
#es: sux vim /etc/issue
### su -c wrapper ###
sux() {
su -c "$*"
}

#mostra valori di ventole e temperature attraverso sensors (lm-sensors)
### sensors ###
sens() {
echo $(echo CPU::sensors|tail|grep temp1|awk '{print $2}';echo '/';sensors|grep fan1 \
|awk '{print $2,$3}')
echo $(echo MB::sensors|grep temp2|awk '{print $2}';echo '/';sensors|grep fan3 \
```

```
|awk '{print $2,$3}')
```

```
}

#calcolatrice
#es: calc 3*3
### calculator ###
calc() {
    echo $*|bc -l
}

#controllo volume
### volume ###
vd() {
amixer set Master 1-| grep -o '\[.*%\]|tail -n1
}
vu() {
amixer set Master 1+| grep -o '\[.*%\]|tail -n1
}

#ricerche nel web tramite il browser w3m
#es: yahoo debianizzati
### web search ###
#yahoo
yahoo() {
    echo -e "\033[32msearching Yahoo for $*"
    local SEARCH=$(echo $* | sed -e 's/ /\%20/g')
    echo -e "\033[32mtranslating search to URL speak..."
    w3m http://it.search.yahoo.com/search?p="$SEARCH"
}

#ricerche tramite wordreference
wrenit() {
    echo -e "\033[32msearching WordReference for $*"
    local SEARCH=$(echo $* | sed -e 's/ /\%20/g')
    echo -e "\033[32mtranslating search to URL speak..."
    w3m http://www.wordreference.com/enit/"$SEARCH"
}
```

```
writen() {
    echo -e "\033[32msearching WordReference for $*"
    local SEARCH=$(echo $* | sed -e 's/ /\%20/g')
    echo -e "\033[32mtranslating search to URL speak..."
    w3m http://www.wordreference.com/iten/"$SEARCH"
}

#ricerche tramite wikipedia
wikit() {
    echo -e "\033[32msearching Wikipedia for $*"
    local SEARCH=$(echo $* | sed -e 's/ /\%20/g')
    echo -e "\033[32mtranslating search to URL speak..."
    w3m http://it.wikipedia.org/wiki/"$SEARCH"
}

wiken() {
    echo -e "\033[32msearching Wikipedia for $*"
    local SEARCH=$(echo $* | sed -e 's/ /\%20/g')
    echo -e "\033[32mtranslating search to URL speak..."
    w3m http://en.wikipedia.org/wiki/"$SEARCH"
}

#converte testo in html tramite vim
#necessita di vim e non vim-tiny
### text2html ###
2html() {
    if [ $# -lt 1 ] || [ $# -gt 2 ]; then
        echo -e "\033[32mconvert text to html"
        echo -e "\033[32musage: 2html filename"
    else
        vim -n -c 'so $VIMRUNTIME/syntax/2html.vim' -c 'wqa' $1 > /dev/null 2> /dev/null
    fi
}

#converte testo in ps e successivamente in pdf
#es: 2pdf .bashrc
### text2pdf ###
2pdf() {
    a2ps $1 --borders=0 -1 -f 10 -B -q -o - | ps2pdf - $1.pdf
}
```

```

}

#cerca nei repositories un binario e mostra a quale
#pacchetto appartiene (un surrogato di apt-file)
#eventualmente modificare "testing" con altro ramo nel link
### binpack ###
binpack() {
    if [ "$1" = "" ]; then
        echo -e "\033[32mshow package that contains the program"
        echo -e "\033[32musage: binpack <program>"
    else
        w3m -dump "http://packages.debian.org/search?searchon=contents&keywords=
                "$@"&mode=path&suite=testing&arch=any"|grep --color=never -B 4 /
    fi
}

#mostra lista dei file di un pacchetto non installato
#eventualmente modificare "amd64" con "i386" nel link
### deblist ###
deblist() {
    if [ "$1" = "" ]; then
        echo -e "\033[32mshows the structure of files in a package not installed"
        echo -e "\033[32musage: deblist <package>"
    else
        w3m -dump http://packages.debian.org/squeeze/amd64/"$@"/filelist
                |grep --color=never -B 2 /
    fi
}

#mostra ip esterno
### myip ###
myip() {
    w3m -dump http://checkip.dyndns.org|awk '{print $4}'
}

### indirizzi alternativi
#w3m -dump http://ipid.shat.net/iponly
#w3m -dump http://ip.dnsexit.com

```



```

ko() {
    local PID=$(pidof $1)
    kill -9 $PID
    echo "$1 ($PID) is dead"
}

#promemoria fatto in casa
#aggiunge al file TODolist le proprie note
#il costrutto if non e' realmente necessario in quanto
#echo crea comunque il file se non esiste, ma mi piaceva :)
#es: todo comprare cd
### post-it ###
todo() {
    local DAFILE="$HOME/TODolist"
    if [ ! -e $DAFILE ]; then
        touch $DAFILE
    fi
    echo $* >> $DAFILE
}

#lista numericamente il promemoria
alias todor='nl $HOME/TODolist'

#cancella riga secondo il numero ottenuto precedentemente
#es: todod 3 (cancella la terza riga)
todod() {
    sed -i ""$1"d" $HOME/TODolist
}

```

Alias e funzioni possono anche essere, come si è detto, utilizzati con profitto assieme alle numerose applicazioni da terminale disponibili nella maggioranza delle distribuzioni Linux. Un campo di applicazione nel quale è molto evidente la convenienza di impostare degli alias è, ad esempio, il multimediale. Mplayer è una delle applicazioni maggiormente diffuse in ambito unix anche per la possibilità di essere interamente gestito mediante terminale. E' possibile automatizzare molte delle sue funzioni mediante gli alias

```

#mplayer per filmati rai
#la variabile $1 è l'url del canale desiderato

```

```

#es:
#http://mediapolis.rai.it/relinker/relinkerServlet.htm?cont=983
alias uaplayer='mplayer -user-agent "Linux" -cache 666 -playlist $1'

#crea playlist nella home leggendo ricorsivamente in /media/dati/ascolto
alias playlist='fapg -r --format=m3u --output=$HOME/allmuzik.m3u /media/dati/ascolto'

#riproduce la playlist
alias muzik='mplayer -shuffle -playlist $HOME/allmuzik.m3u'

#mplayer per webcam con filtro per scattare immagini
#nohup disimpegna il terminale
alias wbcplayer='nohup &>/dev/null mplayer tv:// -really-quiet -vf screenshot &'

#registra video dalla webcam con mencoder aggiungendo data e ora
### mencoder webcam video ###
wbcrec() {
local DATE=$(date +%d-%m-%y%H,%M)
local NAME="$HOME/Desktop/mencam-$DATE.avi"
mencoder -really-quiet tv:// -ofps 25 -nosound -ovc lavc -o $NAME
}

#visualizza video youtube tramite mplayer e youtube-dl
#sia in una console con framebuffer che in un Xserver
#es: mtube http://www.youtube.com/watch?v=xxxxxxx
mtube() {
if [ -z $DISPLAY ]; then
    mplayer -really-quiet -cache 1111 -vo fbdev -vf scale=1280:1024 $(youtube-dl \
        -bg "$1")
else
    mplayer -really-quiet -cache 1111 $(youtube-dl -bg "$1")
fi
}

```

Restando in tema multimediale è possibile anche preparare alias per la conversione di formati

```

#flac2mp3
for maggio in *.flac; do $(flac -scd "$maggio" | lame -Sb 160 - \

```

```

mp3/"${maggio%.flac}.mp3"); done

#flac2ogg
for maggio in *.flac; do $(flac -scd "$maggio" | oggenc -Qo ogg/"${maggio%.flac}.ogg"\
-b 160 -); done

#wav2mp3
for maggio in *.wav; do $(lame -Sb 160 "$maggio" mp3/"${maggio%.wav}.mp3"); done

#ogg2ogg (riduce il bitrate)
for maggio in *.ogg; do $(oggdec -Qo - "$maggio" | oggenc -Qo ogg/"$maggio" -b 64 -);
done

```

mentre per funzioni di grande importanza quali il backup di dati o la gestione di archivi è possibile impostare alcune funzioni un po' più complesse come

```

#crea archivi (tar.bz2/tar.gz/tgz/zip) di file e/o directory
### archive wrapper ###
archive() {
    if [ "$1" = "" ]; then
        echo -e "\033[32mcreates an archive"
        echo -e "\033[32musage: archive <foo.tar.bz2> /foo /bar baz/"
    else
local DATE=$(date +%d-%m-%y)
local NAME=$DATE-$1
        case $NAME in
            *.tar.bz2) shift && tar cPpjf $NAME $* ;;
            *.tar.gz)  shift && tar cPpzf $NAME $* ;;
            *.tgz)    shift && tar czf $NAME $* ;;
            *.zip)    shift && zip $NAME $* ;;
        esac
    fi
}

#crea un tar.bz2 di una directory
#o file preservandone il nome (per quanto possibile :)
jmkbz2() {
local DATE=$(date +%d-%m-%y)

```

```

local NAME=$(basename $0)
  tar cPpjf $NAME-$DATE.tar.bz2 "$@"
}

#estrae ogni tipo di archivio
#es: extract archivio.gz
### extract archives ###
extract() {
if [ -f $1 ] ; then
case $1 in
*.tar.bz2) tar xjf $1 ;;
*.tar.gz) tar xzf $1 ;;
*.bz2) bunzip2 $1 ;;
*.rar) unrar x $1 ;;
*.gz) gunzip $1 ;;
*.tar) tar xf $1 ;;
*.tbz2) tar xjf $1 ;;
*.tgz) tar xzf $1 ;;
*.zip) unzip $1 ;;
*.Z) uncompress $1 ;;
*.7z) 7z x $1 ;;
*) echo "'$1' cannot be extracted via extract()" ;;
esac
else
echo "'$1' is not a valid file"
fi
}

```

E se dopo tanto lavoro occorre tirare un po' il fiato prima di riposare vanno imposti altri due alias come

```

#mostra vignette dal sito xkcd
#necessita di display dal pacchetto imagemagick
### xkcd strips ###
xkcd() {
while true;do wget -q http://dynamic.xkcd.com/comic/random -O -|grep -Eo \
'http://imgs.xkcd.com/comics/.*(png|jpg)'\|wget -q -i - -O -|display -geometry \
+200+200; done

```

```
}
```

```
#noia? qualche perla di saggezza a mani libere  
alias fortune='while true|echo;do fortune -a;sleep 11;done'
```

Sul finire occorre spendere qualche parola sulla gestione degli alias che abbiamo creato. Per la cancellazione di un alias è sufficiente la sua rimozione dal file `.bash_aliases` ma non è infrequente l'esigenza di aggirare un alias temporaneamente piuttosto che rimuoverlo del tutto, e questo capita molto spesso quando come alias si è impostato un comando comunque disponibile per il sistema.

Torniamo per un attimo al nostro alias che elimina (`rm`) un file semplicemente spostandolo (`mv`) nella directory che abbiamo impostato come cestino

```
alias rm='mv "$@" --target-directory=$HOME/cestino'
```

un alias di questo genere rende di fatto non più disponibile il comando `rm` che sarà sempre sostituito da `bash` con `mv`. Ci sono due modi per recuperare momentaneamente l'uso del comando, e quindi il suo comportamento originale, senza rimuovere l'alias impostato su di esso, il primo è quello di specificare il percorso completo del comando, in questo caso

```
/bin/rm
```

il secondo è quello di racchiudere il comando tra apici

```
'rm' (o "rm")
```

Apici e percorsi completi (`path`) pur svolgendo una funzione analoga possono essere utilizzati secondo un criterio di opportunità che suggerisce l'uso degli apici all'interno del terminale, in quanto più comodi ed immediati di `path` che possono essere anche molto lunghi. Viceversa però il `path` completo è sempre da consigliarsi nella definizione dell'alias stesso in quanto permette di prevenire gli errori nella definizione di sequenze complesse di comandi.

Bash dà la possibilità di digitare un alias direttamente all'interno di un terminale che lo terrà in memoria fino alla fine della sessione di lavoro a meno che, cessata la sua utilità, non si utilizzi il comando `unalias` seguito dal nome dell'alias stesso (`$nome`) per rimuoverlo.

Da ultimo una considerazione su alias e funzioni impostate e la possibilità di richiamarle da parte dell'utente. E' possibile prevedere diversi modi di consultare le impostazioni effettuate e richiamare di volta in volta quelle che occorrono senza ricordare tutto a memoria. Per brevità e semplicità il comando suggerito è il seguente

```
$ cat ~/.bash_aliases | less
```

I possibili utilizzi degli alias sono, ovviamente, molti di più e rappresentano una delle numerose possibilità offerte dalla bash mediante la personalizzazione del `.bashrc`. Chissà che non si possa allargare di più il discorso se questo articolo sarà di un qualche interesse per il lettore.

5.3.2 Note di chiusura

Il presente articolo ha un carattere divulgativo senza pretese di completezza né di altro tipo, in quanto rappresenta solo una raccolta di esempi che possono, dando delle idee, invitare alla lettura dei manuali e della documentazione esistente.

gnomob

5.3.3 Ringraziamenti

skizzhg alias pc='startfluxbox'

che ha ideato il presente lavoro ed ha fornito la maggioranza degli esempi utilizzati (che a sua volta ringrazia tutti quelli dai quali ha tratto ispirazione nelle sue ricerche)

mm-barabba alias gtk?='echo Cogito, ergo KDE'

che mi ha proposto di collaborare alla sua stesura (ed amorevolmente minacciato per rispettare le scadenze!)

la comunità dei debianizzati

immancabile punto di riferimento per me nel corso degli anni

debian e tutti quanti concorrono al suo sviluppo ed al suo mantenimento

...debbo davvero spiegare il perché? :-)

Redattore: gnomob

Capitolo 6

Softwares in analisi



Approfondimenti e test su tutto il softwares per la nostra debian.

Il mondo del software libero ha raggiunto dimensioni più che ammirevoli.

In questa sezione cercheremo di presentarvi delle applicazioni utilizzabili con la nostra debian, nel modo più esaustivo possibile.

6.1 Mai più senza backup

6.1.1 Premessa

Con il presente articolo si intende portare all'attenzione del lettore un argomento molto importante sia per gli amministratori di sistema che per i comuni utenti, anche i più prevenuti rispetto alla materia trattata: il *backup*. Per quanto possa sembrare superfluo ricordarlo, il *backup* è una pratica indispensabile per la salvaguardia dell'integrità di un sistema informatico in quanto permette di proteggere i dati dai malfunzionamenti *hardware*, dalla cancellazione (sia essa involontaria o dolosa) e dalla sottrazione indebita. È conoscenza comune, infatti, che il valore dei dati contenuti all'interno di un dispositivo di memorizzazione sia spesso di gran lunga superiore a quello del dispositivo medesimo. Pertanto, sia che si tratti di dati di rilevanza strategica per un'azienda (ad esempio, l'archivio disegni di un ufficio tecnico) che di dati personali di un utente casalingo (ad esempio, la raccolta di foto digitali accumulate in anni di scatti) è altamente consigliabile poter contare in caso di necessità su una procedura sistematica di *backup*. È del tutto evidente, infatti, che in assenza di essa, a fronte di un evento imprevisto, si corre il rischio di perdere parte o tutti i dati coinvolti.

Esistono molte strategie e tantissime soluzioni tecniche sia *hardware* che *software* per instaurare una procedura di *backup*, ognuna con i propri pregi e difetti, tanto che non basterebbe, probabilmente, un intero trattato per descriverle tutte. Stante la complessità dell'argomento, si è ritenuto di proporre in questo primo articolo solo due soluzioni, ciascuna basata su un diverso approccio tecnico e, quindi, con i propri vantaggi e difetti:

- la prima soluzione è basata sull'utilizzo del programma **backup2l** che esegue il *backup* limitandolo, in base alla configurazione, ai soli *file* contenenti dati di interesse per l'utente;
 - ha il pregio di essere semplice, efficace ed attivabile all'interno di un sistema in normali condizioni di esercizio con la possibilità di personalizzare il ripristino dei dati a discrezione dell'utilizzatore (ripristino anche di singoli *file*) e di storicizzare i *backup* eseguiti (con ripristino di una specifica revisione del file interessato in un determinato punto nel tempo);
 - ha il difetto di poter essere difficilmente usato per eseguire il *backup* di un intero disco comprensivo di quelle parti indispensabili al funzionamento del sistema operativo;

- la seconda soluzione è basata sull'utilizzo di una *suite* (insieme di programmi fra di loro collegati) chiamata **clonezilla** avviata da una periferica dedicata (ad esempio, un CD-ROM) e finalizzata ad eseguire una copia integrale del contenuto della memoria di massa (o di sue parti) come un'unica entità logica: per questa tecnica si adotta spesso il termine *clonazione*;
 - ha il pregio di permettere il ripristino del contenuto dell'intera periferica sottoposta a *backup*, con ciò comprendendo anche il sistema operativo (se presente) e perfino quelle parti della periferica estranee a quest'ultimo, ma indispensabili al suo corretto avvio e funzionamento;
 - ha il difetto di richiedere il fermo del sistema su cui è utilizzata ed è, inoltre, impossibile ripristinare singoli *file* (sono ripristinati simultaneamente tutti i *file* contenuti nella periferica con la cancellazione del contenuto preesistente nel disco a tal fine utilizzato), né è possibile storicizzare il *backup* di singoli *file*.

Nei paragrafi di seguito riportati sono esaminate le caratteristiche peculiari di questi due strumenti.

6.2 backup2l

backup2l è uno programma (*script bash*) che, secondo le parole dell'autore, è destinato al *low-maintenance backup/restore tool for mountable media*, cioè uno strumento leggero da utilizzare dalla riga di comando per la generazione, la manutenzione e il ripristino dei *backup*; gli archivi creati a tal fine sono registrati in una parte del *file system* a ciò dedicata dall'utente (ad esempio, la partizione di un disco rigido appositamente collegato al *file system* o qualunque altra sua parte).

La progettazione del programma è stata finalizzata a:

- minimizzare la manutenzione da parte dell'amministratore: una volta installato e configurato, non richiede altri particolari interventi in quanto è avviato giornalmente in automatico dallo *scheduler* di sistema (*cron*);
- ottenere trasparenza nelle procedure di archiviazione: richiama programmi ben conosciuti e consolidati (come *tar*, *afio* e *bzip2*) per la generazione e compressione degli archivi prodotti;

- robustezza nei risultati: implementazione di un algoritmo per *backup gerarchici differenziali* con un numero di livelli e sottolivelli personalizzabili che prevede la rotazione degli archivi creati in modo da impiegare in modo efficiente lo spazio disponibile sulle periferiche utilizzate (per maggiori dettagli sullo schema di rotazione, consultare la pagina di manuale con il comando *man backup2l*).

Per decidere se un *file* deve essere incluso nel *backup* (è nuovo o modificato), il programma, dopo aver verificato i criteri di inclusione configurati dall'utente, ne controlla il nome, la data di modifica, le dimensioni, le proprietà e le autorizzazioni: ciò è eseguito con alcune cautele in modo da mantenere la compatibilità con *file system* di altri sistemi operativi. Tutti i *file* di controllo sono memorizzati congiuntamente al *file* dell'archivio dati nella *directory* di *backup* e il loro contenuto è per lo più autoesplicativo: in caso di emergenza, l'utente non solo può fare affidamento sulle funzionalità di ripristino del programma, ma può - se necessario - accedere anche direttamente agli archivi dati per estrarne manualmente il contenuto. Qualora si preveda che il volume dei dati destinati al *backup* ecceda le dimensioni della periferica rimovibile eventualmente destinata ad archivarli, è possibile utilizzare una funzione di *split-and-collect* per fare in modo che gli archivi abbiano una dimensione che li renda ripartibili su più supporti. Infine, le funzionalità di ripristino, da attivare manualmente, consentono di ricreare facilmente i *file* o le *directory* a partire dagli archivi di *backup* (comprensivi degli attributi originali) .

6.2.1 Installazione

Il programma è disponibile nei *repository on-line* di Debian GNU/Linux e, quindi, può agevolmente essere installato con i consueti gestori di pacchetti, ad esempio, impartendo come utente *root* i seguenti comandi da terminale:

```
# aptitude update
# aptitude install backup2l
```

È bene ricordare che *backup2l* raccomanda l'installazione dei pacchetti *tar* e *afio* e consiglia l'installazione dei pacchetti *bzip2* e *cdlabelgen*.

6.2.2 Configurazione

La configurazione è piuttosto semplice e consiste nella impostazione di alcuni parametri del file `/etc/backup2l.conf` che, di fatto, è uno *script bash* e, quindi, le righe precedute dal carattere `'#'` rappresentano commenti, mentre tutti i rimanenti contenuti rappresentano istruzioni per il programma. Di seguito saranno riportati solo i parametri di cui è indispensabile conoscere l'esistenza per utilizzare da subito *backup2l*.

Volume identification

il parametro `VOLNAME` specifica il prefisso utilizzato dal programma per attribuire la prima parte del nome del *file* destinato ad archiviare i dati secondo lo schema di *backup differenziale*; ad esempio:

```
# Volume identification
# This is the prefix for all output files;
# multiple volumes can be handled by using different configuration files
VOLNAME="all"
```

qualora sia mantenuto il parametro di *default* `all`, gli archivi di *backup* saranno sistematicamente generati, ad esempio, con i seguenti nomi:

```
all.1
all.11
all.111
...
```

dove i numeri posposti al prefisso `all` rappresentano i livelli e sottolivelli di ciascun *backup differenziale* che si è succeduto nel tempo.

Source list

Il parametro `SRCLIST` indica al programma l'elenco delle *directory* del proprio *file system* da includere nel *backup*; per ciascuna di esse' è specificato il percorso completo (*path*) a partire da *directory radice* (`'/'`) ed uno spazio di separazione dalla successiva voce nell'elenco; ad esempio, il valore di *default* del parametro prevede di includere le *directory* `/etc`, `/root`, `/home`, `/var/mail`, `/usr/local`:

```
# Source files
# List of directories to make backups of.
# All paths MUST be absolute and start with a '/'!
SRCLIST=(/etc /root /home /var/mail /usr/local)
```

naturalmente, il valore di questo parametro potrà e dovrà essere modificato in ragione dell'esigenze dell'utente e della collocazione nel *file system* dei dati di suo interesse.

Skip condition

Il parametro SKIPCOND permette di specificare, se necessario, eventuali criteri di esclusione dal *backup* per *file* che soddisfano i requisiti del parametro SRCLIST. Ad esempio, volendo includere nel *backup* la *directory* del progetto di un programma scritto in linguaggio C, si potrebbe volerne includere tutto il contenuto tranne gli *object file* (quelli che terminano con l'estensione .o) prodotti dal compilatore e propedeutici alla generazione dell'eseguibile, ma ritenuti superflui ai fini del *backup*. Ciò può essere ottenuto specificando il *pattern* del nome dei *file* che si intendono escludere secondo la sintassi del comando *find* (consultare il manuale del comando per maggiori dettagli). Ad esempio, il valore di *default* previsto per tale parametro esclude tutti i *file* il cui nome contiene la sequenza di caratteri *.nobackup* oppure (clausola -o che rappresenta un *or* logico) che terminano con *.o*:

```
# The following expression specifies the files not to be archived.
# See the find(1) man page for further info. It is discouraged to
# use anything different from conditions (e. g. actions) as it may have
# unforeseeable side effects.

# This example skips all files and directories with a path name containing
# '.nobackup' and all .o files:
SKIPCOND=(-path "*.nobackup*" -o -name "*.o")
```

Qualora, ad esempio, volessimo escludere anche tutti i *file* che terminano con l'estensione *.tmp* il parametro potrebbe essere così modificato:

```
SKIPCOND=(-path "*.nobackup*" -o -name "*.o" -o -name "*.tmp")
```

Questo parametro è estremamente utile soprattutto in sistemi contenenti grandi volumi di dati in quanto, come è esperienza comune, all'interno di essi sono presenti un gran numero di *file* che è possibile escludere con sicurezza dal *backup* ottimizzando lo spazio occupato sul dispositivo a ciò destinato. In ogni caso, qualora non si abbia bisogno di escludere alcun *file*, il parametro può essere semplicemente annullato commentandolo nel *file* di configurazione, cioè antepoendo ad esso il carattere '#'; ad esempio:

```
# SKIPCOND=(-path "*.nobackup*" -o -name "*.o")
```

Backup directory (destination)

Il parametro BACKUP_DIR specifica il percorso completo (*path*) a partire da *directory radice* ('/') all'interno del quale il programma genererà gli archivi di *backup* ed i *file* di controllo; nello stesso percorso esso cercherà gli archivi di *backup* in caso di ripristino; ad esempio, il valore di *default* del parametro è /disk2/backup :

```
# Destination directory for backups;  
# it must exist and must not be the top-level of BACKUP_DEV  
BACKUP_DIR="/disk2/backup"
```

Naturalmente, esso **deve essere cambiato indicando la collocazione prescelta sul proprio sistema** altrimenti il programma non funzionerà.

Backup parameters

I parametri di seguito indicati controllano l'algoritmo di *backup gerarchico differenziale* utilizzato dal programma:

- MAX_LEVEL specifica il numero dei livelli (da 1 a 9): si ricorda che i *backup completi* corrispondono al livello zero, gli altri livelli corrispondono a *backup differenziali*;
- MAX_PER_LEVEL specifica il numero di *backup differenziali* da eseguire per ogni livello (da 1 a 9);
- MAX_FULL specifica il numero dei *backup completi* da mantenere (da 1 a 8);

- GENERATIONS specifica il numero di generazioni da conservare per ogni *backup* , solitamente è accettabile il valore di default 1;
- CREATE_CHECK_FILE controlla la generazione del *file di check*.

I valori di *default* previsti dal programma sono adeguati alla maggior parte dei casi d'uso e sono di seguito riportati:

```
# Backup parameters
```

```
# Number of levels of differential backups (1..9)
MAX_LEVEL=3
```

```
# Maximum number of differential backups per level (1..9)
MAX_PER_LEVEL=8
```

```
# Maximum number of full backups (1..8)
MAX_FULL=2
```

```
# For differential backups: number of generations to keep per level;
# old backups are removed such that at least GENERATIONS * MAX_PER_LEVEL
# recent versions are still available for the respective level
GENERATIONS=1
```

```
# If the following variable is 1, a check file is automatically generated
CREATE_CHECK_FILE=1
```

Verifica della configurazione dello scheduler

Terminata la configurazione di *backup2l* non resta che verificarne la predisposizione all'esecuzione tramite lo *scheduler* di sistema. Come impostazione di *default*, l'installazione di *backup2l* prevede la creazione di uno *script* nella cartella */etc/cron.daily* con il nome *zz-backup2l* il cui contenuto è il seguente:

```
#!/bin/bash
```

```
# The following command invokes 'backup2l' with the default configuration
# file (/etc/backup2l.conf).
```

```
#
# (Re)move it or this entire script if you do not want automatic backups.
#
# Redirect its output if you do not want automatic e-mails after each backup.

! which backup2l > /dev/null || nice -n 19 backup2l -b
```

Come si può vedere, lo *script*:

- effettua un controllo sull'esistenza del programma *backup2l* (! `which backup2l > /dev/null`)
- in caso affermativo (operatore `||` che esegue il comando alla sua destra se quello alla sua sinistra ha restituito il valore 1) esegue il programma `backup2l -b` (notare il comando `-b` che indica di avviare il *backup*)
- con una priorità di processo (programma `nice`) tale da interferire il meno possibile con il normale funzionamento del sistema (`nice -n 19 backup2l -b`).

Durante l'esecuzione del *backup*, il programma genera un output che lo *scheduler*, di norma, allega ad un email inviato all'amministratore di sistema con la registrazione delle attività svolte.

ATTENZIONE! per poter ricevere le email così generate è indispensabile aver configurato un MTA (*mail transfer agent*) sul proprio sistema (come, ad esempio, `exim`).¹

Il *backup* è, quindi, eseguito almeno una volta al giorno, all'orario specificato nel *file* di configurazione `/etc/crontab`; resta inteso che il *backup* non sarà eseguito qualora all'orario previsto il sistema fosse spento oppure lo *scheduler* non fosse in attivo.

6.2.3 Esempio di un report di *backup*

A partire da una configurazione che prevede i seguenti parametri:

```
MAX_LEVEL=2
MAX_PER_LEVEL=9
MAX_FULL=2
```

¹http://people.debian.org/osamu/pub/getwiki/html/ch06.en.html#_mail.transport_agent.mta

backup2l genera un resoconto completo che è inviato per email, come nell'esempio sotto riportato:

```
/etc/cron.daily/zz-backup2l:  
backup2l v1.4 by Gundolf Kiefer
```

```
Sun Aug 30 06:35:36 CEST 2009
```

```
Running pre-backup procedure...
```

```
pre-backup: nothing to do
```

```
Removing old backups...
```

```
removing <all.11>
```

```
removing <all.12>
```

```
removing <all.13>
```

```
removing <all.14>
```

```
removing <all.15>
```

```
removing <all.16>
```

```
removing <all.17>
```

```
removing <all.18>
```

```
removing <all.19>
```

```
removing <all.1>
```

```
moving <all.21> to <all.11>
```

```
moving <all.22> to <all.12>
```

```
moving <all.23> to <all.13>
```

```
moving <all.24> to <all.14>
```

```
moving <all.25> to <all.15>
```

```
moving <all.26> to <all.16>
```

```
moving <all.27> to <all.17>
```

```
moving <all.281> to <all.181>
```

```
moving <all.282> to <all.182>
```

```
moving <all.283> to <all.183>
```

```
moving <all.284> to <all.184>
```

```
moving <all.285> to <all.185>
```

```
moving <all.286> to <all.186>
```

```
moving <all.287> to <all.187>
```

```
moving <all.288> to <all.188>
```

```
moving <all.289> to <all.189>
```

```
moving <all.28> to <all.18>
```

```
moving <all.291> to <all.191>
moving <all.292> to <all.192>
moving <all.293> to <all.193>
moving <all.294> to <all.194>
moving <all.295> to <all.195>
moving <all.296> to <all.196>
moving <all.297> to <all.197>
moving <all.298> to <all.198>
moving <all.299> to <all.199>
moving <all.29> to <all.19>
moving <all.2> to <all.1>
removing <all.181>
removing <all.182>
removing <all.183>
removing <all.184>
removing <all.185>
removing <all.186>
removing <all.187>
removing <all.188>
removing <all.189>
```

```
Preparing full backup <all.2>...
```

```
57239 / 57239 file(s), 10984 / 10984 dir(s), 5.1GB / 5.1GB (uncompressed)
skipping: 0 file(s), 0 dir(s), 0 B (uncompressed)
```

```
Creating archive using 'DRIVER_TAR_GZ'...
```

```
Checking TOC of archive file (< real file, > archive entry)...
```

```
9210c9210
```

```
< /home/mambarabba/data/karamba/jads/monitor-em-a??ao.png
```

```
---
```

```
> /home/mambarabba/data/karamba/jads/monitor-em-a\303\247ao.png
```

```
Creating check file for <all.2>...
```

```
Running post-backup procedure...
```

```
post-backup: nothing to do
```

```
Sun Aug 30 06:50:58 CEST 2009
```

Summary

=====

```
Backup Date Time | Size | Skipped Files+D | New Obs. | Err.
```

```
-----
all.1 2009-05-22 06:33 |4747413K | 0 63157 |63157 0 | 1
all.11 2009-06-01 06:33 | 25601K | 0 63779 | 1021 399 | 0
all.12 2009-06-11 06:33 | 11918K | 0 63781 | 179 177 | 0
all.13 2009-06-21 06:33 | 18581K | 0 64058 | 471 194 | 0
all.14 2009-07-01 06:33 | 39216K | 0 67235 | 5663 2486 | 0
all.15 2009-07-11 06:38 | 46210K | 0 67812 | 1477 900 | 0
all.16 2009-07-21 06:34 | 21308K | 0 67809 | 722 725 | 0
all.17 2009-07-31 06:34 | 22159K | 0 67944 | 875 740 | 0
all.18 2009-08-10 06:39 | 12044K | 0 68065 | 336 215 | 0
all.19 2009-08-20 06:40 | 43597K | 0 68194 | 1507 1378 | 0
all.191 2009-08-21 06:32 | 14503K | 0 68200 | 158 152 | 0
all.192 2009-08-22 06:33 | 11011K | 0 68205 | 80 75 | 0
all.193 2009-08-23 06:33 | 10960K | 0 68217 | 44 32 | 0
all.194 2009-08-24 06:33 | 10961K | 0 68228 | 43 32 | 0
all.195 2009-08-25 06:33 | 11141K | 0 68202 | 79 105 | 0
all.196 2009-08-26 06:35 | 11023K | 0 68213 | 43 32 | 0
all.197 2009-08-27 06:35 | 21848K | 0 68205 | 759 767 | 0
all.198 2009-08-28 06:44 | 11104K | 0 68198 | 209 216 | 0
all.199 2009-08-29 06:43 | 10979K | 0 68213 | 47 32 | 0
all.2 2009-08-30 06:35 |4842014K | 0 68223 |68223 0 | 1
```

```
Filesystem Size Used Avail Use% Mounted on
/dev/hdb1 38G 9.7G 27G 28% /mnt/backup
```

Come potete vedere, il resoconto è ben dettagliato e qualora vi fossero errori, sono segnalati con il riferimento al *path* completo del *file* interessato per consentire un intervento mirato.

6.2.4 Sintassi

Si riporta di seguito la sintassi del programma tratta dalla pagina di manuale con i riferimenti ritenuti più interessanti:

```
backup2l [ -c conffile ] [ -t backup-ID ] command
```

Opzioni

- **-c, --conf *conf file***: se presente, permette di specificare il file di configurazione qualora fosse diverso da quello di *default* (*etc/backup2l.conf*).
- **-t, --time *BID***: se presente, questa opzione seleziona uno specifico *backup* utilizzando come riferimento il **BID** (**backup ID** o identificativo di *backup*), cioè un identificativo univoco che il programma attribuisce a ciascun archivio di *backup* in funzione del livello e sottolivello a cui si riferisce (consultare il manuale del programma per maggiori dettagli); su tale archivio così selezionato saranno applicati i successivi comandi **--locate** e **--restore**.

Comandi

- **-h, --help**: visualizza alcune istruzioni sintetiche sull'uso del programma;
- **-e, --estimate [*level*]**: visualizza il numero di file, il volume di dati ed altre informazioni relative al *backup* che sarebbe generato secondo la configurazione corrente qualora fosse avviato; non sono creati, né rimossi, archivi; se specificato, il valore del parametro *level* può essere utilizzato in sostituzione di quello previsto per il parametro **MAX_LEVEL** nella configurazione;
- **-b, --backup [*level*]**: esegue un nuovo *backup* rimuovendo gli archivi più vecchi secondo quanto previsto dai parametri della configurazione; se specificato, il valore del parametro *level* può essere utilizzato in sostituzione di quello del parametro **MAX_LEVEL** nella configurazione;
- **-s, --get-summary**: visualizza una tabella riassuntiva relativa ai *backup* eseguiti;
- **-a, --get-available [*pattern list*]**: visualizza l'elenco dei *file* che sono rimossi o aggiunti nei *backup* ('+' indica un file nuovo che è stato aggiunto, '-' indica un file che è stato rimosso o sostituito); se è fornito il parametro **pattern list** l'elenco può essere filtrato secondo la *espressione regolare* fornita; inoltre, tutti i messaggi di stato sono destinati allo *standard error*, in modo che possano essere facilmente reindirizzati;

ATTENZIONE! il *pattern di ricerca* è applicato non solo ai nomi dei *file*, ma anche a tutte le voci presenti nell'output generato a partire dal contenuto dell'archivio di

backup: è, quindi, possibile eseguire ricerche anche su altre proprietà dei *file* come, ad esempio, il creatore o la data di modifica. Qualora si desiderasse limitare la ricerca solo nome del *file*, è possibile far precedere il *pattern di ricerca* da `< space > /*.`

- `-l, --locate [pattern list]`: visualizza la *path* in cui saranno ripristinati i *file* dell'ultimo *backup* o di quello specificato con l'opzione `--time`; se fornito, il parametro `pattern list` è utilizzato in analogia a quanto già indicato per il comando `--get-available`. Sono interessati i *file attivi* cioè quelli presenti nel sistema al momento del *backup*. In sintesi, quindi, questo comando indica quali *file* saranno estratti, da quale archivio per ripristinare lo stato del sistema al momento dello specifico *backup* selezionato;
- `-r, --restore [pattern list]`: esegue le stesse azioni del comando `--locate`, ma in più esegue fisicamente il ripristino del *backup*; è bene precisare che i *file* sono ripristinati nella *directory* corrente, cioè quella da cui è eseguito il comando `backup21 -r` e visualizzabile con il comando `pwd`;

ATTENZIONE! è possibile, ma è **sconsigliato**, ripristinare i *file* del *backup* nelle loro *path* originali eseguendo il comando `cd /` prima del comando `backup21 -r`

- `-p, --purge BID list`: rimuove l'archivio (o gli archivi) specificato/i, e tutti gli altri archivi di *backup* (ricordiamo che è utilizzata la tecnica di *backup differenziale*);
- `-m, --make-check [BID list]`: crea un 'file di controllo' usando il programma `md5sum`; se nessun *BID (backup ID)* è specificato, il *file* è creato per tutti gli archivi che ne sono sprovvisti;
- `-v, --verify [BID list]`: verifica l'integrità degli archivi di *backup* usando il *file di controllo* precedentemente generato; se nessun *BID (backup ID)* è specificato, tutti gli archivi esistenti sono controllati; se il *file di controllo* non è stato generato, è verificata solo l'esistenza del file e l'integrità del *file* secondo quanto previsto dal programma di archiviazione utilizzato;
- `-x, --extract capacity max-free BID-list`: organizza gli archivi dei *backup* specificati in modo da poter essere a loro volta archiviati su supporti rimovibili, come i CD-ROM, di limitata capacità (pari a *capacity MByte*), lasciando su ciascuno

di essi al massimo *max-free MByte* liberi; il parametro *BID-list* può anche essere specificato attraverso *wildcards* come, ad esempio, `1 '2*'`; il comando genera un elenco di *directory* numerate ciascuna delle quali rappresenta il supporto su cui dovrà essere trasferito; la procedura preserva l'ordine degli archivi di *backup* e gli archivi le cui dimensioni eccedono quella del supporto sono suddivisi in più *file* con un numero seriale aggiunto al nome di ciascuno di essi. Le operazioni sono interattive: basta eseguire il comando per capire le attività eseguite ed è possibile interromperlo qualora esse fossero difformi da quanto desiderato.

6.2.5 Utilizzo

Backup manuale

L'uso di *backup2l* è piuttosto semplice. Come precedentemente accennato, una volta correttamente completata la configurazione, il *backup* è eseguito automaticamente su base giornaliera qualora il sistema e il suo *scheduler* siano operativi all'orario previsto. Ciò non di meno, è possibile avviare *backup2l* manualmente, ad esempio nel caso in cui sia avvenuta, nel corso della stessa giornata, un'importante variazione ai dati successivamente all'attivazione automatica della procedura; in tal caso, il comando da impartire è:

```
# backup2l -b
```

È bene ricordare che tale comando di norma è utilizzato con l'utenza di *root* (ad esempio, quando è avviato dal *scheduler* di sistema).

Ripristino

Il comando di ripristino deve essere necessariamente impartito manualmente generalmente con utenza di *'root'*: fortunatamente, è un'operazione molto semplice sebbene vada eseguita con alcune cautele. In particolare, come già richiamato nella spiegazione del comando *-restore*, è bene precisare che *backup2l* per *default* ripristina i *file* provenienti dagli archivi nella *directory* corrente, cioè quella da cui è eseguito e visualizzabile con il comando `pwd`; come abbiamo prima citato, **è possibile, ma è sconsigliato, ripristinare i *file* del *backup* nelle loro *path* originali eseguendo il comando `cd /` prima del comando `backup2l -r`**. Lo stesso autore del programma, Gundolf Kiefer, da me contattato in merito a tale argomento, ha così risposto:

«You can invoke `backup2l -r ...` from `/` to get the files where they were, but that is not recommended as you may accidentally overwrite important files by older versions. It is wiser to restore into a temporary directory first and then move the files wherever you want»

«È possibile richiamare `backup2l-r ...` da `/` [directory radice, ndr] per riportare i file dove erano, ma non è raccomandato in quanto potrebbe accidentalmente sovrascrivere file importanti con versioni più vecchie. È più saggio eseguire il ripristino prima in una directory temporanea e poi spostare i file dove desiderato»

Volendo, quindi, seguire le cautele sopra indicate, prima di eseguire un ripristino, è preferibile:

1. creare una *directory* temporanea per l'estrazione del *backup*
2. portarsi al suo interno
3. eseguire il comando di ripristino

Nell'esempio sotto indicato, a partire dal *backup* con *BID* (*backup ID*) pari a `all.153`, è ripristinato solo il *file* `/home/html/user/prova.txt` perchè espressamente specificato:

```
# mkdir /home/user/backupx
# cd /home/user/backupx
# backup2l -t all.153 -r /home/html/user/prova.txt
```

Alcune osservazioni:

- Qualora non fosse stato specificato uno specifico *backup* attraverso l'opzione `-t BID`, sarebbe stato estratto il contenuto dell'ultimo ad essere stato eseguito;
- qualora non fosse stato specificato il nome di uno specifico *file*, sarebbe stato estratto l'intero contenuto del *backup*

Una volta completato il ripristino del 'file' specificato, esso comparirà nella cartella temporanea:

```
/home/user/backupx/home/html/user/prova.txt
```

dove `/home/user/backupx` è la *path* della *directory* temporanea.

6.2.6 Conclusioni

backup2l è un ottimo strumento, al contempo semplice e potente. È particolarmente indicato per il *backup* dei dati di un'installazione di tipo *server*, ma è comunque possibile utilizzarlo con soddisfazione anche in un'installazione di tipo domestica. È bene ricordare, comunque, che il programma esegue dei *backup differenziali gerarchici* e, quindi, che il ripristino, generalmente, coinvolge non uno, ma più archivi, cioè è necessario risalire nella gerarchia dei *backup differenziali* qualora il *file* ricercato non abbia subito recenti modifiche: pertanto, per il buon esito del ripristino, è indispensabile non modificare manualmente gli archivi generati dal programma per non interferire con l'algoritmo di rotazione previsto dallo stesso.

6.2.7 Riferimenti

<http://backup2l.sourceforge.net/>

<http://www.debianhelp.co.uk/backup2l.htm>

6.3 Clonezilla

6.3.1 Introduzione

Le tecniche ordinarie di *backup* replicano i *file* contenuti nei dispositivi di memoria di massa (dischi rigidi o altre periferiche riconosciute come tali) ad eccezione di alcune parti che, a causa del funzionamento interno del sistema operativo, è preferibile non duplicare o sono difficilmente duplicabili. Inoltre, le tecniche ordinarie di *backup* allo scopo di ottimizzare l'uso delle risorse tendono a selezionare i contenuti di maggiore interesse per l'utente escludendo quelli che, essendo poco soggetti a modifiche nel tempo, possono essere stabilmente reperiti da altre fonti (come, ad esempio, il sistema operativo). Pertanto, in caso di malfunzionamento irrisolvibile della periferica sottoposta a *backup*, la procedura di ripristino prevede generalmente la sostituzione dell'hardware, la re-installazione del sistema operativo, la sua configurazione, la re-installazione dei programmi precedentemente utilizzati ed, infine, il ripristino del *backup* contenente i dati: ciò vale, a maggior ragione, per i sistemi operativi nei quali l'installazione delle applicazioni avviene in maniera semi-automatica attraverso programmi a ciò dedicati (come, ad esempio, nel caso di Debian GNU/Linux). Già da alcuni anni è stata sviluppata una tecnica di *backup* alternativa, consistente nella copia dell'intero contenuto della memoria di massa. In tal caso, il risultato della procedura

di *backup* è, quindi, teoricamente non più l'insieme del contenuto dei singoli *file* come entità logiche indipendenti, ma l'insieme dei dati che occupano determinate aree della memoria della periferica (generalmente chiamate partizioni). Il risultato, quindi, sarà una copia esatta a livello di singoli dati contenuti nella periferica, ovvero una copia in tutto e per tutto identica all'originale non già a livello di singoli *file*, ma di partizione: da qui il termine *clonazione*. Tale tecnica è realizzata attraverso un programma che, per ottemperare al proprio compito, deve accedere alla periferica interessata in modo esclusivo (quindi, quando essa è del tutto inutilizzata da altre applicazioni o dal sistema operativo), così come quando si desidera scattare una fotografia è indispensabile che il soggetto resti immobile. Da qui l'esigenza, nei casi più semplici, di avviare il computer ed il software di *clonazione* da una periferica (generalmente, un CD/DVD-ROM) e con un sistema operativo entrambi dedicati a questo compito.

La procedura di ripristino di un *backup* eseguito con la tecnica della *clonazione* permette, pur con alcune limitazioni che saranno successivamente esposte, di ottenere da subito una copia della memoria di massa il cui contenuto è esattamente identico all'originale e che, quindi, può essere immediatamente utilizzata per ripristinare il contenuto di un dispositivo malfunzionante oppure per preparare un altro dispositivo di sostituzione. Inoltre, la stessa tecnica può, di fatto, essere utilizzata anche per installare ex-novo un sistema operativo replicando un'installazione già disponibile e configurata.

Nell'ecosistema *open source* sono disponibili diverse soluzioni per la *clonazione*, tra le quali ci sembra utile portare all'attenzione del lettore il progetto **clonezilla**.

6.3.2 Il progetto *clonezilla*

clonezilla, sviluppato dal *The National Center for High-Performance Computing*², è un progetto rilasciato con licenza GPL, che supporta il *backup* e *restore* sia per una singola postazione (caso d'uso *stand-alone*), che simultaneamente per un maggior numero di postazioni diverse (caso d'uso *client - server*) collegate da un rete dati. In questa sede si tratterà il primo caso, la cui documentazione è reperibile all'indirizzo internet³ sotto forma di *reference cards*.

²<http://www.nchc.org.tw/en/>

³<http://clonezilla.org/download/sourceforge/doc/clonezilla-live-stable-doc.php>

6.3.3 Architettura

L'architettura di *clonezilla* si basa su un insieme di procedure (*shell script* in *bash* o *perl*) che, una volta avviate, gestiscono l'interfaccia utente e comandano l'esecuzione dei programmi direttamente interessati alla *clonazione* delle memorie di massa o delle loro partizioni; il codice sorgente di tali *script* è disponibile nel *repository* di *sourceforge* che ospita il progetto⁴. La clonazione è realizzata con i programmi **partclone**⁵, **ntfsclone**⁶ e **partimage**⁷ per partizioni che ospitano i seguenti *file system*: ext2, ext3, ext4, reiserfs, xfs, jfs per GNU/Linux, FAT, NTFS e HFS+; è, invece, utilizzato il più comune programma **dd**⁸ per le altre tipologie di *file system*: la differenza tra le due soluzioni è che nella prima la conoscenza della struttura logica del *file system* da parte dei programmi utilizzati consente l'ottimizzazione delle operazioni limitando la copia alle sole aree effettivamente utilizzate. È utile precisare che la documentazione riferisce che è disponibile il supporto per la seconda versione del *Logical Volume Manager* (LVM) di GNU/Linux, mentre manca per la precedente versione di LVM.

Allo scopo di accedere al dispositivo da clonare in modo esclusivo (come indicato in premessa), gli *shell script* e i programmi impiegati da *clonezilla* sono stati confezionati all'interno di un CD-ROM dal quale è possibile avviare il computer interessato; naturalmente, GNU/Linux è il sistema operativo contenuto nel CD-ROM ed utilizzato per avviare gli *shell script* e fornire le funzionalità di base necessarie ad utilizzare i programmi sopra indicati.

L'immagine in formato ISO utilizzabile per creare (masterizzare) il CD-ROM di *clonezilla* per la versione considerata stabile è reperibile all'indirizzo internet <http://clonezilla.org/download/sourceforge/stable/iso-zip-files.php>.

6.3.4 Funzionalità

La versione *stand-alone* di *clonezilla* implementa la funzionalità di copia (*backup*) e ripristino (*restore*) fruibili attraverso una modalità d'uso semplificata con menù preconfigurati (*wizard mode*) oppure una modalità più analitica (*expert mode*): in questa sede si farà riferimento al *wizard mode*.

⁴<http://clonezilla.org/download/sourceforge/src/sourcecodes.php>

⁵<http://partclone.org/>

⁶<http://www.linux-ntfs.org/doku.php?id=ntfsclone>

⁷ <http://www.partimage.org/>

⁸[http://it.wikipedia.org/wiki/Dd_\(Unix\)](http://it.wikipedia.org/wiki/Dd_(Unix))

La procedura di *backup* si articola, a grandi linee, nei seguenti passaggi:

1. avvio del computer interessato dal CD-ROM contenente *clonezilla*;
2. scelta della periferica destinata a contenere il *backup* risultante dalla clonazione;
3. scelta della periferica da clonare;
4. generazione, nella periferica destinata a contenere il *backup*, di una *directory* contenente un *file* corrispondente all'intero contenuto della periferica clonata e alcuni ulteriori *file* (che descrivono la struttura logica del disco e l'hardware impiegato dal computer in cui risiede) necessari al successivo ripristino;
5. spegnimento controllato del computer o ritorno al precedente punto 2) o 1)

Analogamente, la procedura di ripristino si articola, a grandi linee, nei seguenti passaggi:

1. avvio del computer interessato dal CD-ROM contenente *clonezilla*;
2. scelta della periferica che contiene il *backup* di una precedente clonazione e che si intende ripristinare;
3. scelta della periferica (o della partizione) di cui ripristinare il contenuto;
4. avvio della procedura di *restore* con ripristino del contenuto della periferica con i dati di un precedente *backup*;
5. spegnimento controllato del computer o ritorno al precedente punto 2) o 1)

6.3.5 Avvertenze prima dell'uso

È importante, a questo punto, **portare all'attenzione del lettore alcune fondamentali avvertenze d'uso:**

- **l'utilizzo di *clonezilla* richiede una conoscenza approfondita delle convenzioni del kernel Linux per la denominazione dei dispositivi di memoria di massa** (generalmente classificati *block device*^{9,10}) e delle partizioni in essi contenute¹¹; qualora il lettore abbia dei dubbi o incertezze in merito al riconoscimento di tali convenzioni è preferibile che non utilizzi *clonezilla*;

⁹<http://www.kernel.org/pub/linux/docs/device-list/devices.txt>

¹⁰http://tldp.org/HOWTO/html_single/Partition/#devices

¹¹http://tldp.org/HOWTO/html_single/Partition/

- **la procedura di ripristino implica la cancellazione irreversibile del contenuto della periferica o delle partizioni interessate:** pertanto, l'errata scelta del dispositivo sul quale sono ripristinati i dati causa la cancellazione del suo contenuto determinandone la perdita irreversibile; anche in tal caso, qualora il lettore abbia dei dubbi o incertezze in merito al riconoscimento delle periferiche e/o delle loro partizioni è preferibile che non utilizzi *clonezilla*;
- prima di utilizzare *clonezilla*, allo scopo di semplificare il riconoscimento delle partizioni, è **utile attribuire un'etichetta (*volume label*) a ciascuna partizione del disco rigido interessato** (ad esempio, con il comando *e2label*) qualora non lo si fosse già fatto in fase di installazione del sistema operativo ¹²;
- come specificato nella *reference card* di *clonezilla*, è caldamente consigliato di effettuare il *backup* dei dati contenuti delle periferiche interessate alla clonazione con altre e diverse tecniche prima di utilizzare *clonezilla*;

ATTENZIONE! - Incorrect use of the software is easy and can be harmful to the data. Always save your data first!, ¹³

- *clonezilla* è fornito dai suoi sviluppatori secondo quanto previsto dalla licenza GPL senza alcuna garanzia e senza alcuna responsabilità in merito alle conseguenze relative all'uso dello stesso;

ATTENZIONE! - resta inteso che le informazioni e i suggerimenti contenuti nel presente manoscritto hanno solo carattere amatoriale e sperimentale; pertanto, chi desidererà farne uso dovrà assumersi preventivamente la totale responsabilità di eventuali conseguenze negative derivanti dalla loro applicazione: come sono soliti indicare gli anglosassoni, *use it a your own risk*.

6.3.6 Casi d'uso

Allo scopo di provare *clonezilla* e fornirne al lettore un'impressione d'uso, sono stati realizzati quattro scenari di utilizzo, volutamente semplici:

¹² http://www.tldp.org/HOWTO/html_single/Partition/#volumelabels

¹³ http://downloads.sourceforge.net/project/clonezilla/clonezilla_live_doc/QuickReference_Card_0.9.4/ClonezillaLiveRefCard

1. copia (*backup*) di un disco rigido generando un *file immagine* su un'altra periferica;
2. ripristino (*restore*) del contenuto di un disco rigido utilizzando il *backup* precedentemente effettuato;
3. creazione di un CD/DVD-ROM contenente *clonezilla* e con al proprio interno il *file immagine* di un precedente *backup*;
4. ripristino (*restore*) del contenuto di un disco avviando il computer direttamente da un CD/DVD-ROM contenente l'immagine ISO di cui allo scenario precedente.

Si è ricorso, a tal fine, ad un ambiente di test virtualizzato così articolato:

- processore intel 386
- 512 MByte di RAM
- scheda grafica VGA
- due dischi rigidi collegati ad un *controller* IDE e denominati, secondo le convenzioni del kernel Linux:
 - `/dev/hda` (IDE, master) di dimensioni pari a 8 gigabyte con installato tramite *netinstall* l'ultimo rilascio di Debian GNU/Linux (*Lenny 2.6.26-2.686*) con una prima partizione (denominata, `/dev/hda1`) riservata a Debian GNU/Linux ed una seconda partizione (`/dev/hda2`) di *swap*: questa è la periferica destinata da essere clonata;
 - `/dev/hdb` (IDE, slave) di dimensioni pari a 10 GByte contenente un'unica partizione in formato *ext3* (denominata `/dev/hdb1`): questa è la periferica destinata a contenere l'immagine del *backup* del disco rigido `/dev/hda` generata dalla clonazione (nota: a scopo precauzionale, la dimensione di questo disco è stata scelta volutamente superiore a quella del precedente per prevenire criticità legate al suo eventuale riempimento in fase di clonazione).

È bene precisare che, allo scopo di predisporre le operazioni:

- prima di attivare *clonezilla* la periferica destinata a contenere il *backup* (`/dev/hdb`, nel nostro caso) deve essere già partizionata (la partizione è denominata `/dev/hdb1`, nel nostro caso) e formattata (il *file system* prescelto è *ext3*, nel nostro caso); inoltre,

il *file system* deve essere in grado di gestire *file* di grandi dimensioni (anche molti *gigabyte*) (nota: la gran parte dei dischi USB attualmente in commercio utilizzano un *file system* che limita la dimensione massima di un file a 2 GByte circa; qualora la periferica da clonare sia di dimensioni maggiori è molto probabile che si renda necessario installare sul disco USB un *file system* che non abbia tale limite; si rimanda alla documentazione dei produttori per gli elementi di dettaglio riguardo tale aspetto);

- il BIOS del computer deve essere configurato in modo da effettuare l'avvio (*boot*) da CD-ROM o, in alternativa, alla schermata di avvio del BIOS può essere attivata la funzione di selezione manuale della periferica di 'boot' (consultate la documentazione del vostro computer in merito a questo secondo punto).

Per ciascun caso d'uso, è stata eseguita una copia dello schermo del computer con *clonezilla* in esecuzione; le voci dei menu evidenziate (in colore rosso) sono quelle di volta in volta selezionate.

Caso 1: copia (backup) con file immagine

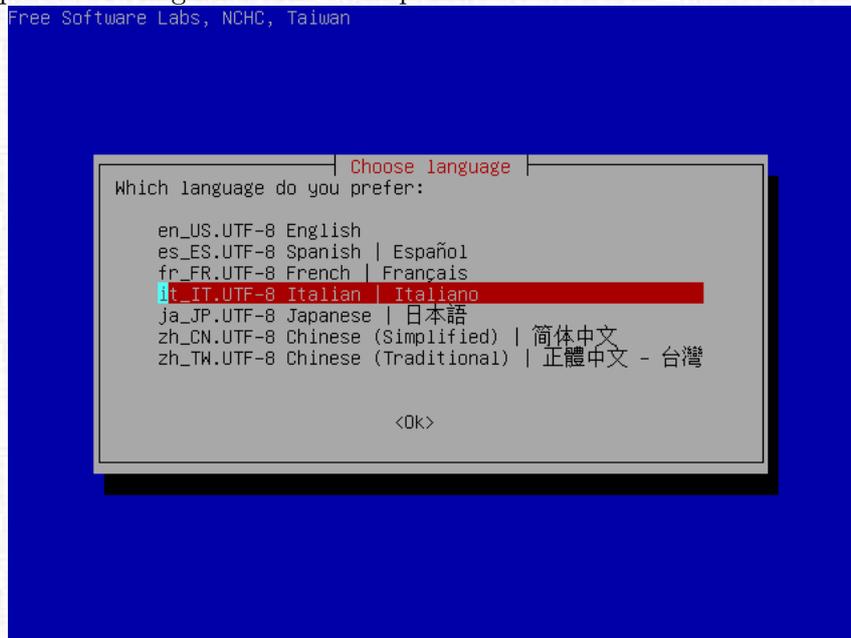
Schermata di avvio Questo menù appare a video dopo aver eseguito l'avvio del computer dal CD-ROM di *clonezilla*; nel nostro caso d'uso è stata scelta la terza voce dall'alto (**Clonezilla live (Default settings, VGA 640x480)**) ed è stato premuto il tasto invio.



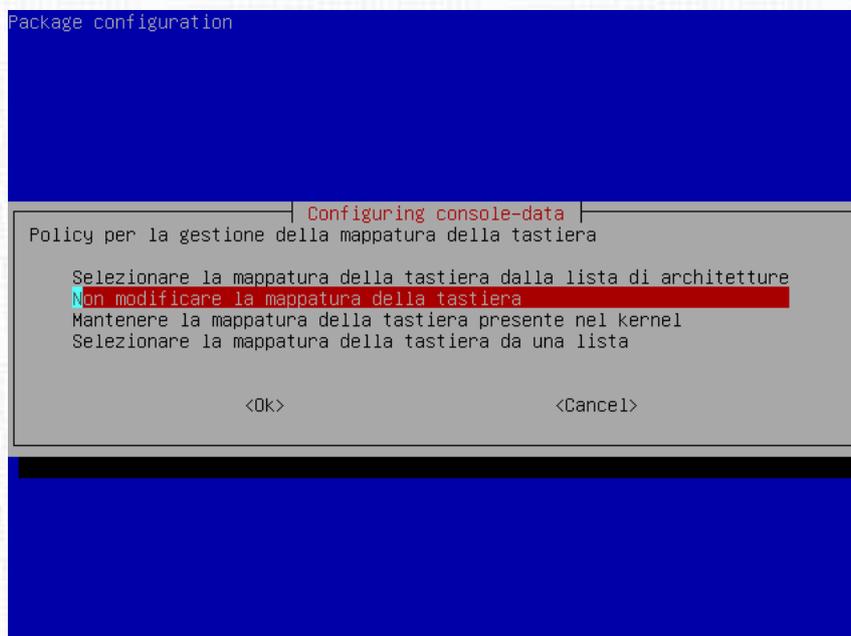
Avvio del kernel Questa schermata mostra l'avvio del sistema GNU/Linux con in corso il riconoscimento dei dischi rigidi presenti all'interno del computer.

```
[ 3.045581] usb usb1: SerialNumber: 0000:00:06.0
[ 3.047314] usb usb1: configuration #1 chosen from 1 choice
[ 3.048130] hub 1-0:1.0: USB hub found
[ 3.048976] hub 1-0:1.0: 8 ports detected
[ 3.136693] ehci_hcd 0000:00:0b.0: PCI INT A -> Link[LNKC] -> GSI 11 (level,
low) -> IRQ 11
[ 3.139062] ehci_hcd 0000:00:0b.0: EHCI Host Controller
[ 3.139867] ehci_hcd 0000:00:0b.0: new USB bus registered, assigned bus numbe
r 2
[ 3.141278] ehci_hcd 0000:00:0b.0: irq 11, io mem 0xf0805000
[ 3.153283] ehci_hcd 0000:00:0b.0: USB 2.0 started, EHCI 1.00
[ 3.155467] usb usb2: New USB device found, idVendor=146b, idProduct=0002
[ 3.155995] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=
1
[ 3.160134] usb usb2: Product: EHCI Host Controller
[ 3.160894] usb usb2: Manufacturer: Linux 2.6.30-backports.1-486 ehci_hcd
[ 3.161648] usb usb2: SerialNumber: 0000:00:0b.0
[ 3.162969] usb usb2: configuration #1 chosen from 1 choice
[ 3.163803] hub 2-0:1.0: USB hub found
[ 3.163996] hub 2-0:1.0: 8 ports detected
[ 3.165054] piix 0000:00:01.1: IDE controller (0x8086:0x7111 rev 0x01)
[ 3.165887] piix 0000:00:01.1: not 100% native mode: will probe irqs later
[ 3.166637] ide0: BM-DMA at 0xc000-0xc007
[ 3.167345] ide1: BM-DMA at 0xc008-0xc00f
[ 3.456180] hda: VBOX HARDDISK, ATA DISK drive
[ 3.792096] hdb: VBOX HARDDISK, ATA DISK drive
[ 3.793517] hda: UDMA/33 mode selected
[ 3.794260] hdb: UDMA/33 mode selected
[ 4.528141] hdc: VBOX CD-ROM, ATAPI CD/DVD-ROM drive
```

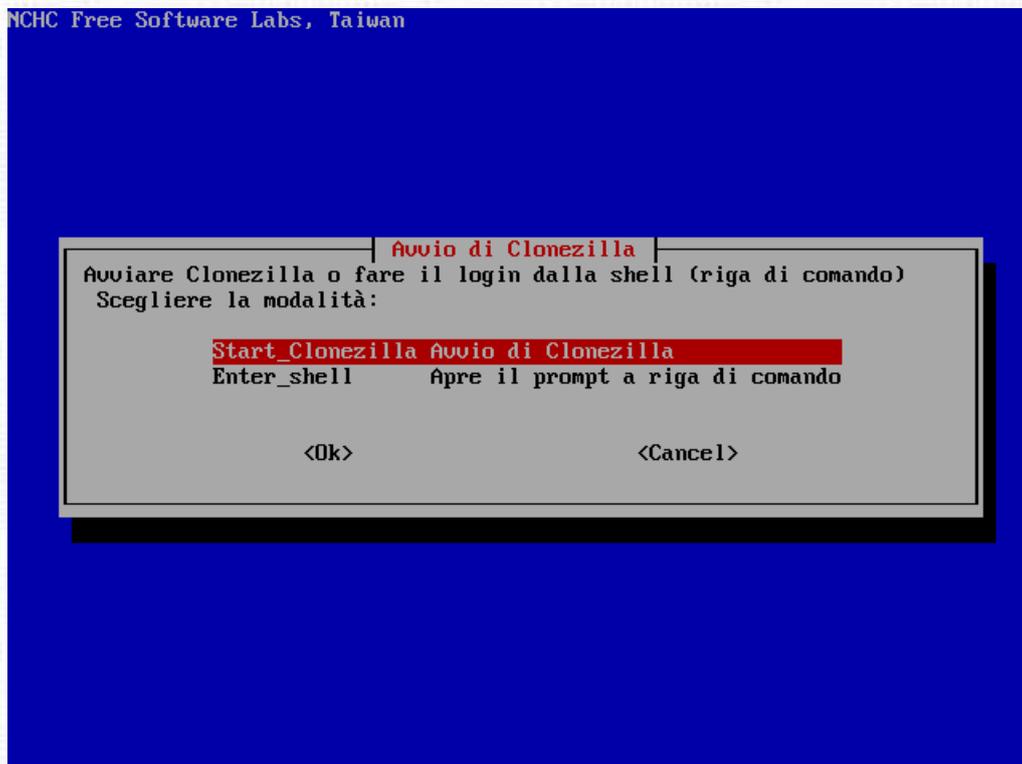
Scelta della lingua Questo menù permette la selezione della lingua utilizzata per i menu e i messaggi presentati l'utente; la traduzione italiana, per quanto abbastanza completa, può essere migliorata in alcuni punti.



Configurazione della tastiera Questo menù permette la selezione della configurazione della tastiera; la configurazione di *default* è quella americana (mantenuta in questo caso d'uso), ma è possibile personalizzare tale configurazione tramite la quarta voce del menu.

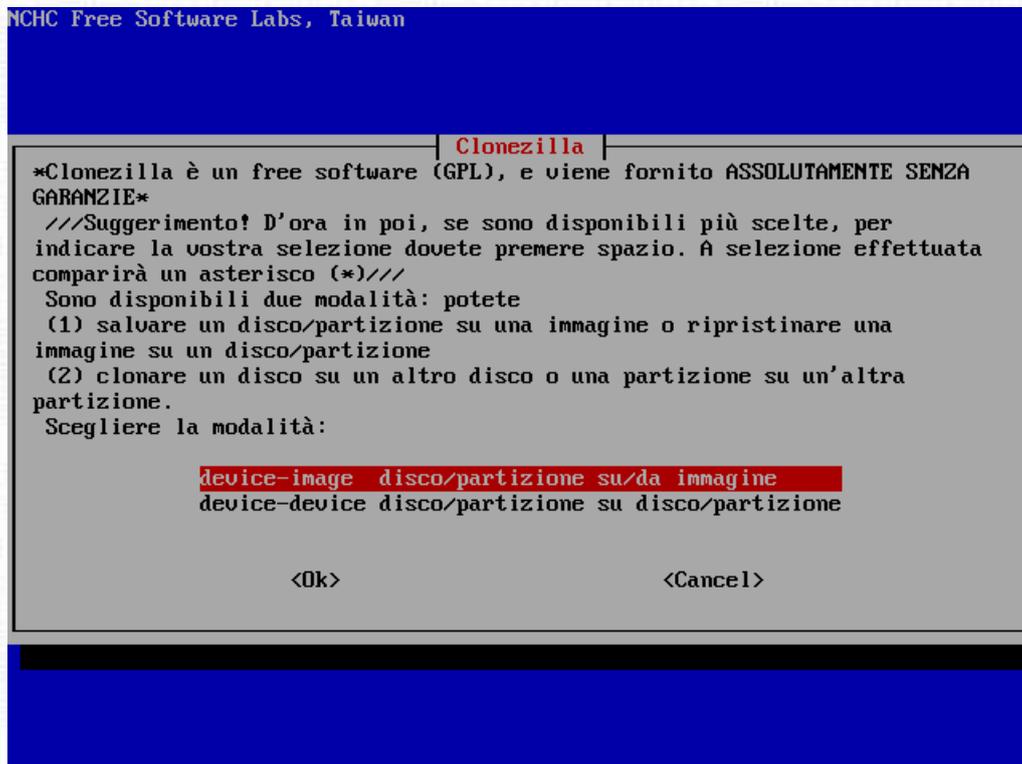


Avvio di *clonezilla* Questo menù permette di avviare *clonezilla* oppure di accedere ad una finestra di terminale di GNU/Linux; non dobbiamo, infatti, dimenticare che *clonezilla* è a tutti gli effetti un GNU/Linux Live CD anche se è ottimizzato per i compiti che deve svolgere e, quindi, ha installato un set di comandi limitato a quanto necessario. Inoltre, l'accesso al terminale permette, agli utenti esperti nell'utilizzo degli *script* alla base di *clonezilla*, di attivarne specifiche funzionalità o avviare manualmente i comandi che sono composti, attraverso i successivi passi, dal *wizard*.



Scelta modalità di clonazione Questo menù permette la scelta della modalità di clonazione:

- nel caso della prima scelta (*disco/partizione su/da immagine*), *clonezilla* creerà una *directory* che utilizzerà successivamente per generarvi all'interno un *file immagine* contenente i dati relativi al disco rigido o alla partizione che si intende clonare; successivamente, il contenuto di tale *directory* potrà essere utilizzato per il ripristino tutte le volte che lo si desidererà;
- nel caso della seconda scelta (*disco/partizione su disco/partizione*), *clonezilla* clonerà il contenuto del primo disco (o sua partizione) in un secondo disco (o sua partizione) rendendo da subito quest'ultimo immediatamente utilizzabile dal punto di vista *hardware*; questa seconda opzione, è maggiormente utile quando si desidera realizzare un disco rigido di *backup* di riserva già pronto ad essere installato in un computer quando, ad esempio, il disco rigido principale ha avuto un'avaria non recuperabile ed è indispensabile riavviare immediatamente il computer.

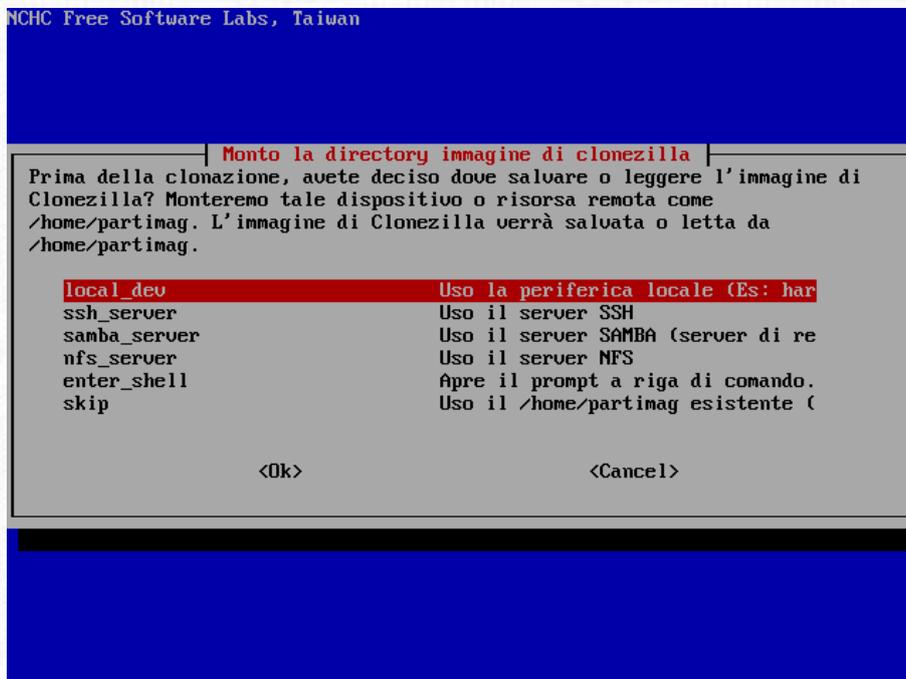


Scelta periferica di destinazione Questo menù permette la scelta della periferica nella quale sarà creata successivamente la *directory* che conterrà il *file immagine*: *clonezilla* avverte che la periferica sarà montata con la *path* `/home/partimag`.

L'opzione utilizzata per il presente scenario è la prima (*localdev*, cioè uso di una periferica fisicamente collegata al computer); ciò nonostante, il programma prevede anche di effettuare tale operazione con periferiche collegate ad altri computer in rete secondo i protocolli:

- *ssh* (*secure shell*)
- *smb* (*samba*)
- *nfs* (*network file system*)

È previsto, inoltre, che questa operazione possa essere eseguita manualmente da riga di comando (opzione *enter shell*) oppure che sia riutilizzata la scelta già adottata in una precedente sessione di clonazione alla quale non ha fatto seguito il riavvio del computer (*skip*).



Una volta eseguita la scelta della prima opzione (*localdev*), *clonezilla* entra in una fase di attesa per offrire l'opportunità di collegare dispositivi USB esterni e dare al kernel GNU/Linux il tempo di riconoscerli; sarà compito dell'utente interrompere questa fase di attesa (generalmente, sono sufficienti cinque secondi) premendo il tasto invio quando è presumibile che il riconoscimento della periferica USB sia stata completato con successo.

```
Prima della clonazione, avete deciso dove salvare o leggere l'immagine di
Clonezilla? Monteremo tale dispositivo o risorsa remota come
/home/partimag. L'immagine di Clonezilla verrà salvata o letta da
/home/partimag.

local_dev      Uso la periferica locale (Es: har
ssh_server     Uso il server SSH
samba_server   Uso il server SAMBA (server di re
nfs_server     Uso il server NFS
enter_shell    Apre il prompt a riga di comando.
skip           Uso il /home/partimag esistente (

                <Ok>                <Cancel>
```

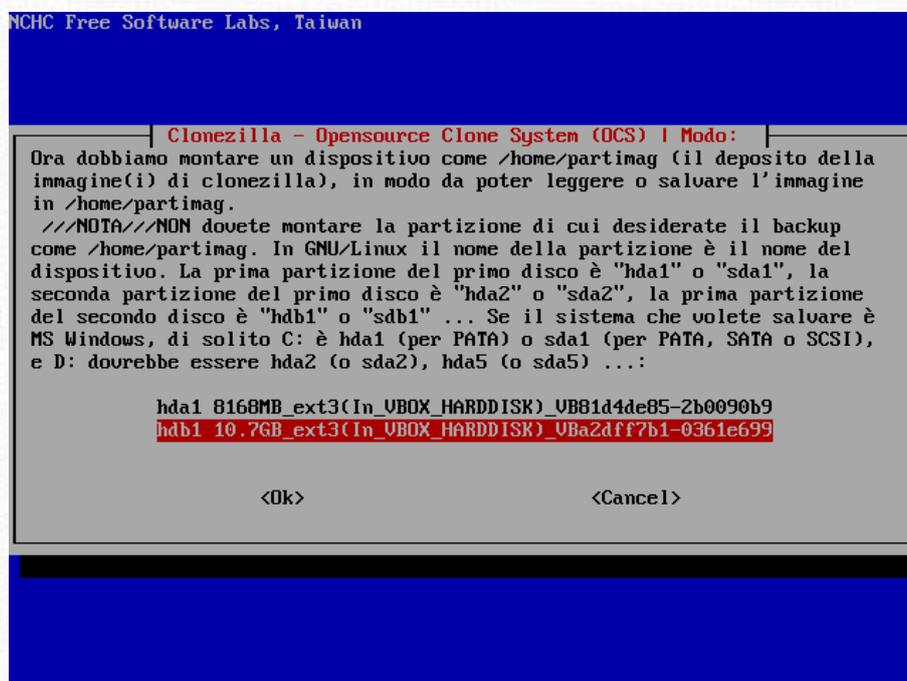
```
ocsroot device is local_dev
Preparing the mount point /home/partimag...
Se volete usare un dispositivo USB come repository delle immagini di Clonezilla,
vi preghiamo di inserire tale dispositivo USB in questa macchina *adesso*. Atte
ndete circa 5 secondi, quindi premete il tasto di Invio in modo che il sistema o
perativo sia in grado di rilevare il dispositivo USB e successivamente lo possa
montare come /home/partimag.
Premete "Invio" per continuare .....
```

Scelta partizione di destinazione Questo elenco di opzioni permette di scegliere la partizione, tra quelle delle periferiche riconosciute dal kernel nelle periferiche collegate al computer, da dedicare alla registrazione dei dati derivanti dalla clonazione. *Clonezilla*, per ogni partizione, presenta all'utente:

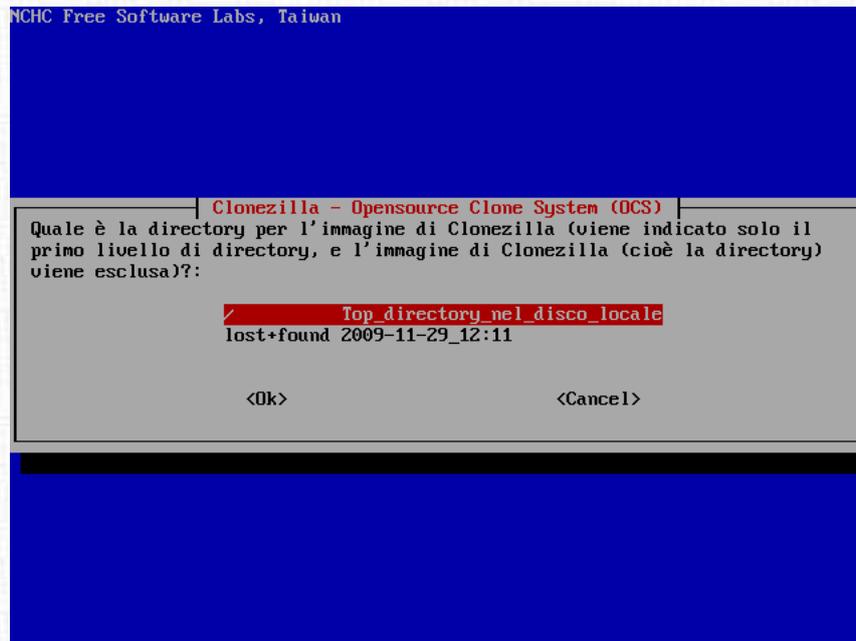
- il nome secondo la convenzione prevista dal kernel Linux
- la sua dimensione
- il tipo di *file system* utilizzato
- la corrispondente *label* e *UUID*.

Pertanto, nel nostro esempio, è stata selezionata la prima partizione del dispositivo `/dev/hdb`, di circa 10 gigabyte, formattata con il *file system ext3*, con *label* pari a `In_VBOX_HARDDISK` e *UUID* pari a `'VBa2ff7b1-0361e699'`.

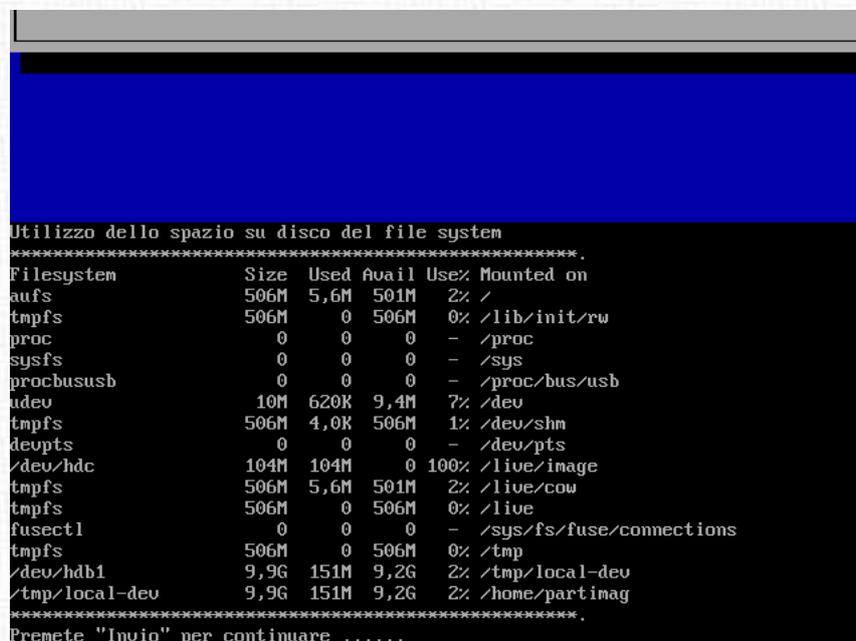
Come accennato nei precedenti paragrafi, si ricorda che il disco rigido di destinazione deve essere partizionato e formattato con un *file system* in grado di contenere *file* di dimensioni corrispondenti alle dimensioni almeno della periferica che si intende clonare.



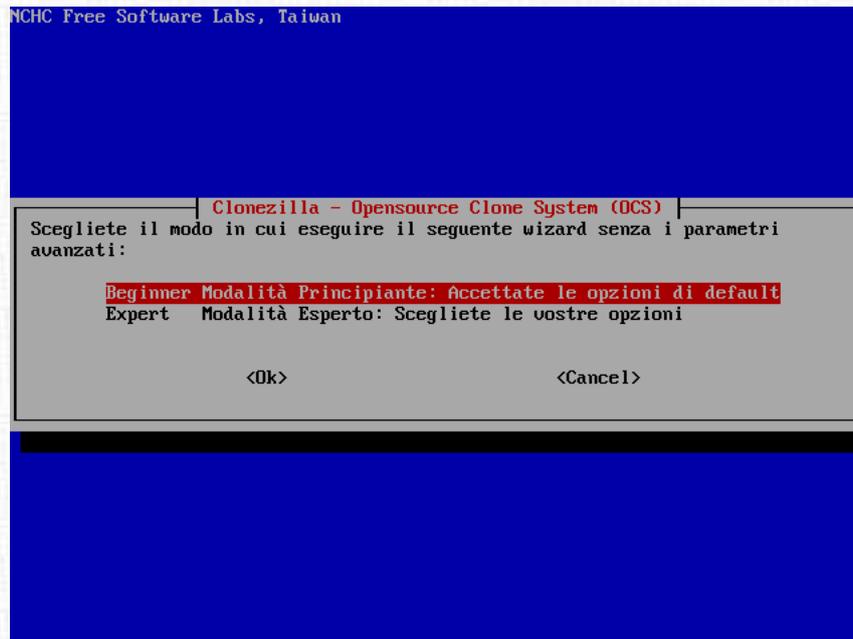
Scelta della *directory* nella partizione di destinazione Questo menù permette di selezionare la *directory* tra quelle presenti della partizione `/dev/hdb1` all'interno della quale successivamente creare la *directory* che conterrà il *file immagine*.



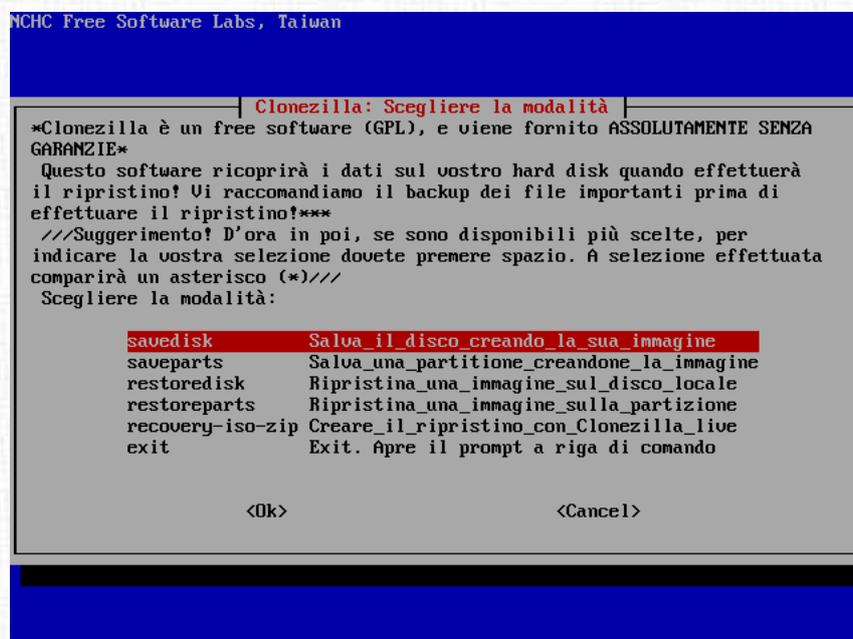
Stato delle partizioni montate da *clonezilla* Questa schermata vuole rappresentare, con finalità di controllo, lo stato delle partizioni attivate da *clonezilla* in previsione del *backup*: le ultime due sono quelle di nostro interesse e ci confermano che la partizione `/dev/hdb1` è stata montata.



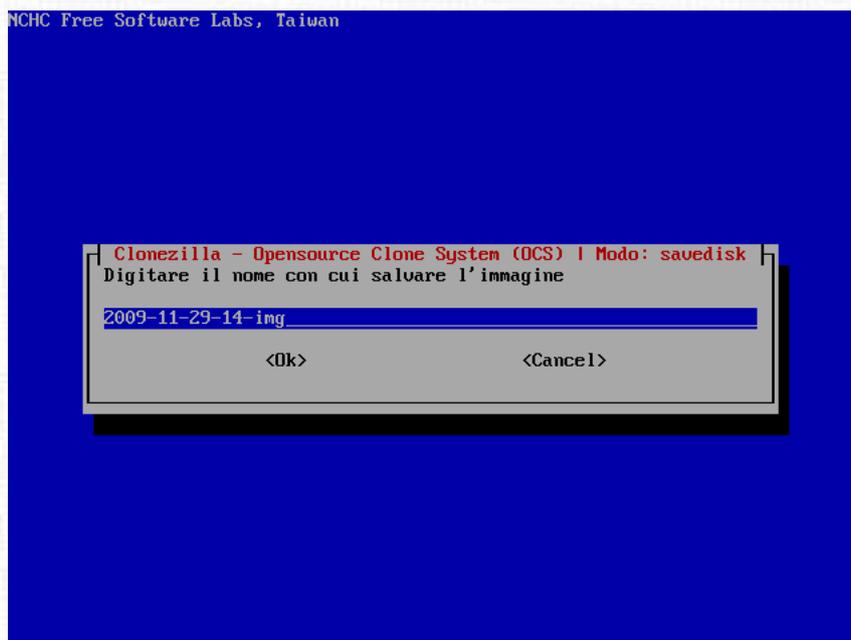
Scelta modalità di funzionamento (*beginner / expert*) Questo menù permette di scegliere se utilizzare i parametri preimpostati da *clonezilla* oppure eseguire una configurazione personalizzata: la nostra scelta ricade sulla prima opzione.



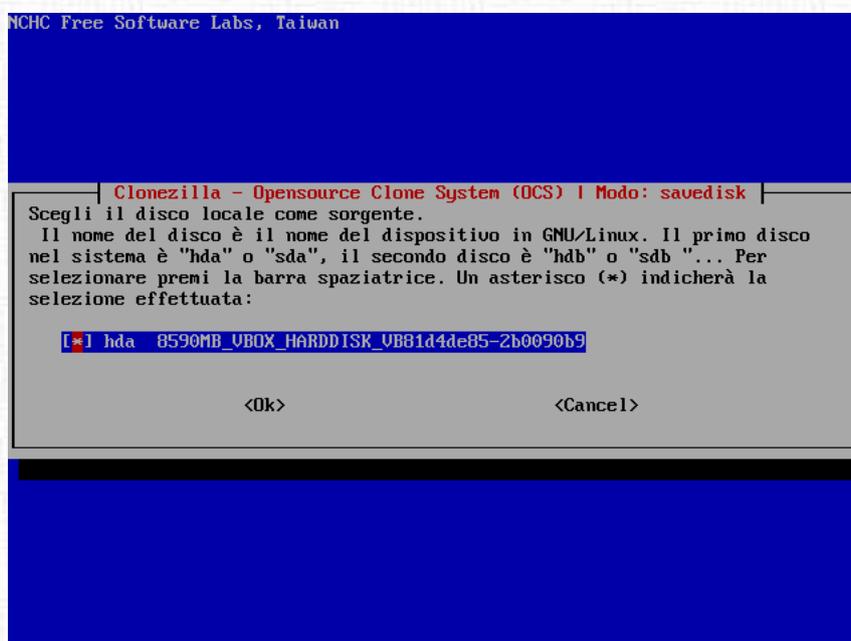
Scelta attività desiderata (*savedisk / restoredisk*) Questo elenco di opzioni permette di selezionare l'attività desiderata: l'opzione **savedisk** corrisponde alla clonazione di un disco (che sarà identificato tramite i passaggi successivi) nella partizione precedentemente specificata.



Selezione nome *file immagine* Questa schermata permette di specificare il nome della *directory* in cui saranno registrati i dati derivanti dalla clonazione e che potranno essere utilizzati per il successivo ripristino; *clonezilla* propone di default un nome derivante dalla data in cui la clonazione è eseguita.



Selezione disco da copiare Questo elenco di opzioni permette di indicare il disco rigido che si desidera clonare (`/dev/hda` nel nostro caso); qualora fossero stati riconosciuti più dischi rigidi, è possibile scegliere quelli desiderati.



Informativa riga di comando Questa schermata ha finalità informativa ed indica esplicitamente lo *script* che *clonezilla* sta per avviare con tutte le opzioni previste; questo può essere utile qualora si desiderasse ripetere in futuro e manualmente la procedura di *backup*.

```

Il nome del disco è il nome del dispositivo in GNU/Linux. Il primo disco
nel sistema è "hda" o "sda", il secondo disco è "hdb" o "sdb"... Per
selezionare premi la barra spaziatrice. Un asterisco (*) indicherà la
selezione effettuata:

[*] hda 8590MB_VBOX_HARDDISK_VB81d4de85-2b0090b9

<Ok> <Cancel>

```

```

Selected device [hda] found!
The selected devices: hda
Shutting down the Logical Volume Manager
Finished Shutting down the Logical Volume Manager
*****
PS. La prossima volta potete eseguire direttamente questo comando:
/opt/drbl/sbin/ocs-sr -q2 -c -j2 -z1 -i 2000 -p true savedisk "2009-11-29-14-img
" "hda"
Questo comando viene salvato con questo nome di file per un uso successivo, se n
ecessario: /tmp/ocs-2009-11-29-14-img-2009-11-29-14-36
Premete "Invio" per continuare ...

```

Conferma backup Questa schermata riepiloga i dati relativi alla periferica da clonare (`/dev/hda`), elenca le partizioni in essa contenute (solo una `/dev/hda1` nel nostro caso) ed indica la *directory* in cui saranno salvati i dati di *backup* (`/home/partimage/2009-11-29-14-img`)

```

Questo comando viene salvato con questo nome di file per un uso successivo, se n
ecessario: /tmp/ocs-2009-11-29-14-img-2009-11-29-14-36
Premete "Invio" per continuare ...
Activating the partition info in /proc... done!
Selected device [hda] found!
The selected devices: hda
Searching for data partition(s)...
Excluding busy partition or disk...
Unmouted partitions (including extended or swap): hda1 hda2
Collecting info... done!
Searching for swap partition(s)...
Excluding busy partition or disk...
Unmouted partitions (including extended or swap): hda1 hda2
Collecting info... done!
The data partition to be saved: hda1
The swap partition to be saved: hda2
Activating the partition info in /proc... done!
Selected device [hda1] found!
The selected devices: hda1
Getting /dev/hda1 info...
*****
Il passo successivo è quello di salvare l'harddisk/partizione di questa macchina
come immagine:
*****
Machine: VirtualBox
hda (8590MB_VBOX_HARDDISK_VB81d4de85-2b0090b9)
hda1 (8168MB_ext3(In_VBOX_HARDDISK_VB81d4de85-2b0090b9)
*****
-> "/home/partimag/2009-11-29-14-img".
Siete sicuri di voler continuare? ? (y/n)

```

Attività di *backup* in progress Queste schermate rappresentano quanto compare durante le operazioni di *backup*: può essere utile prenderne visione per comprendere cosa *clonezilla* mette in pratica dietro le quinte. Al termine del *backup*, saranno proposte tre opzioni: lo spegnimento del computer, il suo riavvio oppure il ritorno al menù di avvio della clonazione.

```

-> "/home/partimag/2009-11-29-14-img".
Siete sicuri di voler continuare? ? (y/n) y
OK, è partito!!
*****
Try to turn on the harddisk "/dev/hda" DMA...
*****
Shutting down the Logical Volume Manager
Finished Shutting down the Logical Volume Manager
Checking the integrity of partition table in the disk /dev/hda...
Reading the partition table for /dev/hda...RETVAIL=0
*****
The first partition of disk /dev/hda starts at 63.
Saving the hidden data between MBR (1st sector, i.e. 512 bytes) and 1st partition, which might be useful for some recovery tool, by:
dd if=/dev/hda of=/home/partimag/2009-11-29-14-img/hda-hidden-data-after-mbr skip=1 bs=512 count=62
62+0 records in
62+0 records out
31744 bytes (32 kB) copied, 0,00457977 s, 6,9 MB/s
*****
done!
Saving the MBR data for hda...
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0,00152179 s, 336 kB/s
*****
Starting saving /dev/hda1 as /home/partimag/2009-11-29-14-img/hda1.XXX...
/dev/hda1 filesystem: ext3.
Checking the disk space...
*****
Use partclone with gzip to save the image.
Image file will be split with size limit 2000 MB.
*****
If this action fails or hangs, check:
* Is the disk full ?
*****
Partclone v0.1.9 (Rev:323M) http://partclone.org
Starting to clone device (/dev/hda1) to image (-)
Reading Super Block
Calculating bitmap...
Elapsed: 00:00:01, Remaining: 00:00:00, Completed:100.00%, Rate: 119.64MB/min,
Total Time: 00:00:01, Ave. Rate: 0.1MB/min, 100.00%% completed!
File system: EXTFS
Device size: 8168 MB
Space in use: 4310 MB
Block size: 4096 Byte
Used block count: 1052095
Elapsed: 00:02:44, Remaining: 00:03:53, Completed: 41.25%, Rate: 650.36MB/min,
Elapsed: 00:02:47, Remaining: 00:03:52, Completed: 41.82%, Rate: 647.51MB/min,
Elapsed: 00:07:01, Remaining: 00:00:00, Completed: 99.99%, Rate: 614.11MB/min,
Total Time: 00:07:01, Ave. Rate: 0.6MB/min, 100.00%% completed!
Syncing... OK!
Partclone successfully cloned the device (/dev/hda1) to the image (-)
Checking the disk space...
>>> Time elapsed: 431.94 secs (~ 7.199 mins), average speed: 594.0 MB/min
*****
Finished saving /dev/hda1 as /home/partimag/2009-11-29-14-img/hda1.ext3-ptcl-img.gz
*****
Saving hardware info by lshw...
Saving DMI info...
Saving package info...

```

```

*****.
Saving swap partition hda2 info in /home/partimag/2009-11-29-14-img/swappt-hda2.
info if it exists...
Saving swap /dev/hda2 info in /home/partimag/2009-11-29-14-img/swappt-hda2.info.
..
Warning! UUID and LABEL of swap dev /dev/hda2 are nothing!
*****.
*****.
*****.
This program is not started by Clonezilla server, so skip notifying it the job i
s done.
Finished!
Now syncing - flush filesystem buffers...

*****.
Se volete usare di nuovo clonezilla:
(1) Rimanete in questa console (console 1), inserite un prompt a riga di comando
(2) Digitate il comando "exit" o "logout"
*****.
Quando tutto è stato fatto, ricordatevi di usare 'poweroff', 'reboot' o di esegui
re il menu che consente di fare un normale spegnimento/riavvio della procedura.
Altrimenti se il supporto di boot che state usando è un dispositivo riscrivibil
e (come una flash drive USB) ed è montato, l'anormale procedura di spegnimento/r
iavvio potrebbe causare il mancato avvio della volta successiva!
*****.
Premete "Invio" per continuare ...

```

Risultati Al fine di comprendere in che modo è stato eseguito il *backup* può essere utile ispezionare il contenuto del disco rigido nel quale *clonezilla* ha registrato il *file immagine*. In particolare, esplorando la periferica di destinazione risulta:

```

debian:~/hdb1$ ls -l
totale 20
drwxr-xr-x 2 root root 4096 29 nov 2009 2009-11-29-14-img
drwx----- 2 root root 16384 29 nov 13:11 lost+found

```

entrando nella *directory* di *backup* denominata 2009-11-29-14-img risulta:

```

debian:~/hdb1/2009-11-29-14-img$ ls -lh
totale 1,7G
-rw-r--r-- 1 root root 4 29 nov 15:45 disk
-rw----- 1 root root 1,7G 29 nov 15:45 hda1.ext3-ptcl-img.gz.aa
-rw-r--r-- 1 root root 36 29 nov 15:38 hda-chs.sf

```

```

-rw-r--r-- 1 root root 31K 29 nov 15:38 hda-hidden-data-after-mbr
-rw-r--r-- 1 root root 512 29 nov 15:38 hda-mbr
-rw-r--r-- 1 root root 326 29 nov 15:38 hda-pt.parted
-rw-r--r-- 1 root root 259 29 nov 15:38 hda-pt.sf
-rw-r--r-- 1 root root 907 29 nov 15:45 Info-dmi.txt
-rw-r--r-- 1 root root 11K 29 nov 15:45 Info-lshw.txt
-rw-r--r-- 1 root root 285 29 nov 15:45 Info-packages.txt
-rw-r--r-- 1 root root 5 29 nov 15:45 parts
-rw-r--r-- 1 root root 17 29 nov 15:45 swappt-hda2.info

```

- il file `disk` contiene il nome del *device* corrispondente alla periferica sottoposta a clonazione secondo la convenzione adottata dal kernel Linux:

```

debian:~/hdb1/2009-11-29-14-img$ cat disk
hda

```

- il file `hda1.ext3-ptcl-img.gz.aa` costituisce il vero e proprio file di *backup*; è da osservare che:
 - la dimensione del file (1,7 gigabyte) è di molto inferiore alla dimensione del disco rigido clonato (circa 10 GByte) poiché *clonezilla* ha copiato unicamente le aree delle partizioni effettivamente occupate da dati;
 - i permessi sul file sono stati impostati per ridurre al minimo il rischio di modifiche e cancellazioni inavvertite;
- i file `hda-chs.sf`, `hda-hidden-data-after-mbr`, `hda-mbr` contengono della periferica `/dev/hda` sottoposta a *backup*, rispettivamente, i dati relativi alla geometria (cilindri, testine e settori), i dati contenuti nel disco tra fine del *master boot record* (MBR) e inizio della prima partizione ed, infine, il contenuto del *master boot record* medesimo; questi dati saranno utilizzati in fase di ripristino per replicare esattamente il contenuto del disco rigido originale.

Volendo, a scopo esemplificativo, riportare il contenuto di `hda-chs.sf` esso è:

```

debian:~/hdb1/2009-11-29-14-img$ cat hda-chs.sf
cylinders=16644
heads=16
sectors=63

```

- i file *hda-pt.parted* e *hda-pt.sf* contengono i dati di dettaglio relativi alle partizioni del disco rigido sottoposto a clonazione:

```

debian:~/hdb1/2009-11-29-14-img$ cat hda-pt.parted
Model: VBOX HARDDISK (ide)
Disk /dev/hda: 16777216s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

```

Number	Start	End	Size	Type	File system	Flags
1	63s	15952607s	15952545s	primary	ext3	boot
2	15952608s	16772111s	819504s	primary	linux-swap	

```

debian:~/hdb1/2009-11-29-14-img$ cat hda-pt.sf
# partition table of /dev/hda
unit: sectors
/dev/hda1 : start=      63, size= 15952545, Id=83, bootable
/dev/hda2 : start= 15952608, size=   819504, Id=82
/dev/hda3 : start=      0, size=      0, Id= 0
/dev/hda4 : start=      0, size=      0, Id= 0

```

- i file *Info-dmi.txt* e *Info-lshw.txt* contengono informazioni relative all'hardware di sistema riconosciuto da *clonezilla* durante il *backup* usando, rispettivamente, i programmi *Dmidecode* e *lshw*.
- il file *Info-packages.txt* specifica, a scopo di referenza futura, i programmi utilizzati per eseguire il *backup*:

```

debian:~/hdb1/2009-11-29-14-img$ cat Info-packages.txt
Image was saved by these Clonezilla-related packages:
drbl-1.9.4-66 clonezilla-2.3.4-2 mkswap-uuid-0.1.1-1 drbl-partimage-0.6.7-1drbl

```

drbl-ntfsprogs-2.0.0-4 partclone-0.1.9-1 drbl-chntpw-0.0.20040818-7 drbl-lzop-1.02-0.8drbl
pigz-2.1.5-1drbl pbzip2-1.0.5-1drbl udpcast-20081213-1drbl

- il file `swappt-hda2.info` contiene informazioni (UIDe LABEL) sulla partizione di *swap* contenuta nel disco sottoposto a *backup*.

Caso 2: ripristino (*restore*) da *file immagine*

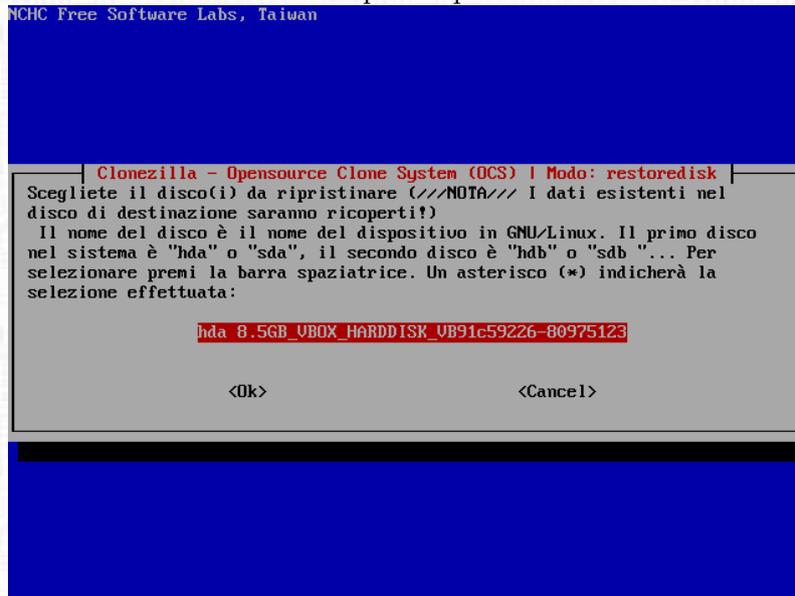
Per provare questo caso d'uso, è stata utilizzato in ambiente virtualizzato un secondo computer con le seguenti caratteristiche:

- processore intel 386
- 256 MByte di RAM
- scheda grafica VGA
- due dischi rigidi collegati ad un *controller* IDE e denominati, secondo le convenzione del kernel GNU/Linux:
 - `/dev/hda` (IDE, master) di dimensioni pari a 8 gigabyte non partizionato, né formattato, il cui contenuto è destinato ad essere ripristinato a partire dai dati della clonazione precedentemente eseguita;
 - `/dev/hdb` (IDE, slave) di dimensioni pari a 10 gigabyte che contiene il risultato della clonazione eseguita durante la procedura di *backup* del caso precedente (caso 1).

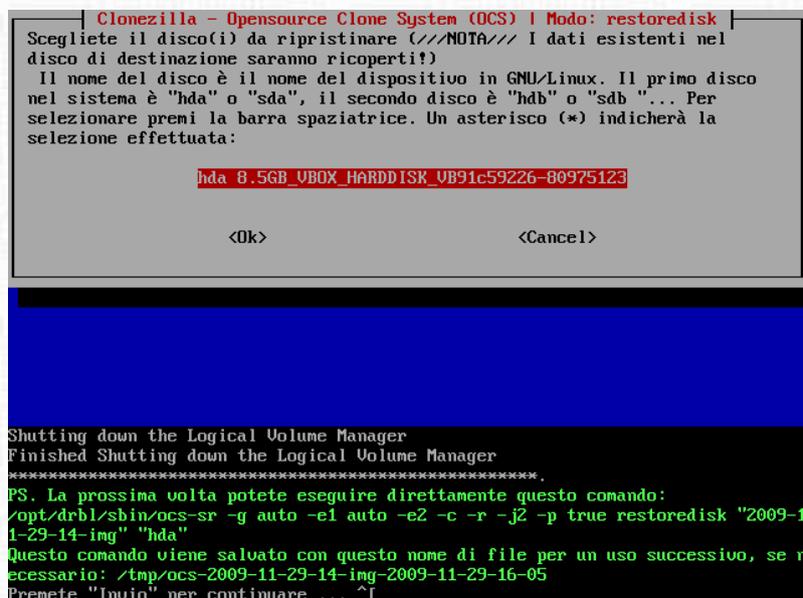
Sono, quindi, stati compiuti tutti i passaggi già esposti per il precedente caso d'uso fino alla selezione dell'attività desiderata (salvataggio / ripristino); in tal caso, a differenza di quanto precedentemente esposto, è stato scelto di effettuare l'attività **restoredisk**.

Scelta del disco rigido da ripristinare Questo elenco di opzioni permette di specificare quale disco rigido si desidera ripristinare (`/dev/hda` nel nostro caso); qualora fossero stati riconosciuti più dischi rigidi, è possibile scegliere quello desiderato. Nella schermata sotto riportata il termine ricoperti deve intendersi sovrascritti cioè **cancellati**;

ATTENZIONE! - con questo messaggio il programma vuole sottolineare il fatto che, a seguito del ripristino, tutti i dati eventualmente presenti nel disco rigido `/dev/hda` prima del ripristino andranno irreversibilmente persi: quindi attenzione!!



Informativa riga di comando Questa schermata ha finalità informativa ed indica esplicitamente lo *script* che *clonezilla* sta per avviare con tutte le opzioni previste; questo può essere utile qualora si desiderasse ripetere manualmente la procedura di ripristino in futuro.



Conferma ripristino *clonezilla* chiede per due volte la conferma del nulla osta al ripristino, stante la particolare delicatezza di tale operazione; come precedentemente ricordato:

ATTENZIONE! - qualora effettuata sulla periferica errata, tale operazione determina la cancellazione irreversibile dei dati in essa presenti

```

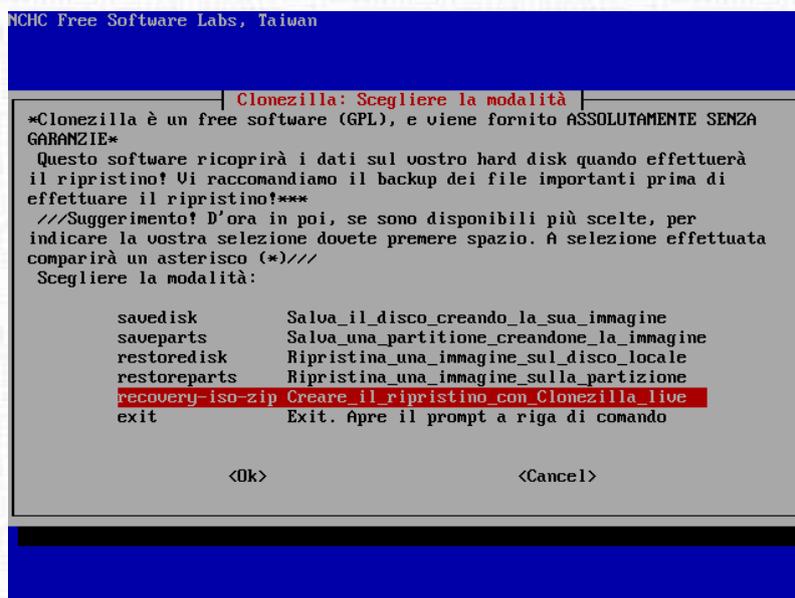
Shutting down the Logical Volume Manager
Finished Shutting down the Logical Volume Manager
*****
PS. La prossima volta potete eseguire direttamente questo comando:
/opt/drbl/sbin/ocs-sr -g auto -e1 auto -e2 -c -r -j2 -p true restoredisk "2009-1
1-29-14-img" "hda"
Questo comando viene salvato con questo nome di file per un uso successivo, se n
ecessario: /tmp/ocs-2009-11-29-14-img-2009-11-29-16-05
Premete "Invio" per continuare ... ^[
*****
Try to turn on the harddisk "/dev/hda" DMA...
*****
Activating the partition info in /proc... done!
*****
Il passo successivo è quello di ripristinare l'immagine nell'harddisk/partizione
di questa macchina: "/home/partimag/2009-11-29-14-img" -> "hda hda1"
ATTENZIONE!!! ATTENZIONE!!! ATTENZIONE!!!
ATTENZIONE! I DATI ESISTENTI IN QUESTO HARDDISK/PARTIZIONE(1), VERRANNO RICOPERT
I! TUTTI I DATI ESISTENTI ANDRANNO PERSI:
*****
Machine: VirtualBox
hda (8.5GB_VBOX_HARDDISK_VB91c59226-80975123)
*****
Siete sicuri di voler continuare? ?
[y/n] y
OK, è partito!!
This program is not started by clonezilla server.
Il passo successivo è quello di ripristinare l'immagine nell'harddisk/partizione
di questa macchina: "/home/partimag/2009-11-29-14-img" -> "hda (hda1)"
ATTENZIONE!!! ATTENZIONE!!! ATTENZIONE!!!
ATTENZIONE! I DATI ESISTENTI IN QUESTO HARDDISK/PARTIZIONE(1), VERRANNO RICOPERT
I! TUTTI I DATI ESISTENTI ANDRANNO PERSI:
*****
Machine: VirtualBox
hda (8.5GB_VBOX_HARDDISK_VB91c59226-80975123)
*****
Permettetemi di chiedervi ancora una volta, Siete sicuri di voler continuare? ?
[y/n] _

```

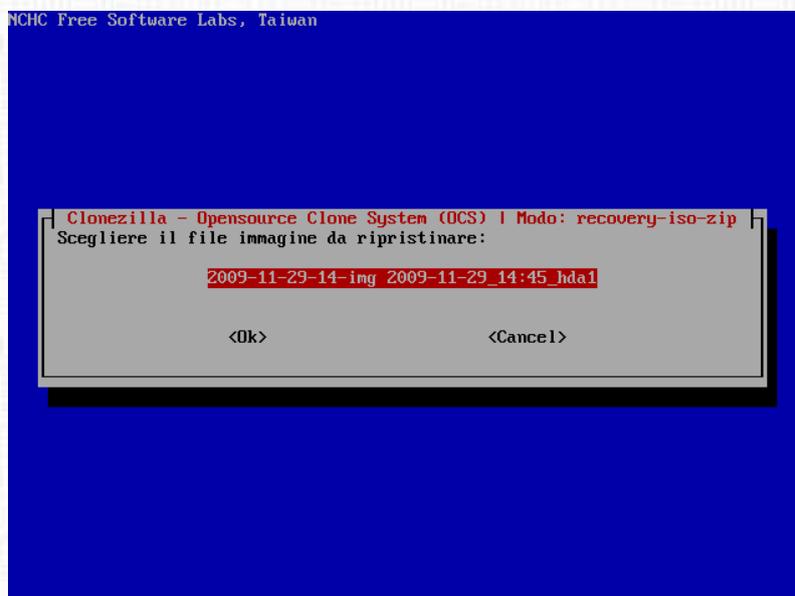
Caso 3: ripristino di un precedente *backup* da CD/DVD-ROM

Per provare questo caso d'uso, è stata utilizzato lo stesso computer del caso 1 ed il *file immagine* generato per il *backup* del disco `/dev/hda`.

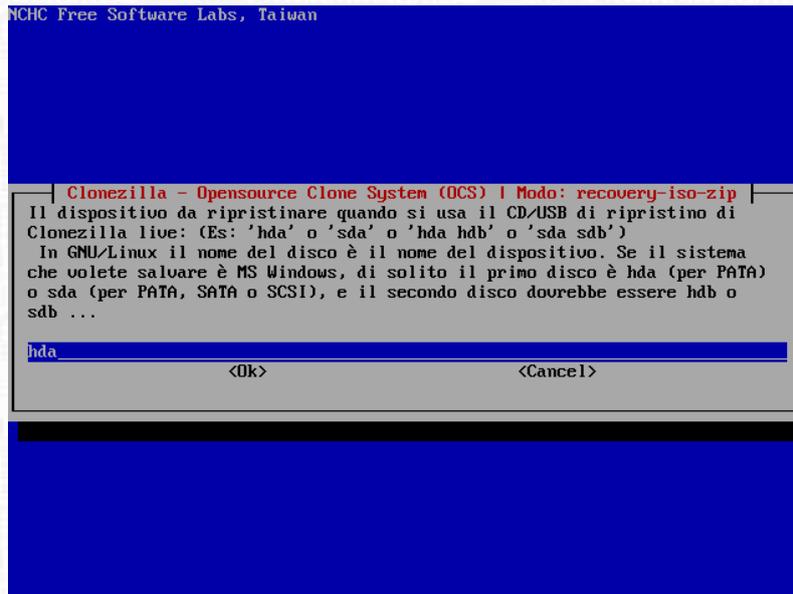
Sono, quindi, stati compiuti tutti i passaggi già esposti per il caso 1 fino alla selezione dell'attività desiderata; a questo punto, a differenza di quanto precedente esposto, è stato scelto di effettuare l'attività **recovery-iso-zip**.



Selezione nome *file immagine* Questo elenco permette di selezionare il *backup* precedentemente effettuato.



Scelta del disco rigido da ripristinare Questo elenco di opzioni permette di indicare il disco rigido che si desidera ripristinare in futuro (`/dev/hda` nel nostro caso); qualora fossero stati riconosciuti più dischi rigidi, è possibile scegliere quello desiderato, e, quindi, tramite passaggi successivi giungere a creare l'immagine ISO del CD/DVD-ROM da utilizzare per il ripristino.



Risultati Al fine di comprendere in che modo *clonezilla* ha eseguito il *backup* può essere utile ispezionare il contenuto del disco utilizzato. In particolare, esplorando il disco di destinazione risulta:

```
debian:~$ \textbf{ls -lh hdb1/}
totale 1,8G
drwxr-xr-x 2 root root 4,0K 29 nov 15:45 2009-11-29-14-img
-rw-r--r-- 1 root root 1,8G 29 nov 2009 clonezilla-live-2009-11-29-14-img.iso
drwx----- 2 root root 16K 29 nov 13:11 lost+found
```

dove il file `clonezilla-live-2009-11-29-14-img.iso` corrisponde all'immagine ISO di *clonezilla* all'interno della quale sono stati inseriti anche i dati del *backup*; stante le dimensioni dell'immagine ISO (> 650 o 700 MByte), essa dovrà essere necessariamente registrata

utilizzando un supporto per DVD-ROM (nota: in fase di creazione dell'immagine, *clonezilla* emette un avviso relativa alla dimensione dell'immagine qualora essa sia di dimensione superiori a quella contenibile da un supporto per CD-ROM).

Caso 4: ripristino automatico di un precedente *backup* da CD/DVD-ROM

Per eseguire il ripristino, è sufficiente avviare il computer da CD/DVD-ROM masterizzato utilizzando l'immagine ISO creata con il caso d'uso numero 3; dopo aver presentato la schermata di avvio, *clonezilla* chiederà di iniziare la procedura di ripristino dal CD/DVD-ROM che contiene anche il *file* di *backup* secondo i parametri precedentemente impostati nel caso d'uso numero 3.

6.3.7 Conclusioni

Nella pagine precedenti sono stati analizzati alcuni casi d'uso di *clonezilla*. In premessa sono state elencate le principali differenze tra le tecniche tradizionali di *backup* e quelle utilizzate nella clonazione dei dispositivi di memorizzazione.

Naturalmente, se è vero che la tecnica della clonazione ha alcuni aspetti positivi, è altrettanto vero che ha anche alcuni limiti di cui tener conto:

- *clonezilla* esegue una copia integrale del disco rigido e, quindi, se quest'ultimo contiene le configurazioni del sistema operativo, anch'esse saranno interamente replicate; se ciò può costituire un ulteriore vantaggio della tecnica, è anche un potenziale limite; infatti, qualora parte o tutta la configurazione *hardware* del computer in cui è eseguito il ripristino sia differente da quello del *backup*, il sistema operativo potrebbe fallire il riconoscimento o la configurazione delle periferiche; questa eventualità, pur mitigata dalla capacità del kernel GNU/Linux di adattarsi a *hardware* molto differenti, può richiedere un intervento manuale per configurare il sistema operativo;
- l'uso di *clonezilla* implica l'impossibilità di utilizzare le periferiche di memorizzazione durante le attività di *backup*: se questo è accettabile per sistemi destinati all'uso personale, non lo è quasi mai per sistemi di produzione che devono essere sempre operativi;
- per volumi di dati molto elevati (o architetture di archiviazione dati come LVM o RAID) *clonezilla* può non essere la soluzione ottimale soprattutto qualora il *file*

system non sia riconosciuto e, quindi, non sia possibile eseguire un *backup* ottimizzato: in tali casi, può essere più utile scegliere tecniche di *backup* che si avvalgano di periferiche di archiviazione specializzate per questo compito;

- *clonezilla* è uno strumento che richiede un rilevante bagaglio conoscitivo, soprattutto per evitare eventi catastrofici legati ad errori dell'utente in fase di ripristino, e quindi in prima battuta se ne ritiene sconsigliabile l'uso agli utenti alle prime armi.

Alla luce di quanto sopra indicato, sembrerebbe opportuno suggerire di utilizzare *clonezilla* come strumento di *backup* e *restore* di secondo livello rispetto a procedure convenzionali con medesima finalità oppure - qualora non esistano particolari esigenze di configurazione e riconoscimento *hardware* - come strumento per replicare velocemente installazioni già effettuate.

6.3.8 Riferimenti

- The National Center for High-Performance Computing, Hsinchu, Taiwan, 2009 ,
<http://www.nhc.org.tw/en/>
- Clonezilla Project, Home Page, 2009,
<http://sourceforge.net/projects/clonezilla>
- Clonezilla Project, Source Codes, 2009,
<http://clonezilla.org/download/sourceforge/src/sourcecodes.php>
- Partclone Project, Home Page, 2009,
<http://partclone.org/>
- Linux-NTFS project, ntfsclone, 2009,
<http://www.linux-ntfs.org/doku.php?id=ntfsclone>

- Partimage, 2009,
<http://www.partimage.org>
- Wikipedia, dd (Unix), 2009,
[http://it.wikipedia.org/wiki/Dd_\(Unix\)](http://it.wikipedia.org/wiki/Dd_(Unix))
- Anthony Lissot, Linux Partition HOWTO, 2005,
http://tldp.org/HOWTO/html_single/Partition/
- Clonezilla Live Quick Reference Card, 2009,
http://downloads.sourceforge.net/project/clonezilla/clonezilla_live_doc/QuickReference_Card_0.9.4/ClonezillaLiveRefCard_EN_0.9.4.pdf
- Torben Mathiasen, Linux Allocate Devices, 2005,
<http://www.kernel.org/pub/linux/docs/device-list/devices.txt>

6.4 Iproute2: il futuro del networking è già qui

6.4.1 Introduzione

Iproute2 è una suite di utility per la gestione avanzata delle configurazioni di rete e per il controllo del traffico TCP/IP in ambiente Linux.

Fa uso intensivo delle rtnetlink socket, moderna e potente interfaccia di configurazione dinamica dello stack di rete.

L'autore originale, Alexey Kuznetsov, è anche conosciuto per l'implementazione QoS nel kernel Linux.

Attualmente il mantainer del progetto è Stephen Hemminger.

Installata di default nelle maggiori distribuzioni, si trova a convivere con la suite net-tools i cui strumenti (ifconfig, route, etc.) sono ancora utilizzati negli scripts di inizializzazione delle interfacce, costituendone standard de facto sebbene risultino inadeguati nei moderni ambienti di rete.

In questo articolo si esploreranno le potenzialità di questa suite mettendola a confronto (quando possibile) con gli strumenti classici che tutti conosciamo.

6.4.2 Installazione

Come detto precedentemente, iproute2 risulta installata in molte distribuzioni moderne tra le quali, ovviamente, Debian.

Nonostante ciò, chi volesse effettuare l'installazione da sorgenti, dovrà scaricare il tarball dal repository git:

```
pmate:~$ git clone git://git.kernel.org/pub/scm/linux/kernel/git/shemminger/ \
                                     \
                                     iproute2.git
```

e, una volta spostatosi nella directory di scaricamento, procedere con i classici strumenti (avendo cura di specificare la variabile SBINDIR):

```
pmate:~$ ./configure
pmate:~$ make SBINDIR=/sbin
pmate:~$ su
password:
debian:~# make install SBINDIR=/sbin
```

6.4.3 Il comando IP

È il comando più importante e potente dell'intera suite. Sostituisce ed integra i più famosi ifconfig, route, etc.

Mostra e manipola il routing e le sue policy, le interfacce di rete e i tunnels.

La sua sintassi:

```
ip [ OPTIONS ] OBJECT { COMMAND | HELP }
```

Per una panoramica dei parametri OBJECTS e OPTIONS, fare riferimento alle pagine man di ip.

Testiamone il funzionamento.

Identificazione delle interfacce di rete

(Nei comandi seguenti si omette l'output dell'interfaccia di loopback per comodità di visualizzazione)

old-style:

```
debian:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:4c:36:07
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2098 (2.0 KiB)  TX bytes:10902 (10.6 KiB)
          Interrupt:11 Base address:0xd020
...
...
```

new-style:

```
debian:~# ip link show
...
...
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ether 08:00:27:4c:36:07 brd ff:ff:ff:ff:ff:ff
```

Assegnamo un indirizzo a quell'interfaccia:

```
debian:~# dhclient eth0
...
...
Listening on LPF/eth0/08:00:27:4c:36:07
Sending on   LPF/eth0/08:00:27:4c:36:07
Sending on   Socket/fallback
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 10.0.2.15 -- renewal in 38509 seconds.
```

e ripetiamo i comandi:

old-style:

```
debian:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:4c:36:07
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4c:3607/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2852 (2.7 KiB)  TX bytes:16978 (16.5 KiB)
          Interrupt:11 Base address:0xd020
...

```

new-style:

```
debian:~# ip link show
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN\
                                             qlen 1000
    link/ether 08:00:27:4c:36:07 brd ff:ff:ff:ff:ff:ff
```

La differenza che più salta all'occhio è che ifconfig fornisce informazioni riguardo l'indirizzo ip dell'interfaccia, al contrario di ip che invece si limita al mac address e allo stato del device:

```
eth0: <BROADCAST,MULTICAST> ...
```

quando l'interfaccia è down,

```
eth0: Link encap:Ethernet HWaddr 08:00:27:4c:36:07
      inet addr:192.168.1.30  Bcast:192.168.1.255
      Mask:255.255.255.0
```

...

```
<BROADCAST,MULTICAST,UP,LOWER_UP> ...
```

quando l'interfaccia è up.

Per avere informazioni riguardo ad indirizzi assegnati all'interfaccia di rete:

```
debian:~# ip address show dev eth0
```

```
7: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen\
                                             1000
```

```
    link/ether 08:00:27:4c:36:07 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.30/24 brd 192.168.1.255 scope global wlan0
    inet6 fe80::219:5bff:fe39:f576/64 scope link
        valid_lft forever preferred_lft forever
```

Assegnazione ip

Assegnamo un indirizzo ad eth0 e abilitiamola:

old-style:

```
debian:~# ifconfig 192.168.1.30 netmask 255.255.255.0 up
```

new-style:

```
debian:~# ip address add 192.168.1.30/24 broadcast 192.168.1.255 dev eth0
```

```
debian:~# ip set eth0 up
```

e assegnamo un alias:

old-style:

```
debian:~# ifconfig eth0:1 192.168.0.55
```

```
debian:~# ifconfig eth0
```

```
eth0 Link encap:Ethernet HWaddr 08:00:27:4c:36:07
      inet addr:192.168.1.30  Bcast:192.168.1.255
      Mask:255.255.255.0
```

new-style:

```
debian:~# ip address add 192.168.0.55/24 broadcast 192.168.0.255 dev eth0
```

```
debian:~# ip address show dev eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN\
                                             qlen 1000
    link/ether 08:00:27:4c:36:07 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.30/24 brd 192.168.1.255
    scope global eth0
    inet 192.168.0.55/24 brd 192.168.0.255
    scope global eth0
```

Si può facilmente notare come con ifconfig non siano visibili gli alias assegnati all'interfaccia, mentre il comando ip fornisce informazioni complete a riguardo. L'host, infatti, è raggiungibilissimo dal ping:

```
debian:~# ping -c1 192.168.1.30
PING 192.168.1.30 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=64 time=1.72 ms
```

```
debian:~# ping -c1 192.168.0.55
PING 192.168.1.30 56(84) bytes of data.
64 bytes from 192.168.0.55: icmp_seq=1 ttl=64 time=2.16 ms
```

e l'ARP table associa correttamente l'ip al mac address:

```
debian:~# arp -a -n | grep 192.168.
? (192.168.1.30) at 08:00:27:4c:36:07
[ether] on eth0
? (192.168.0.55) at 08:00:27:4c:36:07
[ether] on eth0
```

Per ovviare a ciò possiamo usare il parametro *label*:

```
debian:~# ip address add 192.168.0.55/24 label eth0:1 broadcast 192.168.0.255 dev \
```

```
debian:~# ip address show dev eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN\
                                             qlen 1000
```

```
link/ether 08:00:27:4c:36:07 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.30/24 brd 192.168.1.255 scope global eth0
inet 192.168.0.55/24 brd 192.168.0.255 scope global secondary eth0:1
```

```
debian:~# ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:4c:36:07
          inet addr:192.168.1.30   Bcast:192.168.1.255
          Mask:255.255.255.0
          ...

eth0:1    Link encap:Ethernet HWaddr 08:00:27:4c:36:07
          inet addr:192.168.0.55   Bcast:192.168.0.255
          Mask:255.255.255.0
          ...
```

6.4.4 Arp

La tabella di Arp contiene le informazioni relative all'abbinamento degli indirizzi IPV4 con i relativi MacAddress (chiamati anche, secondo il modello ISO/OSI, *indirizzi di livello due*).

La sua gestione può essere molto complessa. In questa sede, pertanto, si mostreranno solo degli esempi pratici riguardanti l'utilizzo del comando *arp*.

Per visualizzare la tabella ARP per quanto riguarda i dispositivi fisicamente collegati all'interfaccia eth0:

```
debian:~# ip neighbour show dev eth0
192.168.1.1 lladdr AA:BB:00:CC:DD:02 STALE
```

Per cancellare dalla tabella le voci riferite al nodo 192.168.1.70, per il collegamento relativo ad eth0:

```
debian:~# ip neighbour del 192.168.1.70 dev eth0
```

Per svuotare la tabella ARP di tutte le voci riferite all'interfaccia eth0:

```
debian:~# ip neighbour flush dev eth0
```

6.4.5 Routing

Quando si ha la necessità di incanalare il traffico dati verso determinati punti di snodo della rete (i gateway...), la suite net-tools mette a disposizione il comando `route`.

Per aggiungere una route statica ad una rete intera, ad esempio:

```
debian:~# route add -net 192.168.1.0 netmask 255.255.255.0 gw 10.0.0.254
```

Per definire, invece, il default gateway:

```
debian:~# route add default gw 192.168.1.1
```

Per rimuoverlo:

```
debian:~# route del default gw 192.168.1.1
```

Per visualizzare la tabella di routing:

```
debian:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.1.0      0.0.0.0         255.255.255.0  U        0      0        0 eth0
0.0.0.0          192.168.1.1    0.0.0.0        UG       0      0        0 eth0
```

(il parametro `*-n*` serve a velocizzare l'operazione evitando il reverse lookup degli IP).

Allo stesso modo, per eliminare il routing verso un determinato host via gateway 192.168.1.1:

```
debian:~# route delete -host 192.106.2.1 -gateway 192.168.1.1
```

o per eliminare un'intera route:

```
debian:~# route del -net 192.168.1.0
```

Il discorso si complica se sopravviene la necessità di gestire più tabelle di routing contemporaneamente...

In questo caso la suite net-tools diventa inadatta e ci si deve affidare ad `iproute2` che, sempre per mezzo del comando `ip` riesce a svolgere questo compito in maniera egregia.

I settaggi concernenti le tabelle di routing sono contenuti nel file `/etc/iproute2/rt_tables`:

```
debian:~# cat /etc/iproute2/rt_tables
# reserved values
#
255    local
254    main
253    default
0      unspec
#
# local
#
1      inr.ruhep
#
# Special table for source routing.
# See /etc/network/interfaces.
```

In questo caso è presente la sola tabella `main` (l'unica vecchia tabella di routing):

```
debian:~# ip route show table main
172.16.12.1 dev tun2  proto kernel  scope link  src 172.16.12.2
88.57.16.24/30 dev eth1  proto kernel  scope link  src 88.57.16.26
10.0.1.0/24 via 172.16.12.1 dev tun2
192.168.2.0/24 dev eth0  proto kernel  scope link  src 192.168.2.2
default via 88.57.16.25 dev eth1
```

All'interno del file `*/etc/iproute2/rt_tables*`, ogni tabella di routing è identificata da un numero compreso tra 1 e 255.

La main routing table, come abbiamo visto, è identificata sempre dal numero `*254*`.


```
debian:~# ip rule add from 10.11.11.0/24 table ISP1
debian:~# ip rule add from 10.12.12.0/24 table ISP1
debian:~#
debian:~# ip route add default via 10.8.8.1 dev eth1 table ISP2
debian:~# ip rule add from 10.22.22.0/24 table ISP2
debian:~# ip rule add from 10.33.33.0/24 table ISP2
```

Per visualizzare le singole tabelle:

```
debian:~# ip route show table ISP1
debian:~# ip route show table ISP2
debian:~# ip route show table main
```

Per visualizzare le regole di ogni singola tabella:

```
debian:~# ip rule show | grep ISP1
debian:~# ip rule show | grep ISP2
```

In questo modo i pacchetti provenienti dalle reti 10.11.11.0/24 e 10.12.12.0/24, saranno veicolati dalle regole contenute nella tabella ISP1 e quindi passate al gateway ISP1. Allo stesso modo, i pacchetti provenienti dalle reti 10.22.22.0/24 e 10.33.33.0/24, saranno veicolate dal gateway ISP2.

Tutti quei pacchetti, invece, provenienti da reti non contemplate dalle regole precedenti, saranno veicolati secondo quelle contenute nella *main routing table*.

6.4.6 Tunneling

Può capitare di avere l'esigenza che due nodi della rete scambino traffico incapsulato con un protocollo differente da IPV4 o che è indirizzato verso una lan privata i cui indirizzi IPV4, ovviamente, non sono validi su internet. In queste situazioni si utilizza una connessione virtuale tra i due nodi chiamata tunnel. Essi non funzionano quando attraversano il processo di NAT. In più, se c'è un firewall, questo dovrà essere opportunamente configurato per permettere il passaggio di questo particolare tipo di traffico.

Di solito si utilizza il tunnel per collegare due nodi IPV6 separati da una rete IPV4. Tra i due nodi si implementerà un tunnel che incapsulerà ogni pacchetto IPV6 in un pacchetto IPV4 di modo che le due entità possano essere interconnesse.

Creare un tunnel con iproute2 è davvero semplice:

```
debian:~# ip tunnel add nostrotunnel mode sit local 192.168.0.1 remote 192.168.1.99\
```

Si è creato il tunnel chiamato nostrotunnel sull'interfaccia *eth0* avente come ip *192.168.0.1* in cui l'altro estremo si trova all'indirizzo *192.168.1.99*.

Controlliamo la situazione:

```
debian:~# ip tunnel show
sit0: ipv6/ip remote any local any ttl 64 nopmtudisc
pippo: ipv6/ip remote 192.168.1.42 local 192.168.0.1 dev eth0 ttl inherit
```

```
debian:~# ip link show
1: lo: <loopback,up> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <broadcast,multicast,up> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:48:54:1b:25:30 brd ff:ff:ff:ff:ff:ff
4: sit0@none: <noarp> mtu 1480 qdisc noop
   link/sit 0.0.0.0 brd 0.0.0.0
11: pippo@eth0: <pointopoint,noarp> mtu 1480 qdisc noop
   link/sit 192.168.0.1 peer 192.168.1.42
```

Da notare che ora l'interfaccia è identificata dal nome pippo@eth0.

Se ci interessa, invece, incanalare traffico IPV4, invece di utilizzare l'opzione *mode sit*, utilizzeremo quella *mode gre*:

```
debian:~# ip tunnel add nostrotunnel mode gre local 192.168.0.1 remote 192.168.1.99\
                                                dev eth0
```

```
debian:~# ip tunnel show
gre0: gre/ip remote any local any ttl inherit nopmtudisc
nostrotunnel4: gre/ip remote 192.168.1.42 local 192.168.0.1 dev eth0 ttl inherit
```

```
debian:~# ip link show
1: lo: <loopback,up> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <broadcast,multicast,up> mtu 1500 qdisc pfifo_fast qlen 100
   link/ether 00:48:54:1b:25:30 brd ff:ff:ff:ff:ff:ff
4: gre0@none: <noarp> mtu 1480 qdisc noop
   link/sit 0.0.0.0 brd 0.0.0.0
11: nostrotunnel4@eth0: <pointopoint,noarp> mtu 1480 qdisc noop
   link/gre 192.168.0.1 peer 192.168.1.42
```

(GRE è un particolare tipo di tunnel supportato, ad esempio, dai router Cisco).

Tip

Quando si usano dei tunnel è importante fare attenzione al parametro TTL (Time To Live) per evitare di creare dei loop all'interno della rete.

Questo valore deve essere tenuto basso (si consiglia di impostarlo pari a 64):

```
debian:~# ip tunnel add nostrotunnel mode gre local 192.168.0.1 remote 192.168.1.99\  
tunnel ttl 64 dev eth0
```

Comandistica

Per attivare l'interfaccia *nostrotunnel*:

```
debian:~# ip link set nostrotunnel up
```

per disattivarla:

```
debian:~# ip link set nostrotunnel down
```

per eliminare il tunnel:

```
debian:~# ip tunnel del nostrotunnel
```

6.4.7 Conclusioni

Come si può intuire l'utilizzo della suite iproute2 permette di compiere operazioni impossibili con gli strumenti classici.

La sua caratteristica più interessante è il fatto che fornisce un unico comando per svolgere tutti i compiti che abitualmente venivano portati a termine da ifconfig, arp, route e ip-tunnel, etc. Certo la curva di apprendimento è sicuramente ripida considerata tra l'altro l'eseguità della documentazione disponibile... .

In quest'articolo si è cercato di dare una panoramica generale (ma non esaustiva) delle potenzialità di questo strumento.

Non sono state trattate le problematiche relative al traffic control, alle statistiche di rete, etc. Per tutto questo si rimanda al lettore il piacere di approfondire e condividere l'argomento.

Happy networking...!

Webografia

<http://www.linuxfoundation.org/Net:Iproute2>

<http://www.pluto.it/files/ildp/lfs/lfs/6.0/chapter06/iproute2.html>

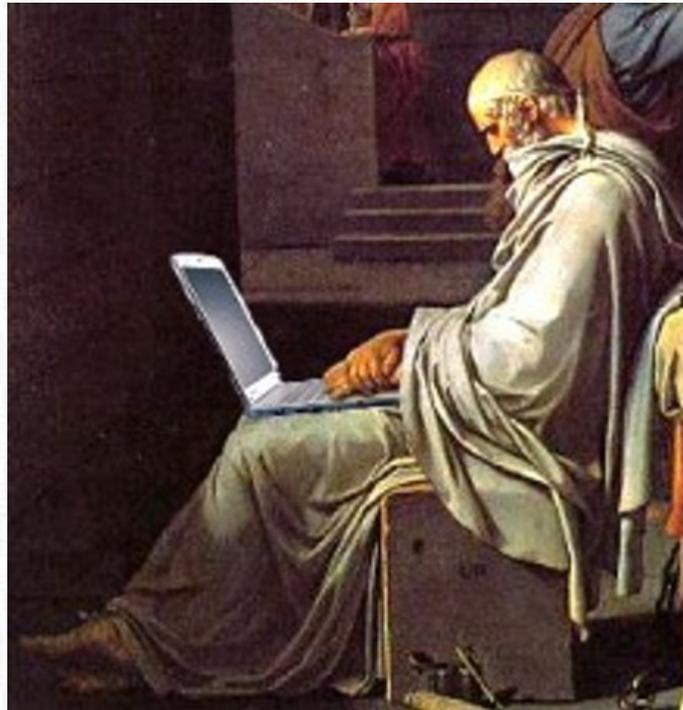
<http://www.deepspace6.net/docs/iproute2tunnel-it.html>

http://www.rigacci.org/wiki/doku.php/doc/appunti/net/source_routing

<http://www.linuxhorizon.ro/iproute2.html>

Capitolo 7

Storia e filosofia del software libero



Nel primo capitolo, argomento uguale, campo diverso. Se debian è un ottimo sistema libero e sicuramente appartiene a questo tema, il software libero si estende ben al di là del *figlio* di Murdock. In questa sezione vi proporremo una serie di articoli relativi all'immenso mondo del software libero. Cercheremo di illustrarvi le basi del pensiero, così come l'evoluzione nel corso degli anni.

7.1 Nexenta: quando opensolaris incontra GNU

7.1.1 Introduzione

In questo numero dedicato al port abbiamo presentato alcune varianti di debian con differenti architetture e differenti kernel. Parlando di quest'ultimi, dopo aver conosciuto Linux, Hurd e in questo numero freeBSD, vi presenteremo in questo articolo un kernel abbastanza conosciuto, ma anche abbastanza criticato: Solaris.

La *Sun microsystem* è fra le prime imprese a lanciarsi nella mischia dei sistemi operativi. L'avventura incomincia nel 1982 con il primo *Sun UNIX 0.7*, basato sullo *UNIX V7* del 1979. Un anno più tardi nascerà il primo *SunOS* versione 1.0, basato sul *BSD 4.1*. Nel 1987, in collaborazione con la AT&T nasce un progetto con l'obiettivo di unificare le più grosse varianti di UNIX sul mercato: il *BSD*, il *System V*, e lo *Xenix* (nota bene, quest'ultimo targato *Microsoft*). Nascerà così il *Unix System V Release 4* (SVR4). Nel 1991 la Sun annuncia di sostituire il *BSD* del *SunOS 4* con il nuovo *SVR4*: si gettano le basi del *SunOS 5*. Quest'ultimo sarà poi chiamato *Solaris 2*; il *SunOS 4* sarà rinominato retroattivamente come *Solaris 1*. Dopo il *Solaris 2.6* si decise di nominare il *Solaris 2.7* come *Solaris 7*. L'ultima versione è il *Solaris 10* del 2005; nonostante potremmo pensare si tratti di una versione obsoleta, l'ultimo aggiornamento risale ad ottobre del 2009.

Nel 2004 si gettano le basi del progetto *Opensolaris*. Nonostante il nome, il progetto - che chiaramente si basa sul fratello *Solaris* - non è un'opera pia della Sun, ma probabilmente una strategia di mercato (n.d.R magari dopo l'esempio dello Sviluppo di Linux negli anni '90). Inizialmente vengono aperte solo parti minime di codice, come ad esempio il *DTrace*. La licenza per le parti libere è la CDDL (*Common Development and Distribution License*, v. webografia); questa licenza è considerata libera dalla FSF (*Free Software Foundation*), ma in ogni caso non compatibile con la GPL (*General Public License*). Fino a questo punto il progetto resta sempre più che altro una bozza. Nel 2007 la Sun ingaggia *Ian Murdock* - il creatore di debian - per rendere *opensolaris* una distribuzione vera e propria, appoggiandosi su GNU e sul desktop GNOME. Murdock riesce abbastanza bene nel suo intento e nel 2008, con *opensolaris 2008.05* si ha una live installabile con il Desktop GNOME. Nel 2009 compaiono i primi portatili Toshiba con *opensolaris* pre-installata. L'ultima versione risale a giugno del 2009, *opensolaris 2009.06*.

7.1.2 GNU/Opensolaris

Questa dicitura riempie le pagine di blog e di forum... in ogni caso resta abbastanza confusa e per alcuni aspetti addirittura inappropriata. Facciamo un paio di considerazioni:

- scritto in questo modo potremmo pensare che opensolaris sia un kernel e forse in parte lo è; in ogni caso opensolaris è un sistema operativo in tutto e per tutto. Il kernel, chiamato correntemente anche opensolaris, è un kernel solaris.
- Il sistema ingloba vari *tools* dal progetto GNU; si potrebbe dunque parlare di ripetizione. A grandi linee, penso che con il termine GNU/Opensolaris si voglia indicare un sistema GNU con kernel opensolaris; in ogni caso, utilizzando il termine opensolaris sia per il kernel che per il sistema operativo si fa una gran confusione.

Il progetto Nexenta si definisce GNU/Opensolaris e si tratta di un sistema operativo basato sul kernel opensolaris appunto e ubuntu LTS (la versione stable di ubuntu).

7.1.3 Nexenta

Il progetto nasce nel 2005 proprio con l'intento di utilizzare un kernel opensolaris/solaris da associare ad una distribuzione GNU/Linux pratica e funzionale. Allo scopo viene scelto ubuntu LTS. Nexenta si basa su due piloni principali:

- NexentaStor: questo è un sistema proprietario ed è rivolto a praticamente tutto ciò assomigli ad un server, NAS, ecc.
- Nexenta OS: anche chiamato NCP (*Nexenta Core Platform*), sarebbe la versione libera. Purtroppo, come suggerisce il nome stesso, *core*, sembrerebbe essere una bozza della versione più completa e più tutto, *NexentaStor*.

La grande briscola di Nexenta è da cercare nel file system fornito dal kernel opensolaris: **ZFS**.

7.1.4 ZFS

ZFS è un file system creato nel 2004 da Solaris. Nel 2005 viene aperto il codice che viene tutt'ora rilasciato sotto licenza CDDL. Senza entrare nei dettagli, il file system ZFS ha molteplici vantaggi rispetto ai file system standard ed è un file system a 128-bit. Vediamo qualche caratteristica del file system:

- **capacità di immagazzinamento** (per esempio, dimensione massima di un file): mentre per il file system ext3 questo limite è posto a 2Tb (2 Terabyte = $2 \cdot 10^{12}$ byte), nel ZFS possiamo usufruire di files da 16 Eb (16 Exabyte = $2 \cdot 10^{18}$ byte), pari a 16 milioni di Terabyte e cioè 8 milioni di volte in più rispetto all'ext3! Se volessimo però guardare i blocchi, rispetto ad un file system a 64-bit (2^{64}), un file system a 128-bit (2^{128}) contiene 16 miliardi di miliardi di dati in più! Questo limite è stato creato per non essere mai raggiunto e dunque per poter effettuare qualsiasi operazione inimmaginabile. Per divertirci un po', grazie a wikipedia (v. webografia) troveremo i dati più sfiziosi a riguardo; il più bello? Secondo mille leggi dalle quali ben poco ho capito, per riempire interamente uno storage a 128-bit non basterebbe l'energia che si otterrebbe facendo bollire gli oceani!
- **Copy-on-write**: questa tecnica è forse uno fra i più grandi vantaggi del file system. Per illustrare meglio come funziona questo principio facciamo un piccolo esempio. Immaginiamo un file da 1Gb, ad esempio un film. Se facessimo ora una piccola modifica al finale di quest'ultimo, risalvando il file avremo occupato ancora 1Gb di memoria, calcolando l'originale, in totale 2 Gb di memoria occupata. Salvando la versione modificata sul file system ZFS, quest'ultimo si limita invece ad una copia virtuale del file originale (essendo magari per il 99% uguale all'originale) andando unicamente a scrivere sul disco la parte modificata (da qui *copy-on-write*). Alla fine avremo un totale di memoria occupata pari a 1Gb e qualche Mb relativi alla parte modificata. Questo principio non solo fa risparmiare spazio fisico sul disco; in un sistema operativo vengono spesso duplicate delle risorse per svariate operazioni. A questo punto, fino al momento che una copia non viene modificata non esiste una vera necessità ad avere fisicamente la copia della risorsa; come già detto ZFS simula la copia e scriverà dei dati sul disco solo al momento in cui saranno veramente necessari e in ogni caso, solo le parti modificate. Visto in questo modo, in questo caso non si tratta di risparmiare qualche byte sul disco, ma bensì di risparmiare un'enormità di tempo (sempre a seconda della grandezza dei files e dalla frequenza delle operazioni), aumentando conseguentemente le prestazioni del sistema (o di un programma).
- **Snapshot**: uno *snapshot* è una copia di file o directory in un determinato momento (immaginatoci la similitudine del termine con una fotografia che riproduce una situazione ad un tempo preciso). Dal momento che ZFS non sovrascrive i dati

nei loro blocchi originari, i blocchi contenenti dati vecchi non saranno cancellati. Grazie a questa proprietà del file system, oltre ad avere *snapshot* rapidi, possiamo ripercorrere lo stato di file e directory in dietro e in avanti nel tempo; questo permette di effettuare back up dinamici come ad esempio con il noto software Time Machine di casa Apple.

I pregi del file system ZFS non sono certo finiti qui; per chi volesse ulteriori informazioni può consultare la webografia al termine dell'articolo.

7.1.5 Debian GNU/Opensolaris

Come abbiamo visto sopra, Nexenta cerca di usufruire dell'organizzazione dei pacchetti e della stabilità di ubuntu LTS (alias, base debian) e delle potenzialità di opensolaris, come visto, soprattutto legate al file system ZFS. Ma come mai non è mai nata una debian GNU/Opensolaris? A quanto si può scrutare in rete la motivazione non è certamente mancata. Il problema, se di problema si può parlare, va cercato nelle licenze e soprattutto nella CDDL, quest'ultima non compatibile con le DFSG (*Debian Free Software Guidelines*). La CDDL è per altro molto simile con la MPL (*Mozilla Public License*); ma come mai vediamo allora tanti plugins di mozilla nei repository principali di debian? Il trucco sta questa volta nelle doppie-licenze: la mozilla rilascia i suoi plugins con licenza MPL E GPL; non fracassatevi ora la testa per comprendere, ma con questa strategia non ci sarebbero problemi legali. Sempre cercando per la rete si trovano informazioni sparse risalenti al 2007 dove la Sun voleva affiancare la GPLv3 ad opensolaris (non è comunque esattamente chiaro se il file system sarebbe anch'esso stato targato con doppia licenza), ossia alla CDDL. A quanto mi risulta questa licenza non è, però, mai stata attribuita, e fino ad oggi opensolaris e il file system ZFS restano per debian un tabù.

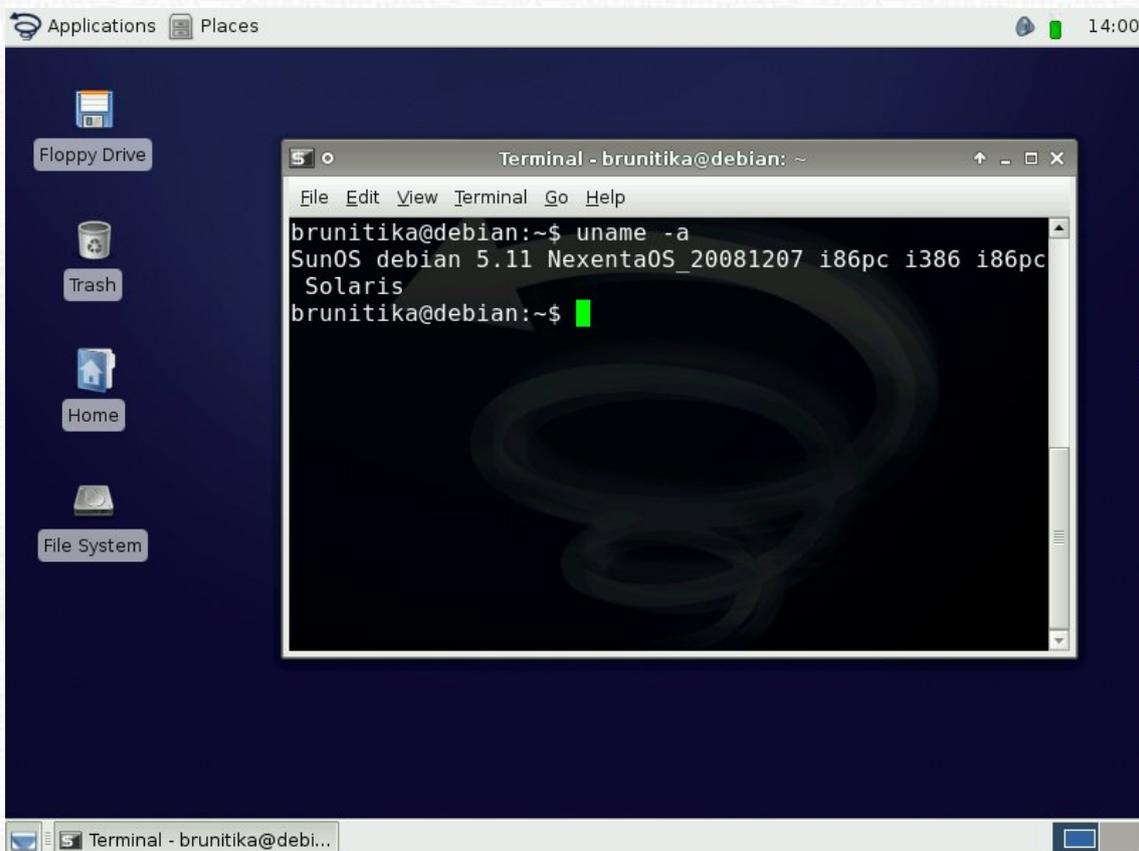
Inoltre non impazzirò per capire cosa ci faccia un pacchetto come *zfs-fuse* nel repository main di debian, il quale permette di installare ZFS come file system virtuale nello *userspace* (*FUSE*)... Mah!! Sarò lieto di sentire chiunque di voi ci capisca qualcosa con queste licenze...

7.1.6 StormOS

Nexenta NCP così come NexentaStor sono sistemi operativi rivolti soprattutto alla rete come sistemi per server. Il sistema operativo StormOS (v. webografia) si basa su Nexenta

stessa, ma si rivolge ad un'utenza di tipo Desktop. A fianco del kernel Opensolaris, sfrutta anch'esso ubuntu LTS con Xfce come *Desktop Environment*.

Per non finire in teoria abbiamo provato ad installare questa distribuzione partendo dall'immagine che si può scaricare dalla pagina del sito ufficiale. L'installer è in veste ncurses; si rispondono alle solite domande e in poco meno di un'oretta si arriva ad un sistema funzionante. Dal momento che non si tratta di una distribuzione debian-compatibile non ci dilungheremo oltre, ma resta comunque un simpatico esperimento al di fuori dei kernel più classici. Giusto per non lasciarvi a bocca asciutta, ecco come si presenta alla vista:



7.1.7 Conclusioni

Il kernel opensolaris (inteso come sviluppato nell'articolo e non *opensolaris* inteso come sistema operativo) è sicuramente interessante, così come lo è il file system ZFS. Praticità e filosofia sono sicuramente fortemente implicate riguardo la licenza che lega questo codice;

del resto, come già discusso, sembra quasi un paradosso l'aver una licenza considerata libera come la CDDL e non riuscire ad integrare maggiormente questo tipo di tecnologia in ambienti *Linux-based*. L'eterno dubbio. In ogni caso è stato interessante venire a conoscenza di questa prima unione GNU/Opensolaris e lasceremo al tempo l'eventualità di vedere un'integrazione maggiore con gli altri sistemi e chissà, magari anche debian.

Webografia

http://en.wikipedia.org/wiki/Nexenta_OS

<http://en.wikipedia.org/wiki/NexentaStor>

<http://en.wikipedia.org/wiki/OpenSolaris>

http://en.wikipedia.org/wiki/Common_Development_and_Distribution_License

[http://en.wikipedia.org/wiki/Solaris_\(operating_system\)](http://en.wikipedia.org/wiki/Solaris_(operating_system))

<http://en.wikipedia.org/wiki/SunOS>

[http://it.wikipedia.org/wiki/ZFS_\(file_system\)](http://it.wikipedia.org/wiki/ZFS_(file_system))

<http://it.wikipedia.org/wiki/Copy-on-write>

<http://www.nexenta.org/os/Documentation>

<http://www.nexenta.org/>

<http://www.nexenta.org/TAoNM/01/1.html>

http://www.osdevcon.org/2009/program_detail.html

http://wapedia.mobi/en/Nexenta_OS

<https://wiki.ubuntu.com/LTS>

<http://www.stormos.org/>

Impressum

Redattori articoli

- Pagina dei lettori (Ripristinare il supporto samba in MC) - *fr4nc3sc0*
- Debian Free Software Guidelines (DFSG) - *brunitika*
- Debian server tramite preseed - *Simone*
- Debian GNU/kFreeBSD - *xtow*
- Debian su Android - *brunitika*
- Vivere l'Emacs - *borlongioffi*
- ALSA & GNOME - *bel.gio*
- Bash: usare gli alias - *gnomob*
- Mai senza Backup: backup2l e clonezilla - *mm-barabba, Aki*
- Il nuovo Networking: iproute2 - *pmate*
- Nexenta: quando opensolaris incontra GNU - *brunitika*

Copertina

Simone, mm-barabba

Impaginazione

borlongioffi (versione stampa), *brunitika* (web-zine)

Collaboratori

fr4nc3sc0, gnomob

Contatto

Tutti i membri del progetto sono reperibili sul forum del portale www.debianizzati.org.

Nota a questa versione stampabile

Quella che state leggendo è la versione stampabile della *e-zine* “Debianizzati” prodotta dalla comunità www.debianizzati.org.

Potete trovare la versione on-line, comodamente consultabile con il proprio browser, all’indirizzo <http://e-zine.debianizzati.org/>.

I sorgenti L^AT_EX di questa versione e delle precedenti, sono disponibili all’indirizzo:

<http://e-zine.debianizzati.org/source/>

Happy Debian, Happy hacking