



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux



Slackware Linux



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Di cosa parleremo?

- Storia e filosofia di Slackware
- Sistema di init
- Gestione dei pacchetti
- Come creare pacchetti per Slackware
- Rami di sviluppo
- Aggiornamento del sistema
- Guida rapida alla configurazione di Slackware
- Slackware links



Storia di Slackware

Slackware è stata la prima distribuzione Linux ad essere ampiamente utilizzata. E' stata creata da Patrick Volkerding nel 1992. Patrick Volkerding si è avvicinato a Linux quando ebbe bisogno di un interprete LISP non costoso per un suo progetto. A quel tempo, c'era un numero esiguo di distribuzioni, così Patrick cominciò con la distribuzione SLS Linux (*Soft Landing Systems*). Tuttavia, SLS aveva alcuni problemi, così Volkerding cominciò a fissare alcuni piccoli *bugs* che aveva individuato. Decise poi di inserire tutti queste correzioni in una sua distribuzione per se stesso e per i suoi amici. Questa distribuzione privata divenne rapidamente popolare e Patrick Volkerding la rese disponibile al pubblico con il nome di Slackware. L'ultima release disponibile è la 9.1 uscita nel settembre 2003.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Filosofia di Slackware

Slackware viene sviluppata e aggiornata da una sola persona, Patrick Volkerding, per questo le release non sono frequenti come per altri prodotti, ma sono curate in modo da non presentare bug o problemi di sicurezza.

Slackware è apprezzatissima da un numero sostanzioso di utenti per la sua semplicità e sicurezza. Semplicità non intesa come facilità d'uso, ma come linearità, pulizia e funzionalità.

Slackware, a detta di molti puristi, rimane, assieme a Debian, forse una delle poche distribuzioni ad incarnare il "vero" spirito Linux.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Slackware in poche parole...

Slackware è una distribuzione con installazione testuale, ma non si tratta assolutamente di un prodotto adatto esclusivamente ad utenti esperti. Una caratteristica propria di Slackware è quella di garantire ai propri utenti un sistema pulito. La linearità e la pulizia fanno sì che sia una delle distribuzioni più veloci.

Slackware è costantemente aggiornata, anche per quel che riguarda gli ambienti grafici, e questo la rende certamente una valida scelta come distribuzione *desktop*. Mancano alcune *utility* di configurazione, quindi l'utente è costretto a editare a mano i vari file di configurazione. Slackware è, più di ogni altra, una distribuzione nata per essere studiata e analizzata.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Dove posso avere la Slackware?

- Scaricandola dal sito ufficiale o dai tanti mirror disponibili.
- Spesso viene allegata a delle riviste Linux.
- Comprandola in negozio, ordinandola presso il sito ufficiale (Slackware Store), o acquistandola sui vari siti di e-commerce (Amazon, ebay,...).



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Releases

9.1 - 26/09/2003	3.1 - 24/07/1996
9.0 - 18/03/2003	3.0 - 24/08/1995
8.1 - 18/06/2002	2.3 - 24/05/1995
8.0 - 01/07/2001	2.2.0 - 30/03/1995
7.1 - 22/06/2000	2.0.1 - 23/09/1994
7.0 - 25/10/1999	2.0 - 2/07/1994
4.0 - 17/05/1999	1.1.2 15/02/1994
3.9 - 10/05/1999	1.1.0 - 5/11/1993
3.6 - 28/10/1998	1.0.4 - 1/10/1993
3.5 - 09/06/1998	1.0.3 - 15/09/1993
3.4 - 14/10/1997	1.0.2 - 5/09/1993
3.3 - 11/07/1997	1.0.1 - 4/08/1993
3.2 - 20/04/1997	1.0 - 17/07/1993



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Slackware 9.1

La versione 9.1 viene distribuita con il kernel-2.4.22, KDE-3.1.4, GNOME-2.4, XFree86-4.3.0 e Gimp-1.2.5.

Tutta la distribuzione è compilata con GCC 3.2.3. Per quanto riguarda l'ambiente di sviluppo, troviamo anche Perl- 5.8.0, Python -2.3.1, php-4.3.3 e j2sdk-1_4_2_01. Questo rende Slackware un ambiente di sviluppo ideale.

Tra gli strumenti offerti da Slackware troviamo *hotplug*, tool per la gestione di schede *hotplug* come quelle PCMCIA e CardBus. C'è anche la possibilità di usufruire di programmi come *swaret* per l'aggiornamento automatico della distribuzione.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

ZipSlack

ZipSlack è una edizione speciale di Slackware Linux che può essere installata in una partizione con filesystem FAT (o FAT32) con almeno 100 MB di spazio libero. Può essere installata e fatta partire da un disco Zip. ZipSlack usa il filesystem UMSDOS e contiene la maggior parte dei programmi che in genere si utilizzano. Questo significa che non è necessario ripartizionare il disco fisso se c'è già installato DOS o Windows.

Questa distribuzione è ideale per chi non ha tanto spazio sul disco fisso, non ha una connessione ad Internet veloce per scaricare una intera distribuzione (ZipSlack è di circa 40MB!), oppure per chi vuole una distribuzione Linux che si può eseguire da un disco Zip.



La Directory `/etc/rc.d/`

I *files* di inizializzazione del sistema sono immagazzinati nella directory `/etc/rc.d/`. Slackware usa lo stile BSD per i suoi *files* di inizializzazione. Ad ogni *task* o *runlevel* è dato il proprio file `rc`. Ciò permette una struttura organizzata che è facile da gestire.

Ci sono varie categorie di files di inizializzazione:

- Avvio del sistema
- Runlevels
- Inizializzazione della rete
- Compatibilità System V



Avvio del sistema

In un sistema Linux il primo programma a venire eseguito è `init`. `Init` esegue lo *script* `/etc/rc.d/rc.S` (*script* di inizializzazione del sistema) per preparare il sistema prima che venga avviato il *runlevel* desiderato. Il file `rc.S` compie le seguenti operazioni:

- Abilita la memoria virtuale
- Monta i *filesystems*
- Pulisce certe directories
- Inizializza le periferiche Plug and Play
- Carica i moduli del kernel (`rc.modules`)
- Imposta le porte seriali (`rc.serial`)
- Fa partire gli *init scripts* di tipo System V (`rc.sysvinit`)



Runlevels

Dopo che l'inizializzazione del sistema è stata completata, `init` prosegue con l'inizializzazione dei *runlevels*. Nella distribuzione Slackware i *runlevels* sono definiti con i seguenti files:

- **rc.0**: arresta il sistema (runlevel 0) ^(*)
- **rc.4**: avvio multiutente (runlevel 4) ^(**)
- **rc.6**: riavvia il sistema (runlevel 6)
- **rc.K**: avvio in modalità utente singolo (runlevel 1)
- **rc.M**: avvio in modalità multiutente (runlevel 2 e 3) ^(***)

(*) Questo file è un *symlink* a `rc.6`

(**) In modalità grafica con KDM, GDM, o XDM come *login manager*.

(***) Con il *login standard* a riga di comando. Questo è il *runlevel* predefinito.



Runlevels...

```
$ cat /etc/inittab
[... ]
# Default runlevel. (Do not set to 0 or 6)
id:4:initdefault:

# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S

# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K

# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M

# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -r now

# Runlevel 0 halts the system.
l0:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
l6:6:wait:/etc/rc.d/rc.6
[... ]
```



Come posso far partire Slackware in modalità grafica?

Slackware può essere avviata in modalità grafica modificando il file `/etc/inittab`. La riga da modificare è la seguente:

```
id:3:initdefault:
```

che dovrà venire modificata in:

```
id:4:initdefault:
```



Come posso specificare il Window Manager di default?

Può essere specificato un *Window Manager* unico per tutti gli utenti che hanno accesso al sistema con il programma `xwmconfig`. In alternativa si può entrare nella directory `/etc/X11/xinit` e creare un *link* simbolico al `xinitrc` da utilizzare:

```
# cd /etc/X11/xinit
# rm xinitrc
# ln -s xinitrc.kde xinitrc
```

Questo specificherà che l'interfaccia da utilizzare sarà KDE.



Inizializzazione della rete

I *runlevels* 2, 3, e 4 avviano i servizi di rete. I seguenti *files* sono responsabili dell'inizializzazione della rete:

- **rc.inet1**: configurazione delle interfacce di rete, sulla base dei parametri specificati in `rc.inet1.conf` (indirizzo IP, netmask, dhcp, *default gateway*).
- **rc.inet2**: parte dopo `rc.inet1` e avvia i servizi di rete basilari (nfs, RPC portmapper, samba, syslog, inetd,...).



Compatibilità System V

La compatibilità con il sistema di init System V è stata introdotta a partire dalla Slackware 7.0. Molte altre distribuzioni (Debian, Mandrake,...) utilizzano questo sistema invece di quello BSD. Con il System V, ad ogni *runlevel* è assegnata una sottodirectory per gli script di avvio, mentre lo stile BSD associa uno script di avvio ad ogni *runlevel*.

Lo script `rc.sysvinit` cerca tutti gli *scripts* di tipo System V che sono presenti in `/etc/rc.d` e li fa partire.



Altri files...

Esistono altri *files* oltre a quelli elencati in precedenza, di solito avviati dagli *scripts* principali:

- **rc.syslog**: avvia il *logger* di sistema
- **rc.keymap**: carica la mappa della tastiera
- **rc.font**: carica il *font* prestabilito per la *console*
- **rc.local**: contiene comandi specifici per lo *startup*

A seconda del software installato compariranno altri *scripts*:

- **rc.httpd**: avvia il web server Apache
- **rc.samba**: avvia Samba

...



Come si abilita uno script?

Per abilitare/disabilitare uno *script* basta aggiungere/togliere i permessi di esecuzione allo script con il comando `chmod`.

In `rc.inet2` ci sono delle sezioni come questa:

```
# Start the OpenSSH SSH daemon:
if [ -x /etc/rc.d/rc.sshd ]; then
    echo "Starting OpenSSH SSH daemon: /usr/sbin/sshd"
    /etc/rc.d/rc.sshd start
fi
```

Il servizio partirà se e solo se lo script `rc.sshd` ha il permesso di esecuzione.



Se l'APM non funziona...

Se, quando si spegne il sistema, il sistema chiude tutti i processi, compare la scritta "Power down" e per arrestare il sistema è necessario premere sul pulsante di *power*, significa che non è stato caricato il modulo APM (*Advanced Power Management*) all'avvio. Per farlo basta editare il file `/etc/rc.d/rc.modules`, trovare la seguente sezione:

```
#### APM support ###  
# APM is a BIOS specification for saving power using several different  
# techniques. This is mostly useful for battery powered laptops.  
#/sbin/modprobe apm
```

e decommentare l'ultima riga (togliendo il “#”)



Gestione dei pacchetti

Anche la distribuzione Linux Slackware utilizza un *software* di *package management*, che permette di tener traccia dei pacchetti installati, rendendone facile la rimozione o l'aggiornamento, oltre che l'installazione di nuovi programmi.

I pacchetti della distribuzione Slackware sono archiviati in un formato molto semplice, `tar+gzip`, e i file relativi utilizzano l'estensione `.tgz`. In pratica, sono il risultato di un'archiviazione attraverso `tar` e di una successiva compressione attraverso `gzip`. L'archivio che si ottiene è fatto in modo da conservare la struttura di directory a partire dalla directory radice e senza contenere i collegamenti simbolici.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Gestione dei pacchetti...

Slackware Linux mette a disposizione alcune *utility* per gestire l'installazione, la rimozione e l'aggiornamento del software e per mantenere traccia delle operazioni tramite un “database”. I principali *tools* forniti da Slackware sono:

- **installpkg**: installa un pacchetto
- **upgradepkg**: aggiorna un pacchetto
- **removepkg** : disinstalla un pacchetto



installpkg

installpkg [opzioni] *package.tgz*

Opzioni:

- warn:** visualizza i cambiamenti nel sistema che avverrebbero in caso di installazione del *package*.
- r:** installa ricorsivamente i *packages* contenuti nella directory corrente e nelle subdirectory. E' possibile utilizzare le *wildcards*.

Al termine dell'installazione viene eseguito, se presente, uno script di nome `doinst.sh` nella subdirectory `install/` del *package* che permette di rifinire l'installazione creando, per esempio, link simbolici, aggiungendo utenti/gruppi,...



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

upgradepkg

upgradepkg *newPackage*
upgradepkg *oldPackage%newPackage*

Aggiorna un pacchetto, disinstallando prima il pacchetto vecchio e installando dopo quello nuovo. Se il nome del pacchetto è lo stesso, non richiede l'indicazione del vecchio pacchetto.



removepkg

removepkg[opzioni] *package*

Opzioni:

-warn: la rimozione viene soltanto simulata.

-preserve: il *package* viene rimosso e copiato per sicurezza in nella directory `/var/log/setup/tmp/preserved_packages`

-copy: il *package* non viene rimosso ma viene copiato nella directory `/var/log/setup/tmp/preserved_packages`.



pkgtool

Pkgtool è un' *utility* di tipo *menu driven* che permette di visualizzare, installare e rimuovere i pacchetti. Con questo *tool* è possibile vedere quali sono i pacchetti installati, visualizzare il contenuto di ogni *package*, disinstallarlo o scegliere da dove installarne uno nuovo. Pkgtool non consente però di effettuare l'aggiornamento dei pacchetti, prerogativa disponibile solo per le *utility* di tipo *command line*, che dispongono anche di un maggior numero di opzioni.

Pkgtool consente anche di avviare alcuni *script* di installazione utili per il *setup* del sistema, come la creazione di un dischetto di boot, la configurazione di LILO, la selezione del *window manager* di *default*, la configurazione della rete,...



Package tools

A differenza degli altri formati di pacchettizzazione come i `.deb` e gli `.rpm`, i pacchetti di Slackware sono decisamente più semplici da preparare... del resto le funzioni che svolgono sono molto più elementari in quanto non è previsto un controllo sulle dipendenze. Esistono numerosi *tools* con i quali è possibile creare dei pacchetti per Slackware Linux. I più utilizzati sono:

- Checkinstall
- Makepkg
- RPM
- rpm2tgz
- Alien



Compilazione dei sorgenti

Chi utilizza Slackware generalmente compila il software a partire dai sorgenti, per avere ottimizzazioni per la propria architettura e/o configurazioni personalizzate.

```
$ tar zxvf pacchetto.tar.gz
$ cd directory_appena_creato
$ ./configure [opzioni]
$ make
$ su
# make install
```

Con questo approccio risulta difficile la gestione del software installato, in particolare per quanto riguarda la disinstallazione (ad esempio perchè il Makefile è stato cancellato oppure non prevede la direttiva di uninstall,..)

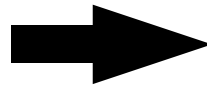


Checkinstall

Checkinstall (<http://asic-linux.com.mx/~izto/checkinstall/>) è una *utility* che consente di creare dei pacchetti .tgz per Slackware^(*) come parte integrante del normale processo di installazione a partire dai sorgenti. Checkinstall usa l'*utility* `installwatch` per tenere traccia dei files modificati durante l'installazione.

NORMALE

```
$ tar zxvf pacchetto.tar.gz  
$ cd directory_appena_creat  
$ ./configure  
$ make  
$ su  
# make install
```



CON CHECKINSTALL

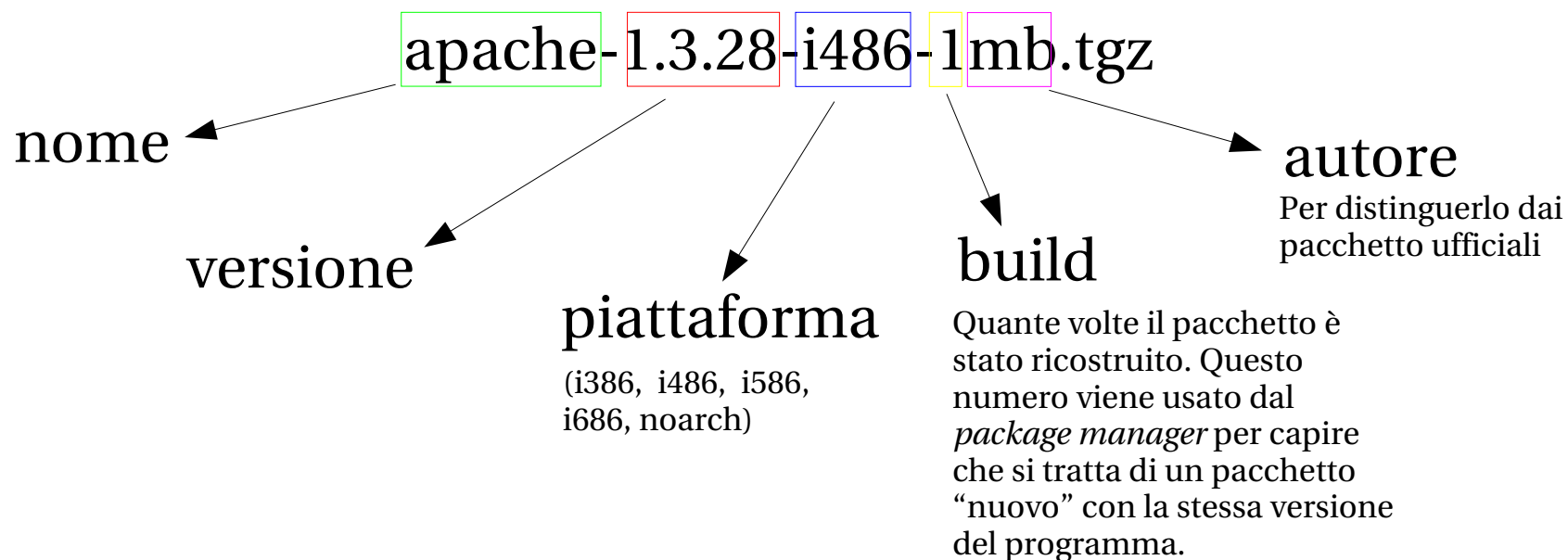
```
$ tar zxvf pacchetto.tar.gz  
$ cd directory_appena_creat  
$ ./configure  
$ make  
$ su  
# checkinstall
```

(*) Checkinstall può creare anche dei pacchetti .rpm e .deb.



Il package perfetto (nome)

Per costruire un pacchetto perfetto è necessario seguire alcune regole. E' importante porre attenzione al nome in quanto esso “descrive” il pacchetto: il nome e la versione del programma, la piattaforma, la versione di *build* e autore. Ad esempio:





Il package perfetto (directory)

E' necessario definire le `directory` che verranno utilizzate durante l'installazione. Le `directory` variano a seconda del programma, ma una tipica configurazione “minima” potrebbe essere :

```
./
+--- /install
|
|   +--- slack-desc (descrizione)
|   +--- slack-required (optionale) (eventuali dipendenze)
|   +--- doinst.sh (comandi da eseguire dopo l'installazione)
|
+--- /usr/doc/package-version/ (documentazione)
|
+--- /etc (files di configurazione)
|
+--- /usr (binari)
|
+--- /var (files, logs, pids etc..)
```



Il package perfetto (configurazione)

Seguono delle tipiche opzioni usate in fase configurazione:

/usr: in questa area vengono installati tutti i binari. Per ottenere questa configurazione basta usare il comando di configurazione `--prefix=/usr`.

1

```
configure --prefix=/usr
          --sysconfdir=/etc \
          --localstatedir=/var
```

/etc: questa area contiene i files di configurazione principali. Il comando di configurazione `--sysconfdir=/etc` fa sì che i files di configurazione vengano messi in questa directory. (se si specifica semplicemente `--prefix=/usr` e il pacchetto ha dei files di configurazione, questi saranno messi in `/usr/etc/`). Alcune volte è preferibile mettere i files di configurazione in una sottodirectory di `/etc`.

/var: questa area contiene files, logs, pids etc.. Il comando di configurazione corrispondente è `--localstatedir=/var`.



Il package perfetto (Makepkg)

Dopo la configurazione deve essere eseguita la compilazione (*make*) e l'installazione (*make install*). Di *default* l'installazione viene fatta usando come “base” la directory radice (*/*), mentre noi vogliamo eseguire l'installazione in un directory “pulita”, ad esempio `/tmp/build`, che replichi la struttura propria della *root*.

2

```
$ make
# make install DESTDIR=/tmp/build
```



```
/tmp/build
|
+--- /install
+--- /usr/doc/package-version/
+--- /etc
+--- /usr
+--- /var
```

3

```
# cd /tmp/build
# makepkg -l y -c n /tmp/apache-1.3.28-i386-1mb.tgz
```



Risultato: gli SlackBuilds!

Lo SlackBuild è semplicemente uno *script* contenente una lista di comandi strutturati in modo tale da generare un pacchetto Slackware base, installabile con i tool `pkgtool`, `installpkg` o `upgradepkg`. Lo *script* lavora sulla base di programmi *standard* come `make` e `gcc`. L'utilizzo è molto semplice, basta scaricare i sorgenti, lo `*.SlackBuild` e lo `slack-desc` in un'unica directory, eventualmente modificare le CFLAGS in base al tipo di processore e, come utente `root`, eseguire lo script:

```
# ./*.SlackBuild --cleanup
```

Il pacchetto pronto per l'installazione verrà creato nella directory `/tmp`. L'opzione “`--cleanup`” elimina automaticamente le directory di lavoro.



Il package perfetto (slack-descr)

Il file `slack-desc` è usato dal sistema di *package management* di Slackware. Esso permette di contenere una descrizione del pacchetto da installare. Se il formato non è corretto, i vari *tools* non mostreranno correttamente queste informazioni. Il file deve contenere esattamente 11 linee. Il file `slack-desc` è molto importante poiché aiuta a tenere il database dei pacchetti aggiornato con delle buone informazioni. Inoltre esso aiuta anche alcuni *tools* a identificare il pacchetto stesso.

All'indirizzo <http://www.linuxpackages.net/slackcreator.php> è disponibile una utility *online* per la creazione di un perfetto file `slack-descr`.



Il package perfetto (slack-descr)...

A titolo di esempio, si riporta un *layout* “raccomandato”

```
# HOW TO EDIT THIS FILE:
# The "handy ruler" below makes it easier to edit a package description. Line
# up the first '|' above the ':' following the base package name, and the '|'
# on the right side marks the last column you can put a character in. You must
# make exactly 11 lines for the formatting to be correct. It's also
# customary to leave one space after the ':'.

|-----handy-ruler-----|
bind: bind version (DNS server and utilities)
bind:
bind: The named daemon and support utilities such as dig, host
bind: and nslookup. Sample configuration files for running a
bind: simple caching nameserver are included. Documentation
bind: for advanced setup can be found in /usr/doc/bind-9.x.x/.
bind:
bind:
bind: Package created by YOUR NAME HERE
bind:
bind:
```



Il package perfetto (documentazione)

Praticamente ogni tipo di licenza richiede che la licenza sia distribuita assieme al programma, questo è il motivo per cui è richiesta la directory `/usr/doc/package-version`. Si tratta soprattutto di *free software* e sono poche regole che bisogna seguire. Questa directory deve anche contenere delle informazioni su come usare il programma (INSTALL, README...). E' anche un buon posto per gli esempi dei files di configurazione usati dal programma.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Il package perfetto (permessi)

I *files* installati da un programma devono essere di proprietà del giusto “*user*” (root, bin, etc...). Se questi permessi non sono corretti, si può compromettere l'intera macchina!



Il package perfetto (slack-required)

Come vedremo, esistono vari *tools* automatici per aggiornare la distribuzione. Affinché il pacchetto che vogliamo creare possa essere usato da tutti questi *tools*, è necessario specificare la lista di tutte le dipendenze richieste nel file `slack-desc` e nel sito web dove rendiamo disponibile il pacchetto. Si può anche aggiungere nella directory `/install` un file chiamato `slack-required`. Questo file contiene una entry per linea con i seguenti formati:

1`package_name`

Ad esempio:

`man`**2**`package_name [condition] [version]`Dove `[condition]` può essere: `=`, `>=`, `=<`, `<`, or `>`. Ad esempio:`man >= 1.51-i386-1`



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Rami di sviluppo

Slackware viene sviluppata parallelamente su due versioni:

- Versioni stabili
- Current

Le versioni stabili sono le varie *releases* e subiscono delle modifiche solo per quanto riguarda la sicurezza. La current è invece il ramo di sviluppo, che darà origine alla successiva *release* stabile.



Aggiornamento della distribuzione

Esistono numerosi *tools* con i quali è possibile aggiornare in modo automatico una Slackware, oppure passare da una versione all'altra “on the fly”. I più utilizzati sono:

Per un singolo PC:

- Swaret
- Slapt-get
- Emerde

Per tante macchine:

- SlackCheck



Swaret

Swaret (<http://www.swaret.org>) è in grado di eseguire il download, aggiornare, installare e re-installare pacchetti, *patches* e sorgenti (con *build scripts*). Supporta il check delle dipendenze e il “tracks and fix” delle librerie mancanti (tramite `ldd`). La risoluzione delle dipendenze usa un “*master file*” sul sito web di Swaret. Swaret non richiede delle particolari modifiche ai pacchetti Slackware. Come fonti si possono usare i mirrors di Slackware (HTTP, FTP, rsync), oppure un *filesystem* locale. E' anche possibile utilizzare pacchetti di terze parti, ad esempio quelli disponibili su LinuxPackages.net o altri *repository*. E' supportata l'internazionalizzazione (i18n). Per Swaret è disponibile Kswaret (<http://www.kswaret.org>), una interfaccia grafica basata su KDE/QT, Gswaret (basato sulle libreria ncurses) e zswaret (basato su Zenity).



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Slapt-get

Slapt-get (<http://software.jaos.org>) è un sistema APT-like per Slackware. Attraverso semplici comandi consente di trovare i pacchetti nei vari *mirrors* Slackware e nei *repository* di terze parti (come www.linuxpackages.net), eseguire l'aggiornamento dei pacchetti installati, installare nuovi pacchetti. A differenza di Swaret, il check delle dipendenze è realizzato attraverso un file da inserire nei pacchetti (`slack-required`). Questo approccio è più affidabile rispetto a quello usato da Swaret in quanto le dipendenze non sono collegate esclusivamente alle dipendenze delle sole librerie. Per Slapt-get è anche disponibile `slapt-GUI` (<http://akinimod.sourceforge.net/slaptgui/>), una interfaccia grafica basata su Gnome/GTK.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Emerde

Emerde (<http://emerde.freaknet.org>) è un *port* del sistema Portage presente nella distribuzione Gentoo. Consente di installare, rimuovere e aggiornare il software in modo rapido e semplice. Emerde consente di installare pacchetti .tgz, garantendo un corretto check delle dipendenze.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

SlackCheck

SlackCheck permette di aggiornare molti PC con Slackware Linux con gli ultimi pacchetti disponibili sui *mirrors* ufficiali. Tutti gli aggiornamenti sono eseguiti da un singolo computer tramite SSH. SlackCheck genera uno script di upgrade e una lista dei pacchetti non standard per ogni macchina.



Configurazione

Esistono vari *tools* con i quali è possibile configurare la Slackware. I più famosi sono:

- **netconfig**: hostname, domainname, IP address, gateway,...
- **liloconfig**: bootloader
- **mouseconfig**: mouse, gpm
- **timeconfig**: UTC, timezone
- **pppsetup**: account dialup
- **xwmconfig**: default windowmanager



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Distribuzioni derivate da Slackware



Vector Linux (<http://www.vectorlinux.com>)



College Linux (<http://linux.college.ch>)



Buffalo Linux (<http://buffalolinux.dyndns.org>)



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Le distribuzioni derivate da Slackware



SLAX (*<http://slax.linux-live.org>*)



RUNT - ResNet USB Network Tester
(*<http://www.ncsu.edu/resnet/runt/>*)



La pagella (fonte: HTML.IT)

Sicurezza: voto **5/5** ★★★★★

Slackware rimane una delle distribuzioni in assoluto più sicure assieme a Trustix e Debian. Sotto questo punto di vista la si può ritenere quasi allo stesso livello di quest'ultima.

Stabilità: voto **5/5** ★★★★★

Oltre che sicura, Slackware risulta essere anche una distribuzione decisamente solida, alleggerita com'è da tutti gli orpelli grafici presenti in altre distribuzioni.

Semplicità: voto **2/5** ★★

Slackware non è di certo indicata a un utente alle prime armi, l'assenza di *tools* grafici poi non dispone bene l'utente che ha poca voglia di andare a modificare manualmente dei file per configurare il proprio sistema.



Belluno Linux User Group
<http://belluno.linux.it>

slackware
linux

Fonti e links

- The Slackware Linux Project (www.slackware.com)
- The Official Slackware Store (<http://store.slackware.com/cgi-bin/store>)
- Userlocal (<http://www.userlocal.com>)
- Linux Packages (www.linuxpackages.net)
- The SlackFiles (<http://www.slackfiles.net/index.php>)
- Abnormal Penguin (<http://abnormalpenguin.com>)
- Slackware Support (<http://www.slackwaresupport.com>)
- Cybertron Slackware site (<http://slackware.cybertron.info/>)
- Slacky.it (<http://www.slacky.it>)
- Slackware Italia (<http://www.slackware-italia.com>)
- The Slack-it Linux Project (<http://web.tiscali.it/gnu.linux>)
- Slackware.Azzurra.org (<http://slackware.azzurra.org>)
- SWARET - SlackWARE Tool (<http://www.swaret.org>)
- Slapt-get (<http://software.jaos.org>)
- SlackCheck (<https://georgi.cybcom.net>)