



BSD: ma che “diavolo” è?

- Preambolo
- BSD licensing style
- Slices e partizioni
- Sistema = Kernel + Userland
- Sincronizzazione del sistema
- Rami di sviluppo (-current, -stable, -release)
- Sistema di gestione dei pacchetti (sorgente e binario)
- Cenni al sistema di init
- Documentazione
- Altre particolarità





Belluno Linux User Group

Distribuzioni Linux e *BSD



Preambolo

- Che “**diavolo**” è **BSD**?
 - **Berkeley Software Distribution**, inizialmente un insieme di “patch” ed “utilities” per l'originale “Bell Unix®”, è diventato un autonomo sistema operativo, sviluppato all'università di Berkeley (California), il cui antenato è, appunto, *NIX:
- Quanto e quale software si può trovare?
 - **Tutto** il software scritto per *NIX/Linux (tramite compilazione);
 - E' il *medesimo* software reperibile in Linux (gli stessi “Gnome”, “KDE”, “Apache”, “OpenOffice”, “Gimp”, “mplayer” e tutto l'open source possa venire in mente!);
 - Soluzioni di compatibilità binaria/emulazione qualora non fossero disponibili i sorgenti.



Licenza BSD: è software libero!

- *BSD viene fornito gratuitamente e con il codice sorgente *completo*;
- Ne è permessa la redistribuzione e l'uso (“*fatene ciò che volete*”) sia in formato sorgente che binario, con due uniche condizioni:
 - Non ci sono garanzie implicite o esplicite;
 - Deve essere riconosciuta la nota di copyright degli autori;
- Rispetto alle licenze GNU sono *assenti* i vincoli imposti dal “*copyleft*”. Ad esempio, è possibile utilizzare il codice e derivare lavori senza obbligo di redistribuzione dei sorgenti. La licenza è quindi estremamente **permissiva!**
- E' una licenza **approvata** “**OSI**” (Open Source Initiative);
- La versione “*modified*” (ovvero quella “corrente”, cui è stata tolta la nota di copyright dal testo, non essendo quella originale quasi più utilizzata) è elencata tra le licenze “**GPL-compatibili**”.



Slices e partizioni (arch PC/i386)

- *BSD va installato in una “partizione primaria” (PC BIOS partition);
- Nell'architettura “PC” vi possono essere fino a 4 partizioni (BIOS) primarie;
- *BSD chiama “**slice**” ogni partizione (BIOS) primaria, e definisce le proprie “partizioni”, in apposita tabella delle partizioni (**disklabel**) definita nella sua slice;
- Ovviamente è possibile montare partizioni Linux, fat32, ntfs (quest'ultima in lettura), ed altre;
- Altrettanto ovviamente, è possibile sfruttare la rete per montare filesystems samba, NFS, ed altri.



Sistema = kernel + userland

- *BSD ha sempre avuto un modello di sviluppo **centralizzato** e **coordinato** (nonché **controllato**, attraverso CVS);
- Oltre al kernel, fanno parte del sistema di base anche tutti i *comandi* e le *utilities* necessarie per il corretto funzionamento del sistema stesso, eseguiti nello spazio utente (la “**userland**”);
- La userland viene rilasciata (e testata) assieme al kernel con cui dovrà funzionare;
- I kernel *BSD **non** sono parenti del kernel Linux. Comandi ed utilities non sono necessariamente gli stessi di Linux (ad es: “ls”, la “libc” sono BSD e non GNU) pur essendo presente in BSD software GNU (gcc ha sostituito cc);
- Tutto il software aggiuntivo è trattato in modo differente dal sistema di base, attraverso un potente sistema di gestione dei pacchetti.



Sincronizzazione di FreeBSD

- E' banale tenere aggiornati i sorgenti per tutto il sistema o parte di esso (kernel, userland, docs, pacchetti) in primis:
 - Anonymous **CVS**, **CVSup** (*pull* update);
 - **CTM** (via e-mail, *push* update);
- E' possibile sincronizzare qualsiasi “versione” (v. *infra*), o i sorgenti di interesse a qualsiasi data (il CVS è attivo fin dalle origini);
- E' possibile sincronizzare sistematicamente (ed in modo “continuo”) il sistema (o una parte, o un ramo) in base ai sorgenti disponibili in CVS, con un approccio graduale e non a blocchi monolitici (l'update, volendo, è continuo).



Esempio di aggiornamento tramite CVSUp su FreeBSD

Il seguente esempio aggiorna i sorgenti dell'intero sistema base all'ultima versione -STABLE (v. *infra*):

```
# cp /usr/share/examples/cvsup/stable-supfile my-supfile
# vi my-supfile (indicare il mirror da usare, e volendo altre opzioni)
# cvsup -g -L 2 my-supfile
```

è quindi possibile ricompilare il sistema stesso (vedere anche l'eventuale file **/etc/make.conf**):

```
# cd /usr/src
# make buildworld
# make buildkernel
# make installkernel
# reboot

# mergemaster -p
# make installworld
# mergemaster
# reboot
```



Rami di sviluppo

- **-CURRENT**: ultime versioni funzionanti dei sorgenti, contenenti lavori in corso, modifiche sperimentali, nuove tecnologie ed eventualmente tecnologie che potrebbero non essere presente nelle successive versioni. Gli utenti che tracciano -current devono essere in grado di risolvere da soli problemi anche complessi, e seguire le M.L!
- **-STABLE**: ramo di sviluppo dal quale originano le principali release. E' ancora un ramo di sviluppo, ma il contenuto è già stato testato in -current;
- **-RELEASE**: versioni rilasciate ufficialmente dei rami current o stable (ad es, le immagini "iso" dei CD).



Gestione dei pacchetti

- Due tecnologie complementari;
- **Packages**: pacchetti binari precompilati:
 - Minori necessità di spazio in installazione;
 - Minore necessità di tempo (no compilazione);
- **Ports/Pkgsrc**: struttura con le direttive per download e compilazione automatizzata dei pacchetti (e patch):
 - Compilazioni ottimizzate e specifiche per il sistema ospite;
 - Combinazione a piacere dei parametri di compilazione;
 - Albero dei ports/pkgsrc aggiornabile con CVS(up) ed altro;
- Risoluzione automatica delle dipendenze e dei download, funzionamento in **rete** o da **CD/DVD/HD** ed *altri supporti*;
- Tracciabilità, aggiornamento e disinstallazione.



Esempi packages

`# pkg_add -r bash` (utilizza `fetch(3)` per scaricare i files)
installa dalla rete la versione adeguata di `bash` per il sistema
corrente, e le eventuali dipendenze;

`# pkg_info | grep python` lista dei software python
installati;

`# pkg_delete python-2.3.2_3` cancella il pacchetto
indicato;

`# pkg_version` verifica le versioni installate con quelle nei
ports;

Tutte le informazioni sui pacchetti vengono memorizzate in
`/var/db/pkg`.

(Gli esempi si riferiscono a FreeBSD)



Esempi ports/pkgsrc

Installazione di bash e delle necessarie dipendenze (esempi riferiti a FreeBSD):

```
# cd /usr/ports  
# make search name=bash  
# cd /usr/ports/shells/bash2  
# make install && make clean  
  
# make deinstall (disinstalla)  
  
# make reinstall (reinstalla)
```

Stampa dei pacchetti necessari alla compilazione/installazione:

```
# make pretty-print-build-depends-list
```

Aggiornamento via rete dell'ultima versione disponibile dei pacchetti:

```
# cp /usr/share/examples/cvsup/ports-supfile myports  
# vi myports (scelta del mirror)  
# cvsup -g -L 2 myports
```

I sorgenti e le patches scaricate vengono salvate in **/usr/ports/distfiles**. Se sono già presenti e corrispondono a quelli del port non viene scaricato nulla.



Portupgrade (FreeBSD)

- Per facilitare le operazioni di aggiornamento (compilazione nuovo pacchetto, deinstall vecchio, install nuovo, clean) anche e soprattutto in presenza di molteplici dipendenze esiste:
- **sysutils/portupgrade**

```
pkg_add -r portupgrade  
pkgdb -F
```
- Portupgrade può aggiornare l'intero sistema, un solo pacchetto, pacchetti dipendenti da un altro o quelli da cui dipende, utilizzando solo i packages, solo i ports, una combinazione preferenziale dei due, utilizzando files "locali" o la rete o entrambi, e molto altro ancora...



Cenni al sistema di init (rc)

- `/etc/rc`: esegue gli script contenuti in `/etc/rc.d/`;
- `/etc/rc.*`: esecuzione di insiemi specifici di istruzioni (*ad es, `/etc/rc.shutdown` [Free] o `/etc/shutdown` [Net] chiamano gli script a rovescio, `/etc/rc.sysctl` [Free] consente di impostare i parametri `sysctl` del kernel all'avvio, e così di seguito*);
- `/etc/defaults/rc.conf`: default per tutti i parametri di avvio del sistema “rc”, utilizzati dai vari script. Da **NON** modificare;
- `/etc/rc.conf`: personalizzazioni dei parametri rispetto ai defaults. La gran parte della configurazione del sistema si può effettuare modificando questo file;
- `/usr/local/etc/rc.d`, e `/usr/X11R[x]/etc/rc.d/`: [Free] esegue tutti gli script eseguibili trovati nella directory, terminanti in `.sh`, passandogli un parametro: **start** all'avvio, **stop** allo shutdown;
- `/etc/rc.local`: ulteriori personalizzazioni locali (non presente di default);
- L'acronimo “rc” deriva una utility di creazione di script chiamata “runcom” (*r*)un (*c*)ommands.



Documentazione

- Essendo sistemi molto ben coordinati, la documentazione è univoca ed abbondante;
- Esistendo “poche” varianti la documentazione è realmente applicabile alle concrete necessità;
- Ogni “versione” dispone della propria documentazione (concetto di “sistema”);
- “**HandBook**”, articoli, man pages, README, FAQ, ...
- Internet, `/usr/share/doc`, `/usr/share/examples`



Particolarità (non è un "System V")

- **Devices:** i nomi dei dispositivi in *BSD sono definiti dalla categoria di dispositivo stessa, ad esempio:
 - Rete: `rln`, `fxpn`, `sisn`, `sipn`, ... -> `ethn`;
 - Disco: `adn[s][x]`, `acdn`, `wdn[x]`, `cdn[x]`, ...
~> `hd[a-d][n]`;
- **Sysctl:** consente di leggere ed impostare lo stato del kernel (parametri...) in esecuzione (ad es: `sysctl kern.maxproc` restituisce il numero massimo di processi consentiti nel sistema);
- **Filesystem:** `man hier`



Belluno Linux User Group

Distribuzioni Linux e *BSD



Bibliografia e riferimenti

“La storia di tutte le maggiori civiltà galattiche tende ad attraversare tre fasi distinte e ben riconoscibili, ovvero le fasi della Sopravvivenza, della Riflessione e della Decadenza, altrimenti dette fasi del Come, del Perché e del Dove.

La prima fase, per esempio, è caratterizzata dalla domanda Come facciamo a procurarci da mangiare?, la seconda dalla domanda Perché mangiamo? E la terza dalla domanda In quale ristorante pranziamo oggi?

[...]

E allora andiamo a mangiare qualcosa – disse Zaphod. - il ristorante *al termine dell'universo* è giusto da queste parti.”

(Douglas Adams - *The Hitch-Hiker's Guide To The Galaxy*)



- BLUG – Belluno Linux User Group:
 - <http://belluno.linux.it>
- Principali *BSD (in ordine alfabetico):
 - <http://www.freebsd.org>
 - <http://www.netbsd.org>
 - <http://www.openbsd.org>
- Siti italiani (in ordine alfabetico):
 - <http://www.gufi.org>
 - <http://www.netbsd.it>
 - <http://www.openbeer.it>

Queste slides si possono trovare all'indirizzo:

<http://linux.studiomasson.it/manopc/varie/diavoliepinguini-2004-04-24/>