

Introduzione al mondo FreeBSD

Corso Avanzato

- Struttura
- Installazione
- Configurazione
- I ports
- Gestione
- Netstudent <http://netstudent.polito.it>
- E.Richiardone err@netstudent.polito.it
- Novembre 2012
- CC-by
<http://creativecommons.org/licenses/by/3.0/it/>

The FreeBSD project - 1

- E` un progetto software open
- Lo scopo e` mantenere e sviluppare il sistema operativo FreeBSD
- Nasce su CDRom come FreeBSD 1.0 nel 1993
- Deriva da un patchkit per 386BSD, eredita codice da UNIX versione Berkeley 1977
- Per problemi legali subisce un rallentamento, release 2.0 nel 1995 con codice royalty-free
- Dalla release 4.0 (2000) assume la struttura che ha oggi
- Disponibile per x86 32 e 64bit, ia64, MIPS, ppc, sparc...
- La mascotte (Beastie) nasce nel 1984

The FreeBSD project - 2

- Erede di 4.4BSD (e` la stessa gente...)
- Sistema stabile; sviluppo uniforme; codice molto chiaro, ordinato e ben commentato
- Documentazione ufficiale ben curata
- Licenza molto permissiva, spesso attrae aziende per progetti commerciali:
- saltuariamente progetti collaborano con implementazioni ex-novo (i.e. Intel, GEOM, NDISwrapper, ZFS, GNU/Linux emulation)
- Semplificazione di molte caratteristiche tradizionali UNIX

Di cosa si tratta

Il progetto FreeBSD include:

- Un sistema base
 - Bootloader, kernel, moduli, librerie di base, comandi e utility di base, servizi tradizionali
 - Sorgenti completi in /usr/src (~500MB)
 - E` gia` completo (i.e. ipfw, ppp, bind, ...)
- Un sistema di gestione per software aggiuntivo
 - Ports e packages
- Documentazione, canali di assistenza, strumenti di sviluppo
 - Handbook, mailling list, forum, pointyhat

Chi fa` cosa - 1

- Struttura gerarchica
- Il sistema base e` aggiornato su un server CVS pubblico, in scrittura da account autorizzati (i committer) per diversi ambiti
- La struttura delle directory riflette questi ambiti
- Canali di comunicazione ufficiale: mailing list pubbliche, forum
- Chiunque puo` sottoporre richieste o patch (meglio!)
- Spesso c'e` un responsabile esperto per diversi ambiti (network, ports, device driver, ...)
- Un core team viene eletto ogni 2 anni per indicare la strada da seguire per le nuove release

Chi fa` cosa - 2

- La notifica di potenziali bug e l'inserimento e aggiornamento dei ports utilizzano la procedura di PR
- La procedura e` verificata da un committer o responsabile, notifiche per mailing list
- Tutte le procedure sono definite da appositi handbook
- Documentazione sistema analogo
- Ci sono comitati speciali che seguono progetti estesi (i.e. KDE, Gnome, Java, Xen, ...)
- Sono tutti volontari, circa 4000 persone

Releasing - 1

- Con RELEASE si intende una versione ufficialmente completa: sono stati generati CDROM e packages di tutti i ports, il sistema e` ragionevolmente stabile
- Ogni BRANCH introduce alcune novita` sostanziali
- Vengono aggiornate nuove RELEASE della BRANCH **attuale** e **precedente**: nella prima con nuove funzionalita`, nella seconda con bug e security fix
- La versione STABLE si ottiene per ogni BRANCH dall'ultima RELEASE aggiornando da CVS
- La versione CURRENT e` di testing: da questa si estraggono novita` per l'ultima RELEASE o si passa ad una nuova BRANCH

Releasing - 2

- A maggio 2011:
- La versione piu` recente disponibile su CDROM e` :
 - FreeBSD 8.2-RELEASE
- La versione piu` recente stabile e` :
 - FreeBSD 8-STABLE
- La versione legacy piu` recente e` :
 - FreeBSD 7.4-RELEASE
- Analogamente la versione legacy piu` stabile e` :
 - FreeBSD 7-STABLE
- La versione current e` :
 - FreeBSD 9-CURRENT

Installazione

- Sono disponibili diversi CDROM per ogni release:
 - bootonly / disc1 / disc2 / livefs
- E` possibile eseguire molti tipi di installazione
- Per chi inizia o non ha collegamento di rete:
 - Installazione da disc1 e disc2, utilizzando sysinstall/bsdinstall e pacchetti precompilati (packages)
- Per chi vuole un sistema ottimale:
 - Installazione minimale da uno qualsiasi dei dischi, aggiornamento alla STABLE dei sorgenti, ricompilazione; installazione dei pacchetti da sorgente (ports)

Configurazione

- **Sysinstall**, programma di base in ncurses per iniziare
- Da FreeBSD 9, **bsdinstall**
 - piu` intuitivo
 - meno dipendenze
 - scriptabile
 - utilizza **pkgng**
- Viene eseguito dal cdrom dopo l'avvio

Configurazione fine

- File classici *nix, vedi /etc
- Configurazione del sistema (impostazioni host, servizi da eseguire, etc...):
 - **/etc/rc.conf**
- Configurazione delle compilazioni (anche perl, python, etc...):
 - **/etc/make.conf**
- Configurazione parametri di avvio:
 - **/boot/loader.conf**
- Dispositivi plug n play:
 - **/etc/devd.conf**

Interfacce

- Interfaccia di configurazione parametri del sistema:
 - **sysctl**
 - Non e` altro che un comando che interroga e imposta il comportamento del kernel
- Info e controlli dei bus ATA (dischi IDE)
 - **atacontrol**
- Info e controlli dei bus SCSI (dischi SCSI, SATA, molto utile per i RAID hardware)
 - **camcontrol**
- Comandi dischi ottici
 - **cdcontrol**

Gestione fisica

- Elenco e diagnostica dispositivi:
 - **devinfo**
 - **pciconf -lv**
- vedere i messaggi di sistema
 - `/var/log/messages < dmesg`

Gestione sistema

- sysctl verso kernel land
- Verso user land:
 - **systat**
 - -iostat, vmstat, ...
 - fstat
 - netstat

Ports - 1

- I ports non sono altro che una collezione di Makefile e patch per aggiungere svariati programmi al sistema base a partire dai sorgenti ufficiali
- I makefile permettono altre svariate funzioni
- Questi programmi aggiuntivi sono oltre 20000 e organizzati in sotto-directory per argomento
- Gestione dipendenze di esecuzione e compilazione
- Presenti **/usr/ports** dalla prima installazione
- Tipicamente i binari installati vanno sotto **/usr/local/bin**, le librerie sotto **/usr/local/lib**, le configurazioni sotto **/usr/local/etc**, gli script di avvio **/usr/local/etc/rc.d**

Ports - 2

- Per esempio dai ports si lancia:
 - **make config**
 - **make install**
 - **make clean**
 - **make deinstall**
- ... oppure combinati i.e. **make install clean**
- Compilazione da utente, installazione da root
- Di default la compilazione di ports scarica i sorgenti in **/usr/ports/distfiles**, e compila i file in **./work**

Ports e packages

- L'elenco dei programmi installati con le relative dipendenze (B-deps e R-deps) e` in un database in **/var/db/pkg**
- Per ogni release e ogni architettura tier-1 sono disponibili tutti i ports precompilati (packages)
- Due strumenti di gestione:
 - **pkg_add pkg_delete pkg_info**
 - **pkg** (pkgng)
- L'utilizzo di packages e ports e` interoperabile
- **pkg_*** utilizza pacchetti .tgz e per le dipendenze l'albero dei ports locale
- **pkgng**, utilizza pacchetti .txz e dipendenze remote

Aggiornamento ports

- L'aggiornamento delle directory dei ports avviene come per il sistema base tramite CVS, oppure tramite **portsnap**
- Per aggiornare i pacchetti tramite pkg_* ci sono utility nei ports:
 - **portupgrade**
 - **pkgdb**
- pkgng include la gestione di aggiornamenti (**pkg upgrade**)
- pkg_* e pkgng NON sono intescambiabili (formato database)
- Altre utility su **ports-mgmt/**
- <http://www.freshports.org>

Rete - 1

- Gestione UNIX classica della rete
- Ifconfig permette la gestione di:
 - Indirizzi IPv4, IPv6
 - alias
 - Vlan
- Impostazioni livello 3 ma anche livello 2 (fisico):
 - configurazione wifi 802.11
- route permette la configurazione instradamento
- netstat diagnostica livello 3

Rete - 2

- gateway_enable in rc.conf per fare forwarding tra interfacce
 - uguale a `# sysctl net.inet.ip.forwarding=1`
- routed demone che offre RIP
- mrouted per routing multicast
- libreria libalias offre il supporto NAT
- PPP ha due implementazioni:
 - User ppp, che utilizza interfaccia tun del kernel
 - Kernel ppp
- La prima e` la piu` usata (man ppp)

Firewall

- 3 implementazioni principali:
 - La classica IPFilter
 - Ipfilterscript
(<http://thebeastie.org/bsd/index.html>)
 - La portata pf (da OpenBSD)
 - altq, etc...
 - La nativa ipfw2
 - ipdivert, natd (via libalias)
 - dumynet
 - intuitiva e ricca di funzionalita`

netgraph, jail

- Sistema flessibile e modulare per implementare oggetti kernel che operano su funzioni di rete
- Stack bluetooth (OBEX, serial, ppp, ...)
- ng_*

- Le jail sono dei meccanismi di chroot isolati, accessibili solo da rete
- Isolare servizi di rete
- Creare una copia della parte di filesystem da utilizzare
- Segnali isolati

Il boot

- Il boot viene eseguito in piu` passi
- boot0
 - Bootloader in 512 byte che riconosce filesystem delle partizioni
- boot1
 - nel primo settore della partizione
- boot2
 - Carica loader oppure direttamente il kernel, puo` leggere il filesystem
- /boot/loader
 - Carica il kernel, eventuali moduli, con opzioni

Il boot - 2

- Configurazioni per loader: moduli, impostazioni moduli, impostazioni loader
 - `/boot/loader.conf`
- Configurazioni per device non PNP (irq, etc)
 - `/boot/device.hints`
- Variabili kernel caricate da loader o a mano:
 - `kenv`

Gestione dischi

- Infrastruttura d
- Raid
- Volumi logici
- ZFS
- Cifratura
- Geom
- gbde
- geli

Dischi ottici

- Acd
- cd
- cdcontrol
- Masterizzare
- Burncd
- atapicam

ACL

- chflags
 - Campi sui file oltre ai classici rwx
- Securelevel
 - Livelli di esecuzione del sistema che introducono limiti
- TrustedBSD MAC Framework
 - www.trustedbsd.org

Compilazione sistema

- `/etc/make.conf`
- Prima si compila il sistema base ed il kernel
- Poi si installa il kernel e si riavvia
- Quindi si installa il sistema base
- Si fa` il merge delle configurazioni (**mergemaster**)
- Il file da leggere **categoricamente** e` **`/usr/src/UPDATING`**
- Tutti i moduli possono essere inclusi nel kernel
- Tipicamente sempre compilati
- C`e` un file di configurazione totale, il default e` **`/usr/src/sys/ARCH/conf/GENERIC`**

recovery

- Bootloader, in due parti, boot0 e boot2 leggono il filesystem
- Copia del vecchio kernel in kernel.old
- **/rescue** nella root
- Recovery con livefs
- livecd

livecd

- Il livefs e` un disso bootonly con una console e comndi vari di emergenza, non e` grafico
- Esistono soluzioni per utilizzi particolari third party:
 - Freesbie, grafico ad uso desktop, sviluppato da un gruppo di italiani
 - Gli script che si utilizzano per creare il livecd e` disponibile nei ports: ognuno puo` farsi il suo
 - Frenzie, per diagnostiche di rete, russo
 - "Livecd", Brasile
 - firewall (per cd e flash): m0n0wall, pfSense

aiuto!

- Handbook ufficiale
- Book specifici ufficiali (porter, developer, ...)
- I man sono molto dettagliati e coprono anche aspetti specificatamente "how-to" (man hier, man tuning)
 - Online
 - `/usr/share/doc/`
- Esempi di configurazioni
 - `/usr/share/example`
- Archivi delle diverse mailing lists
- IRC su efnet (inglese)
- GUFi italia

domande?