

The Power To Serve

Wprowadzenie do FreeBSD

Tomasz Wendlandt
<juggler@7thguard.net>



Historia BSD

- **1969:** Ken Thompson pisze pierwszą wersję systemu UNIX dla komputera PDP-7
- **1976:** Bell Laboratories udostępnia Unix Time-Sharing System, Sixth Edition(V6),system ten trafia do różnych uczelni m.in. do University of California w Berkley
- **1977:** Pojawia się pierwsza wersja **Berkley Software Distribution – 1BSD**
- **1978:** 2BSD
- **1979:** 3BSD (udoskonalenie VM i stronicowanie pamięci)
- **1982:** rodzi się juggler 😊
- **1992:** publikacja kodu BSD, co nie podobało się UNIX System Laboratories, podległej AT&T

Historia BSD

- **1993: FreeBSD 1.0** (oparty o 386BSD)
- **1980 – 1994: powstało najwięcej udoskonaleń systemu**
- **1994:** zahamowanie rozwoju v1.0 i oparcie się na BSD4.4-lite2 (uważany za pierwowzór obecnego systemu) tak powstaje FreeBSD 2.0
- **2000:** FreeBSD 4.0
- **2003:** FreeBSD 5.0
- **2004:** FreeBSD 5.3
- **2005:** FreeBSD 5.4

Historia BSD

- **NetBSD** (Of course it runs NetBSD)
 - <http://www.NetBSD.org/>
- **OpenBSD** (Secure by Default)
 - <http://www.OpenBSD.org/>
- **DragonFly BSD**
 - <http://www.DragonFlyBSD.org/>

Licencja

- Początek w 1992 roku za sprawą publikacji kodu
- Trzy główne założenia:
 - nie podszywać się pod autora programu
 - nie winić autora jeżeli program przestanie działać
 - nie wolno używać naszej nazwy do promocji własnych produktów

Informacja we własnym produkcie o wykorzystaniu własności UC.

A obecnie?

Charakterystyka

- System UNIX-like (dlaczego ?)
- Architektura: i386, alpha, sparc, amd64, ia64 i inne...
- Założenia projektu:
 - Jakość kodu (uważany za wzorcowy stos TCP/IP, VM, UFS)
 - Optymalizacja (wykres)
 - Bezpieczeństwo (Jail, TCP Wrappers, securelevels)
 - Rozwój
- W pełni otwarty model projektu (rozwijany przez 2 grupy twórców: committers i contributors)
 - 1): ~500 src committers, port committers, doc committers
 - 2): „tysiące”

Charakterystyka

- Core Team
- Release Engineering Team
- Security Officer Team
- Documentation Engineering Team
- Port Management Team
- Donations Team

Zgodność jądra z binarkami Linuksa, SCO Unix, *BSD,
Solaris

Natywny system plików FFS zwany też UFS (soft updates)

Soft updates

options

SOFTUPDATES

shutdown now

umount /partycja

tuneefs -n enable /partycja

mount /partycja

exit

Instalacja

- Min. Wymagania
- Instalacja base systemu zajmuje ~130MB
- Sposoby instalacji:
 - CD/DVD
 - Partycja DOS
 - NFS, FTP (również passive mode), HTTP
 - PPP, SLIP, PLIP, Ethernet
 - Dyskietki 😊

Instalacja

Program uruchomieniowy

Program instalacyjny

/stand/sysinstall

Urządzenia

węzeł urządzeń

/dev/fd*

dyskietka

/dev/acd*

CDROM IDE

/dev/cd*

CDROM SCSI

/dev/ad*

HDD IDE

/dev/da*

HDD SCSI

Urządzenia

- `/dev/ad0`, `/dev/ad1`, `/dev/ad2` ...
- `/dev/ad0s1` plastry (slices)
- `/dev/ad0s1b`

Nazewnictwo

CURRENT ----> STABLE

```
    |||
    ,-' | `-.
    ,-' | `-> RELEASE x.x.x
    |   `-> RELEASE z.z
    `-> RELEASE y.y
```

Maciej Korzeń <eaquer@ceti.pl>

Nazewnictwo

- etykieta wydanie
- . FreeBSD-CURRENT
- RELENG_5 FreeBSD 5-STABLE
- RELENG_4 FreeBSD 4-STABLE
- RELENG_5_3 FreeBSD 5.3-RELEASE
- RELENG_4_11 FreeBSD 4.11-RELEASE

Porty vs. Pakiety

- Instalacja na dwa sposoby
- 1): **Porty (/usr/ports)**
- # cd /usr/ports
- # make search name=hot-babe ;-)
- # make search key=p2p
- # cd net/skype
- # make install clean
- # make readmes ---> README.html
- <http://www.FreeBSD.org/cgi/ports.cgi>
- **/usr/ports/INDEX-5**

Porty

```
# cd /usr/ports/mail/postfix
```

```
# ls
```

```
Makefile
```

```
distinfo
```

```
pkg-descr
```

```
pkg-install
```

```
pkg-message
```

```
pkg-plist
```

```
files/
```

```
scripts/
```

Porty

Etapy instalacji

make fetch

make checksum

make depends

make extract

make patch

make configure

make build

make install (po instalacji są traktowane w systemie jak pakiety)

Zależność poszczególnych etapów instalacji od siebie oraz
możliwość optymalizacja i modyfikacja

Wniosek: system portów jest bardzo inteligentny 😊

Pakiety

- `# cd /usr/ports/deskutils/hot-babe`
- `# make package`

- `# pkg_add {-r } {nazwa}-6.1.tbz`
- `# pkg_delete`
- `# pkg_info`
- `# pkg_which`

- Ewidencja (portów i pakietów) w **`/var/db/pkg`**

Aktualizacja systemu

- `# cvsup -g -L 2 supfile`
- `*default host=cvsup.pl.FreeBSD.org`
- `*default base=/usr`
- `*default prefix=/usr`
- `*default release=cvs tag=RELENG_5_3`
- `*default delete use-rel-suffix`
- `*default compress`
- `src-all`

Aktualizacja systemu

- `/usr/sup/refuse`
- `cvsup -L 2 -g supfile -i sys`

- `/usr/src/UPDATING`
- `# cd /usr/src`
- `# make buildworld`
- `# make installworld`
- `# mergemaster`

- `# cd /usr/src/sys/i386/conf`
- `# config KERNEL`
- `# cd ../compile/KERNEL`
- `# make depend all install`
- `# reboot`

Aktualizacja portów

- `# cvsup -g -L 2 ports-supfile`
- `*default host=cvsup.pl.FreeBSD.org`
- `*default base=/usr`
- `*default prefix=/usr`
- `*default release=cvs tag=.`
- `*default delete use-rel-suffix`
- `*default compress`
- `ports-all`

Aktualizacja portów

- # portsdb -Uu
- # portversion -v -l „<„
- # portupgrade -arR

Filtry Pakietów

- ipfw (dummynet)
- options IPFWALL
- options IPFWALL_VERBOSE
- options IPFWALL_DEFAULT_TO_ACCEPT
- options DUMMYNET

- ipf
- options IPFILTER
- options IPFILTER_LOG
- options IPFILTER_DEFAULT_BLOCK

Filtry Pakietów

- pf (od 5.X w portach, od 5.3 w base systemie)
- device pf
- device pflog
- device pfsync

- ALTQ i CARP
- options ALTQ
- options ALTQ_CBQ # Class Bases Queuing (CBQ)
- options ALTQ_RED # Random Early Detection (RED)
- options ALTQ_RIO # RED In/Out
- options ALTQ_PRIQ # Priority Queuing (PRIQ)
- options ALTQ_NOPCC # Required for SMP build
- options ALTQ_HFSC # Hierarchical Packet Scheduler (HFSC)

Filtry Pakietów

wan = "r10"

lan = "r11"

webserver = "xx.xx.xx.xx/32"

spoofed = "{ 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 255.255.255.255/32 }"

table <spammers> persist file "/etc/spammers"

scrub in all

block in log all

block in log quick on \$wan from \$spoofed to any

block in log quick on \$wan from <spammers> to any

pass in log-all quick proto tcp from any to \$webserver port = { 21, 22 } flags
S/SA synproxy state

...

CARP

- zastosowanie
- **Hot Standby Router Protocol:**
 - komercyjny
 - przestarzały
- **Virtual Router Redundancy Protocol:**
 - obciążony patentem
- **Common Address Redundancy Protocol**

CARP

- **zalety:**
- różne platformy sprzętowe
- różne systemy operacyjne
- dowolna wielkość klastra
- IPv4 i IPv6
- darmowy
- otwarty kod

- **wady:**
- problemy z połączeniami VPN

Linki

- <http://www.FreeBSD.org/>
- <http://www.FreeBSD.org/handbook>
- <http://www.freshports.org/>
- <http://www.BSDGuru.org/>
- <http://FreeBSD.hello.pl/>
- <http://FreeBSD.kie.pl/>
- <http://google.pl/>
- <news://pl.comp.os.bsd>

Dziękuję za uwagę

<juggler@7thguard.net>
<http://linux.net.pl/~juggler>

PLUG Poznań 2005r.
<http://poznan.linux.org.pl/>
<plug@poznan.linux.org.pl>