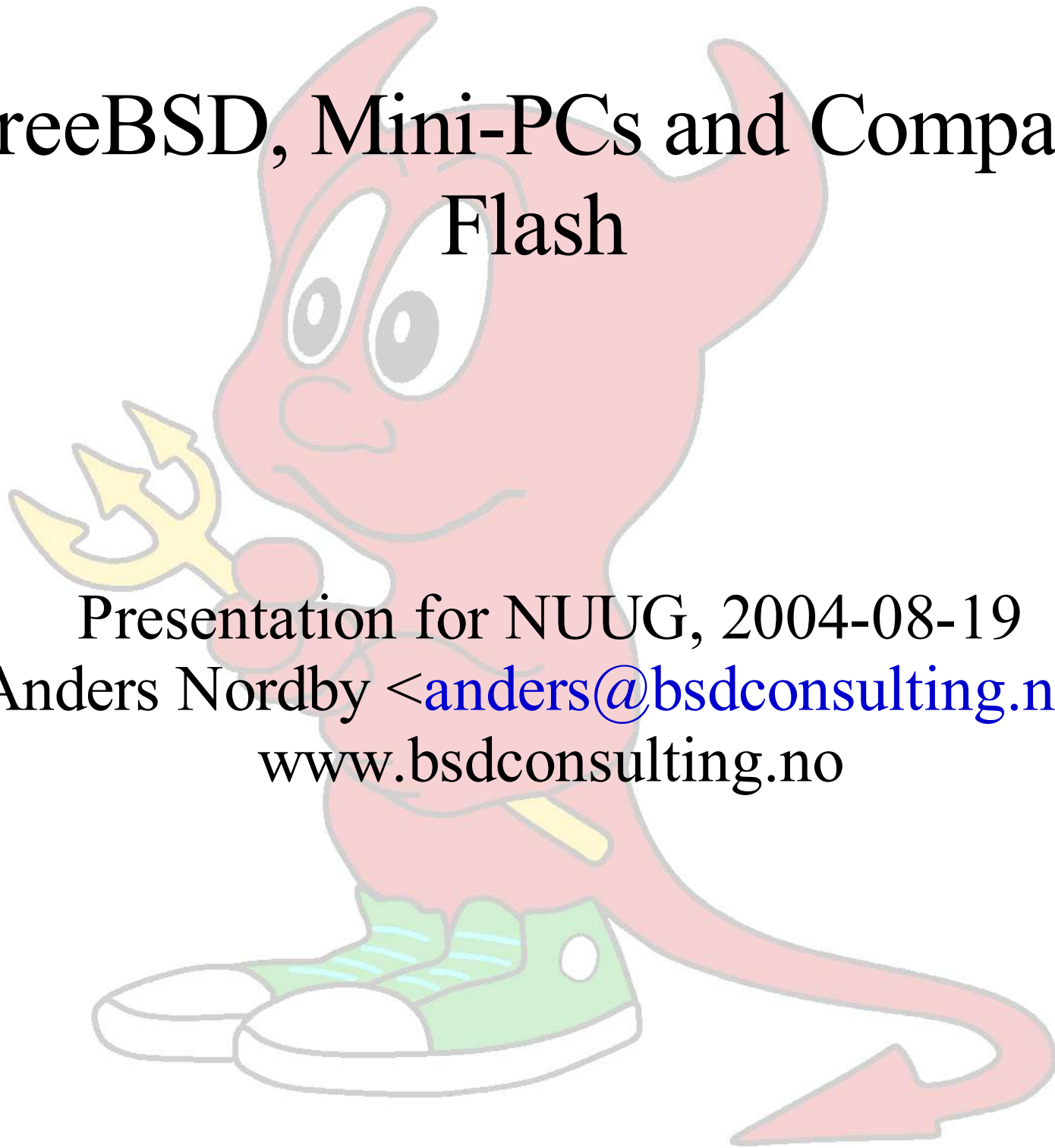


FreeBSD, Mini-PCs and Compact Flash

Presentation for NUUG, 2004-08-19

Anders Nordby <anders@bsdconsulting.no>

www.bsdconsulting.no





Anders Nordby?

- Has worked with UNIX system administration and security since 1999, for companies like Skrivervik Data, Tiscali World Online and now Aftenposten AS.
- Responsible for the NUUG server Nerdhaven and nuug.no Internet services since 1999.
- FreeBSD ports committer, works on expanding the wealth of software available for FreeBSD and its general usability.



A small PC? Welcome, Mini ITX.

- Mini ITX dates back to 1999, when VIA bought Cyrix from National Semiconductor.
- Mini ITX form factor mainboard reference design, the second ITX iteration, was released november 2001. It requires mainboards to be max. 170*170mm.
- VIA makes processors for Mini ITX: C3 and Eden. C3 is faster, Eden typically comes in a fanless configuration. They are both low on power consumption, one of their main features.
- Mini ITX and Via C3/Eden is mainstream, cheap and available through plenty of normal PC shops.



Compact Flash

- “CompactFlash® is a small, removable mass storage device. First introduced in 1994, CompactFlash cards weigh a half ounce and are the size of a matchbook. They provide complete PCMCIA-ATA functionality and compatibility.”
- Has no movable parts, providing better safety for your data and system than any magnetic disk drive (harddrive).
- Requires little power, makes no noise.
- One disadvantage: limited number of write operations per memory cell. Not suitable for swapping! 500000/1000000 write operations normal.



I want a small and noiseless firewall, what hardware do I get?

What I settled for:

- High Green Cupid 3688 Silver cabinet. A small, pretty-looking Mini ITX cabinet with external power-supply (no fan!).
- VIA EPIA CL6000E mainboard with 600 MHz fanless Eden CPU. Integrated dual NIC, perfect for a firewall.
- 128 MB TwinMOS Compact Flash card, 36x. Yields around 1,5 MB per second when writing to it sequentially.
- PC Engines Compact Flash to ATA converter.
- Total price: around 3000 NOK.



FreeBSD?

- A source-oriented UNIX-like operating system for I386, Alpha, AMD64, SPARC64 etc.
- Fully self contained. Rebuildable kernel and OS when installed with source.
- The normal upgrade procedure is to “make world”: build the OS, build the kernel, install them, and upgrade /etc using the included tool mergemaster.
- FreeBSD source code is kept in a central CVS repository, distributed internationally through a network of cvsup mirrors for rapid and bandwidth-friendly synchronization.



Preparing for the build

- Synchronize FreeBSD's src-all cvsup collection (all the source) using the CVS tag RELENG_4 for FreeBSD 4-STABLE. I use FreeBSD 4 for conservative reasons, not wanting to change my CF building environment more than necessary.
- Locate the make variables to make our FreeBSD installation small by exercising make buildworld looking for NO* variables in all Makefiles (grep is your friend).
- On your build system, consider getting a PCMCIA or USB card writer, as you can hot-plug CF cards to them (you can not with a CF to ATA adapter).



Kernel configuration

- Copy `/usr/src/sys/i386/conf/GENERIC` to a new file, e.g. `CFBSD`, and modify it.
- Disable IPv6 (and its pseudo-device companions `gif` and `faith`) if you don't use it.
- Add `IPFILTER`, `IPFILTER_LOG` and `IPFILTER_DEFAULT_BLOCK` for a IP Filter based firewall.
- Add options `NO_SWAPPING`, to completely disable swapping.



What not to build? (make.conf)

My choice of what to not build (not needed for my firewall) is:

```
NOPROFILE=yes  NOMAN=yes  NOGAMES=yes  NO_TCSH=yes  
NO_CVS=yes  NO_BIND=yes  NO_GDB=yes  NO_CXX=yes  NO_OBJC=yes  
NO_CPP=yes  NO_GCOV=yes  NO_I4B=yes  NO_LPR=yes  
NO_FORTRAN=yes  NOSHARE=yes  NOINFO=yes  NOLIBC_R=yes  
NO_INET6=yes  NORADIUS=yes  NOATM=yes  NO_MODULES=yes
```

This keeps the installation around 80-85 MB.

`NO_MODULES` drops kernel modules.



Building CFBSD

1) Go to /usr/src.

2) Build the world:

```
make __MAKE_CONF=/path/cfbsd-make.conf  
MAKEOBJDIRPREFIX=/space/cfobj buildworld
```

3) Build the kernel:

```
make KERNCONF=CFBSD __MAKE_CONF=/path/cfbsd-make.conf  
MAKEOBJDIRPREFIX=/space/cfobj buildkernel
```

__MAKE_CONF overrides which `make.conf` to use
(separate `make.conf` is practical),
MAKEOBJDIRPREFIX sets a object destination dir
different from the system one (`/usr/obj`).



Installing CFBSD

- 1) Go to `/usr/src`.
- 2) Install the directory hierarchy:
`make XXX hierarchy`
- 3) Install the FreeBSD world (userland):
`make XXX installworld`
- 4) Install the `/etc` configuration directory:
`cd etc; make XXX distribution`
- 5) Install the custom kernel:
`cd ../; make XXX KERNCONF=CFBSD installkernel`

XXX is `DESTDIR=/space/cfroot`

__MAKE_CONF=`/path/cfbsd-make.conf`
MAKEOBJDIRPREFIX=`/space/cfobj`.



Post-installation

Relative to your temporary installation directory:

1) Add pre-made config-files, extra binaries etc..

2) Edit the central configuration file `/etc/rc.conf`:

- `update_motd="NO"` (/ is read-only)
- `varsize="<512-byte blocks>"` (/var filesystem size)
- `diskless_mount="/etc/rc.diskless2"` (what script to use for diskless mounting)
- `root_rw_mount="NO"` (mount root read-only)
- `weekly_whatism_enable="NO"` (do not update whatism database)



Post-installation (2)

3) `/etc/fstab`: Add root file-system with device `/dev/ad0s1a` (`ad2s1a` if on the secondary IDE controller) and mount options `ro` (read-only). Do NOT add a swap device. Keep `/proc` not mounted by default (`noauto`) for security.

4) Copy the `/usr/share` files you want from a system running the same release/OS version:

- `syscons/fonts/iso*` (console fonts)
- `syscons/keymaps/norwegian.iso`
- `syscons/screenmaps/iso*` (screenmap)



Post-installation (3)

- 4) Add `/boot.config` with contents `-Dh` for a dual serial/keyboard console, edit `/etc/ttys` to have a `ttyd0` (COM1) console login service enabled.
- 5) Set the root password.
- 6) Add a root mail alias, run `newaliases`.
- 7) Patch `rc.diskless2` to not run `newaliases`, modify the `mount_md` subroutine to mount memory filesystems with mount option “noexec”.



Make the image

We want to write the CF card sequentially to write as few times per memory cell as possible, so we need to generate a binary image. For a 128 MB TwinMOS card, I need a fixed geometry due to USB/IDE CF card reader incompatibilities. Image size (for dd) equals sectors reported by FreeBSD * 512 (block size).

Variables:

```
fdiskconf=fdisk.tmos128; size=251904
```

Geometry (fdisk option file - 984 cylinders/16 heads/16 sectors):

```
g c984 h16 s16
p 1 165 16 251888
a 1
```



Make the image (2)

Create the image file, configure it as a virtual node
(vn) device :

```
dd if=/dev/zero of=cfbsd.img bs=512 count=$size  
vnconfig -s labels -c vn0 cfbsd.img
```

Partition/label/make filesystem:

```
fdisk -f $fdiskconf -v vn0  
disklabel -rw vn0s1 auto  
disklabel vn0s1 >dlb  
egrep -v '^ [a-h]: ' dlb >dlbnew  
egrep '^ c: ' dlb | sed -E "s|^ c: | a: |;s|unused|  
4.2BSD|" >>dlbnew  
egrep '^ c: ' dlb >>dlbnew  
disklabel -RB vn0s1 dlbnew  
newfs -i 1000 -m 0 /dev/vn0s1a
```




Populate and write the image

Mount the virtual filesystem created:

```
mkdir /cfbsd; mount /dev/vn0 /cfbsd
```

Copy everything over:

```
tar -cpf - -C /space/cfroot . | tar -xpf - -C /cfbsd
```

Unmount and unconfigure the vn device:

```
umount /cfbsd; vnconfig -u vn0
```

Write the image file to your SCSI (USB mass storage devices pop up like SCSI devices in FreeBSD) device:

```
dd if=cfbsd.img of=/dev/rda0 bs=8k
```



Voila!

Insert your Compact Flash card into the CF to ATA adapter, boot your system and enjoy!

Your PC will recognize the CF card as a plain IDE disk, and booting will work with no BIOS or other changes if you properly bootstrapped and partitioned it.



Links

Mini ITX: www.mini-itx.com

High Green cabinets: www.highgreen.nl

VIA Technologies, Inc: www.viavpsd.com

TwinMOS: www.twinmos.com

Compact Flash: www.compactflash.org

PC Engines: www.pcengines.ch

FreeBSD: www.freebsd.org

IP Filter firewall: www.ipfilter.org

CVSUP: www.cvsup.org

FreeBSD CVSUP information:

<http://www.freebsd.org/doc/en/books/handbook/cvsup.html>



CFBSD

For a copy of this presentation, go to www.bsdconsulting.no. Scripts for building/installing/etc. may also be provided at this URL when ready.

Background picture courtesy of Poul-Henning Kamp, drawn using xfig.