

Exercise 1 – FreeBSD Installation

Announced Date: 2004/9/20

Due Date: 2004/10/3

Bonus Due Date: 2004/9/27

Outline

- > FreeBSD version
- > Installing FreeBSD
- > Update source and make world
- > Rebuild kernel

FreeBSD branches

> Two parallel development branches:

— *-CURRENT*

- Latest working sources for FreeBSD
- Latest release version: 5.2.1 in Feb. 2004.

— *-STABLE*

- Receive only well-tested bug fixes and other small incremental enhancement
- Latest release version: 4.10 May. 2004.

FreeBSD version

> A.B.C – Type

- A: major version Number
- B: minor version Number
- C: slight patch version number
- Type: version type
 - **SNAP**
 - **ALPHA** 、 **BETA** 、 **GAMMA**
 - **RELEASE**
 - **RELENG**
 - **STABLE**
 - **CURRENT**

Snapshot → { Alpha
Beta → Release → Releng → Stable
Gamma

FreeBSD view of Disk (1)

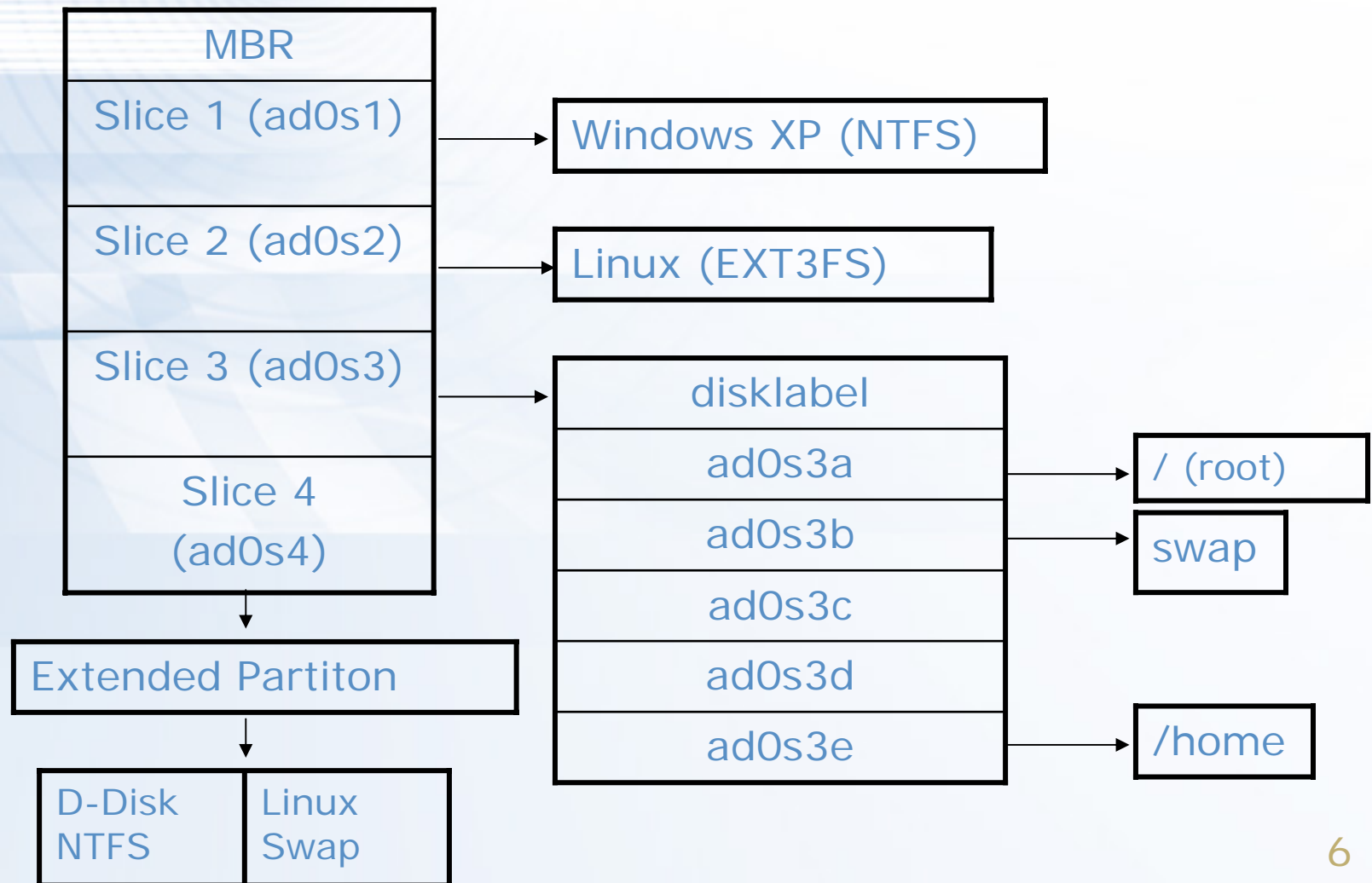
> What is the meaning of ad0s1e

- Disk name
 - **IDE:** **ad**
 - **SCSI:** **da**
- Slice is equal to the partition of common use
 - **Primary partition: s1 ~ s4**
 - **Extended partition: s5 ~ s8**
- Label in each slice
 - **a: root partition**
 - **b: swap**
 - **c: entire disk**
 - **d: entire partition**
 - **efgh: /usr, /home, ...**

MBR	
Slice 1 (/dev/ad0s1)	
Slice 2 (/dev/ad0s2)	
Slice 3 (/dev/ad0s3)	
Slice 4 (/dev/ad0s4)	
Slice 5 /dev/ad0s5	Slice 6 /dev/ad0s6

FreeBSD view of Disk (2)

An Example



Installing FreeBSD

> Steps

1. Knowing your hardware
2. Obtaining installation file
3. Booting from CD
4. Kernel Configuration Menu
5. sysinstall main menu
6. Custom Installation Options
 1. **Partition**
 2. **Label**
 3. **Distribution**
 4. **Media**
 5. **Commit**
7. Post Installation

Installing FreeBSD –

1. knowing your hardware

- > CPU
 - 32bit or 64bit, Xeon, Intel 、 AMD or other brand
- > RAM
 - Size
- > HD
 - Size, amount, SCSI or IDE
- > VGA
 - Brand, ram size
- > Sound
 - Brand
- > Network Interface Card
 - Brand
 - IP 、 Netmask 、 default gateway 、 Hostname 、 DNS
- > Other Special device

Installing FreeBSD –

2. Obtaining installation file

> FreeBSD installation CD

- <ftp://freebsd.csie.nctu.edu.tw/pub/ISO-IMAGES-i386/4.10/4.10-RELEASE-i386-disc1.iso>
- Burn!

> Boot Floppy Image

- <ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/4.10-RELEASE/floppies/>
- <ftp://ftp.freebsd.org/pub/FreeBSD/tools/fdimage.exe>
- C:\fdimage.exe kern.flp a:\

Installing FreeBSD –

3. Booting from CD

```
Uncompressing ... done

BTX loader 1.00  BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
BIOS drive B: is disk1
BIOS drive C: is disk2
BIOS 639kB/129984kB available memory

FreeBSD/i386 bootstrap loader, Revision 0.8
(root@freebsd-stable.sentex.ca, Thu Apr  3 08:41:45 GMT 2003)
/kernel text=0x280131 data=0x33018+0x3311c !
-
Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [kernel] in 4 seconds...
```

Installing FreeBSD –

4. Kernel Configuration Menu

- > Install first and configure kernel later
 - Choose “Skip kernel configuration and continue with installation”
 - Then it will probe the devices in your system

```
Kernel Configuration Menu

Skip kernel configuration and continue with installation
Start kernel configuration in full-screen visual mode
Start kernel configuration in CLI mode

Here you have the chance to go into kernel configuration mode, making
any changes which may be necessary to properly adjust the kernel to
match your hardware configuration.

If you are installing FreeBSD for the first time, select Visual Mode
(press Down-Arrow then ENTER).

If you need to do more specialized kernel configuration and are an
experienced FreeBSD user, select CLI mode.

If you are certain that you do not need to configure your kernel
then simply press ENTER or Q now.
```

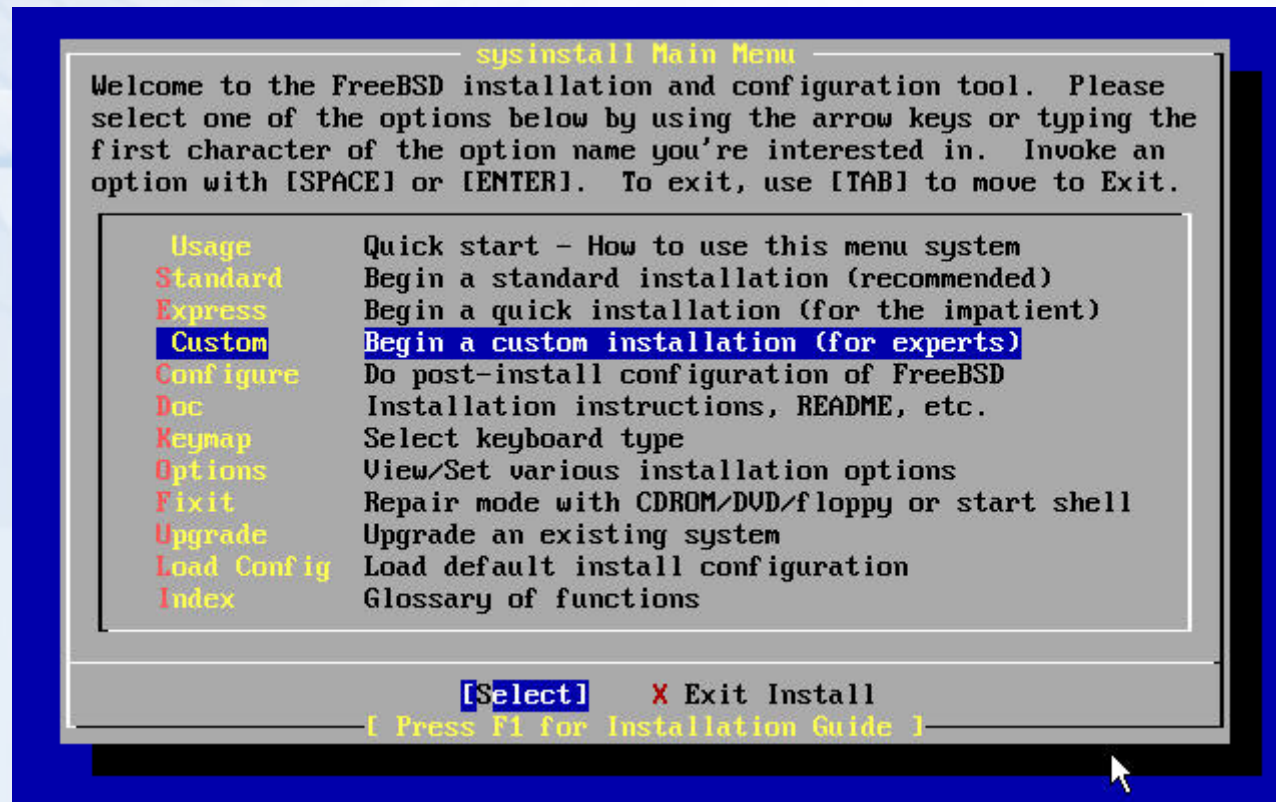
Installing FreeBSD –

5. sysinstall Main Menu

- You can press “Scroll Lock” key to see probe results.

> sysinstall Main Menu

- Choose “Custom”

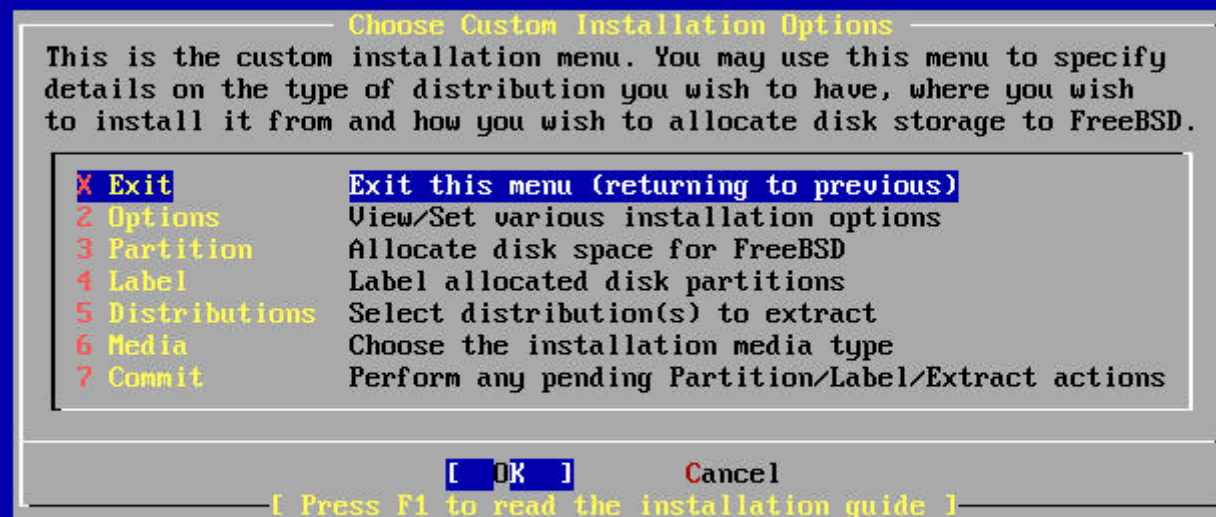


Installing FreeBSD –

6. Custom Installation Options

> 4 major steps

- Partition and label your disk
- Choose what to install and how to install
- Commit



Installing FreeBSD –

6. Custom Installation – partition (1)

> Create slice and choose boot manager

- Press “C” to create a new slice or press “A” to use entire disk
- Press “S” to toggle ad0s1 as bootable (we will put / on this slice)
- Press “Q” to next step (Select Boot Manager)

```
Disk name: ad0 FDISK Partition Editor
DISK Geometry: 33288 cyls/16 heads/63 sectors = 33554304 sectors (16383MB)

Offset      Size(MB)      End      Name  PType  Desc  Subtype  Flags
    0         0         62      -     6    unused     0
    63       8191    16777151  ad0s1  3    freebsd   165     CA
16777152     8191    33554303  ad0s2  3    freebsd   165

The following commands are supported (in upper or lower case):

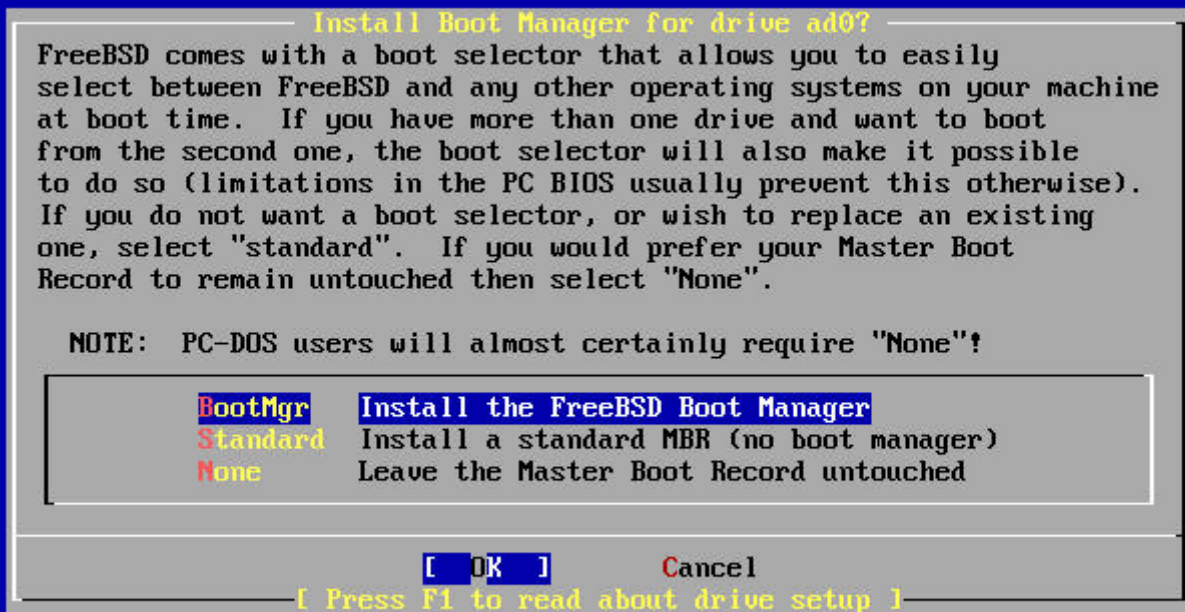
A = Use Entire Disk  G = set Drive Geometry  C = Create Slice  F = 'DD' mode
D = Delete Slice    Z = Toggle Size Units  S = Set Bootable  i = Wizard m.
T = Change Type     U = Undo All Changes  Q = Finish

Use F1 or ? to get more help, arrow keys to select.
```


Installing FreeBSD –

6. Custom Installation – partition (2)

- Select “BootMgr” for ad0
- After press OK, it will back to Custom Installation Options menu



BootMgr

- multiple OS

Standard

- single OS

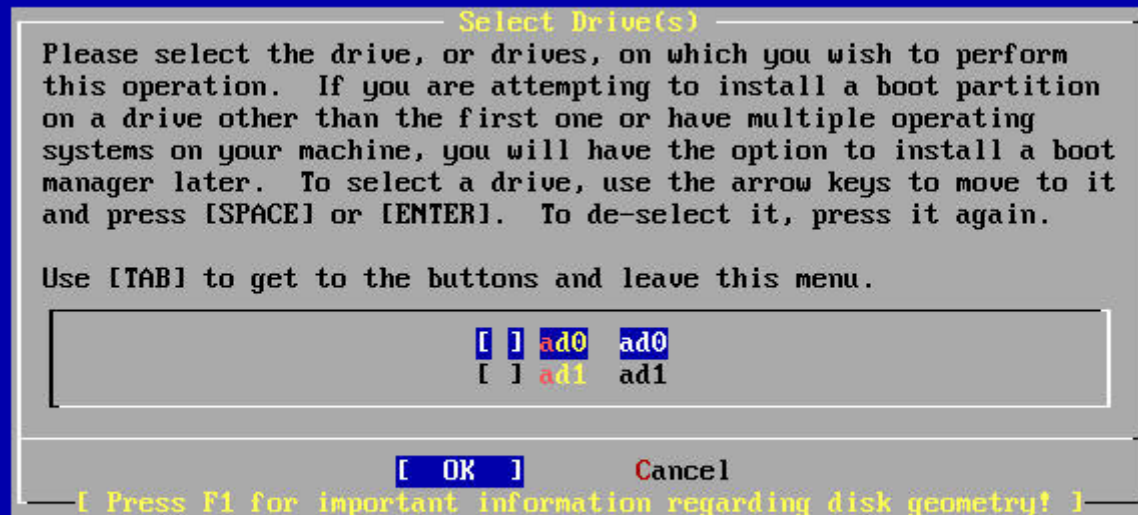
None

- Other BM

Installing FreeBSD –

6. Custom Installation – partition (3)

- > If you have more than one disk...
 - You can choose whether to partition it.
 - Install “BootMgr” for first disk and “none” for rest ones



Installing FreeBSD –

6. Custom Installation – Label (1)

> Disklabel Editor

- Move blue bar to select slice
- Press “C” to create disk label
 - **/ , swap, /home**
 - > Specify size
 - > Choose type (either swap or FS)
 - > Specify mount point
- Press “S” to toggle SoftUpdates (async written to disk)
- Press “Q” to next step (back to custom installation options menu)

Installing FreeBSD –

6. Custom Installation – Label (2)

- Create label in ad0 and specify size

```
FreeBSD Disklabel Editor
Disk: ad0      Partition name: ad0s1  Free: 33554241 blocks (16383MB)
Disk: ad1      Partition name: ad1s1  Free: 33554241 blocks (16383MB)

Part  Mount      Size Newfs  Part  Mount      Size Newfs
-----

```

Value Required

Please specify the partition size in blocks or append a trailing G for gigabytes, M for megabytes, or C for cylinders.
33554241 blocks (16383MB) are free.

33554241

[OK] Cancel

```

The following commands are valid here (upper or lower case):
C = Create      D = Delete    M = Mount pt.
N = Newfs Opts  Q = Finish   S = Toggle SoftUpdates
T = Toggle Newfs U = Undo      A = Auto Defaults   R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.
```

Installing FreeBSD –

6. Custom Installation – Label (3)

– Complete disklabel

```
FreeBSD Disklabel Editor

Disk: ad0      Partition name: ad0s1  Free: 0 blocks (0MB)
Disk: ad1      Partition name: ad1s1  Free: 0 blocks (0MB)

Part      Mount      Size Newfs  Part      Mount      Size Newfs
-----
ad0s1b    swap      512MB SWAP
ad0s1a    /          15871MB UFS+S Y
ad1s1e    /home     16383MB UFS+S Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete    M = Mount pt.
N = Newfs Opts  Q = Finish    S = Toggle SoftUpdates
T = Toggle Newfs U = Undo      A = Auto Defaults    R = Delete+Merge

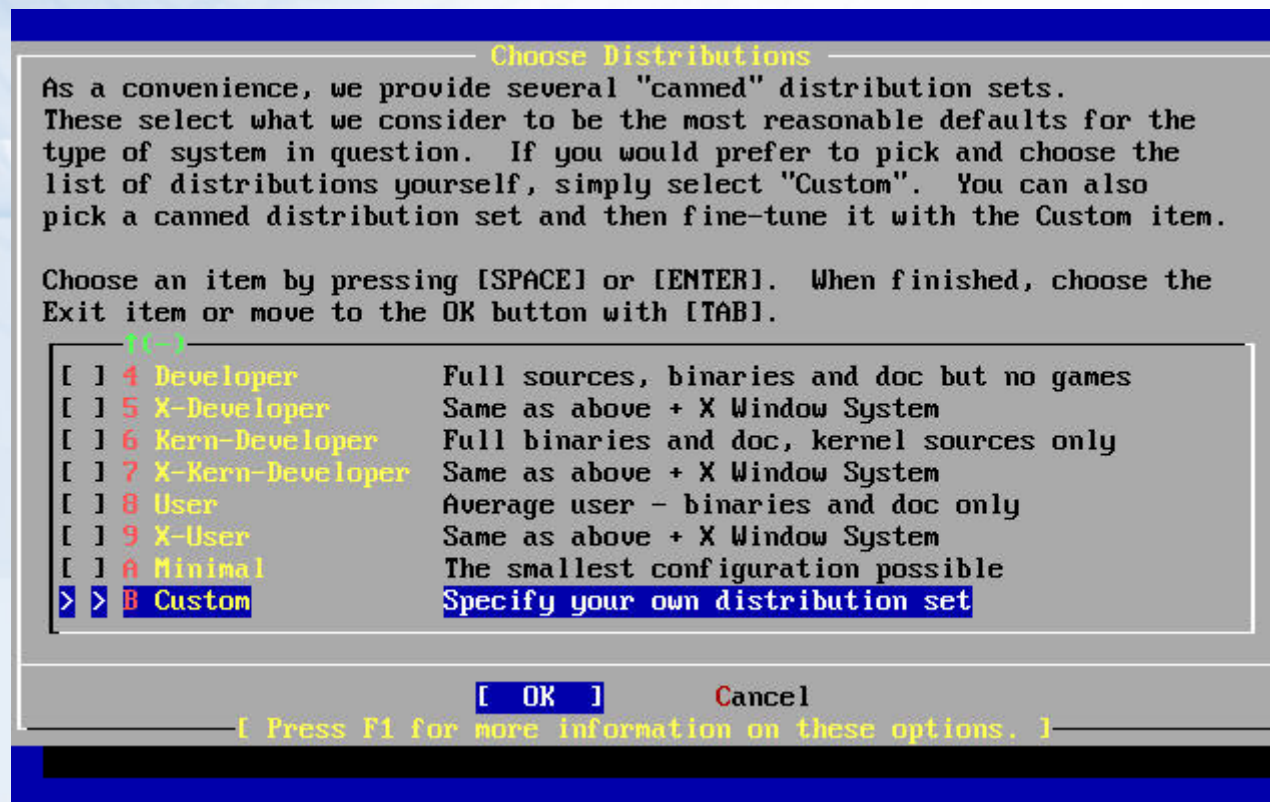
Use F1 or ? to get more help, arrow keys to select.
```


Installing FreeBSD –

6. Custom Installation – distri. (1)

> Choose Distributions Menu

– Choose “Custom”

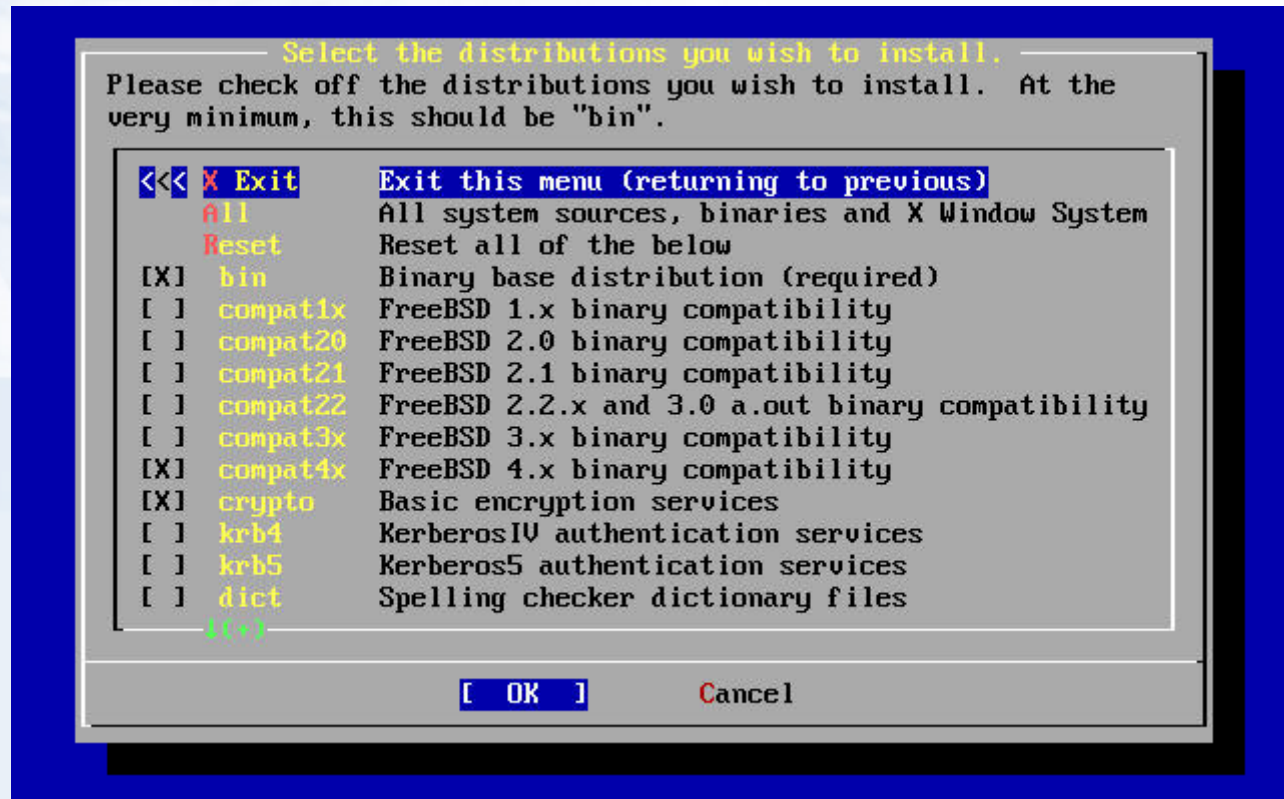


Installing FreeBSD –

6. Custom Installation – distri. (2)

– Select

- **bin** : binary
- **compat4x** : 4.x binary compatibility
- **crypto** : encryption service
- **man**
- **src (all)**
- **ports**



Installing FreeBSD –

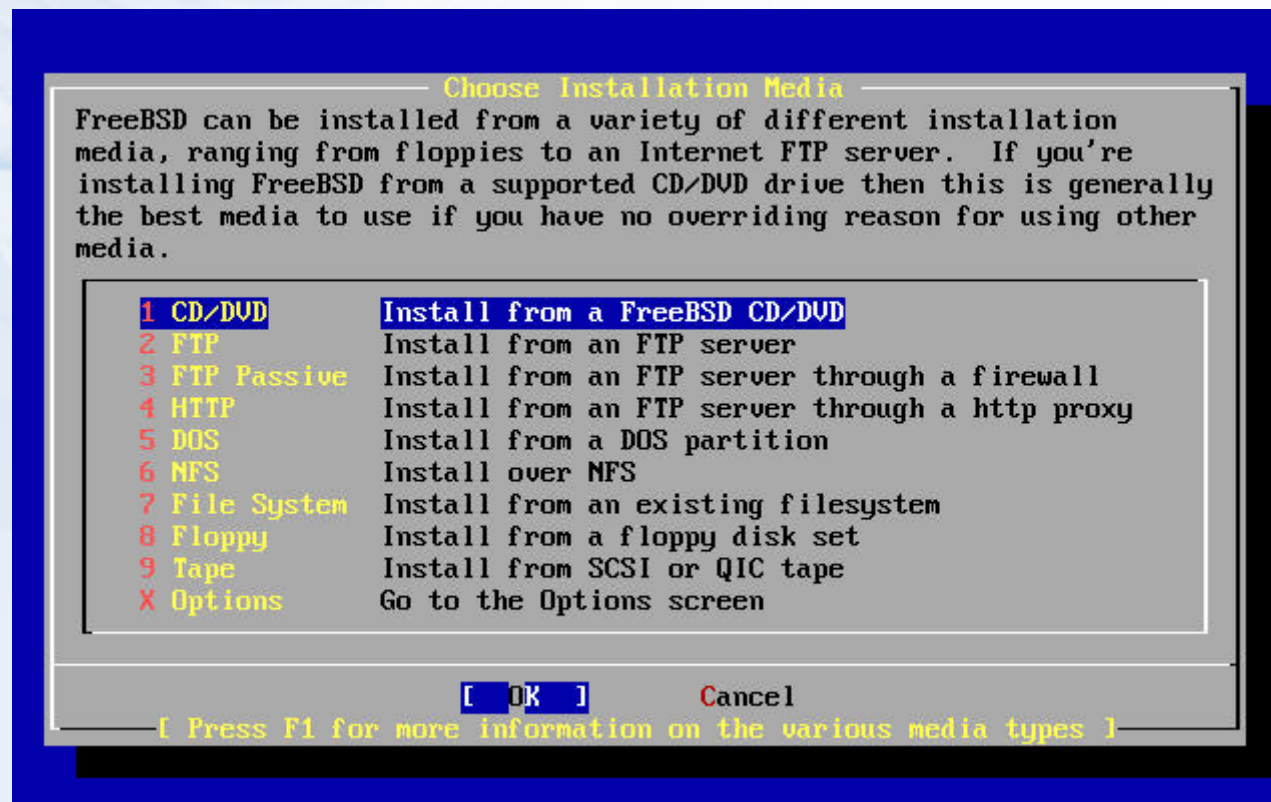
6. Custom Installation – distri. (3)

- Press “OK” and it will return to “Choose Distributions menu”
- Press “OK” again to back to “Custom Installation Options menu”
- Select “Media”

Installing FreeBSD –

6. Custom Installation – Media (1)

- Choose CD/DVD if you have 4.10 Stable CD
- Choose FTP if your NIC is detected
 - Choose FTP Passive if you in private network

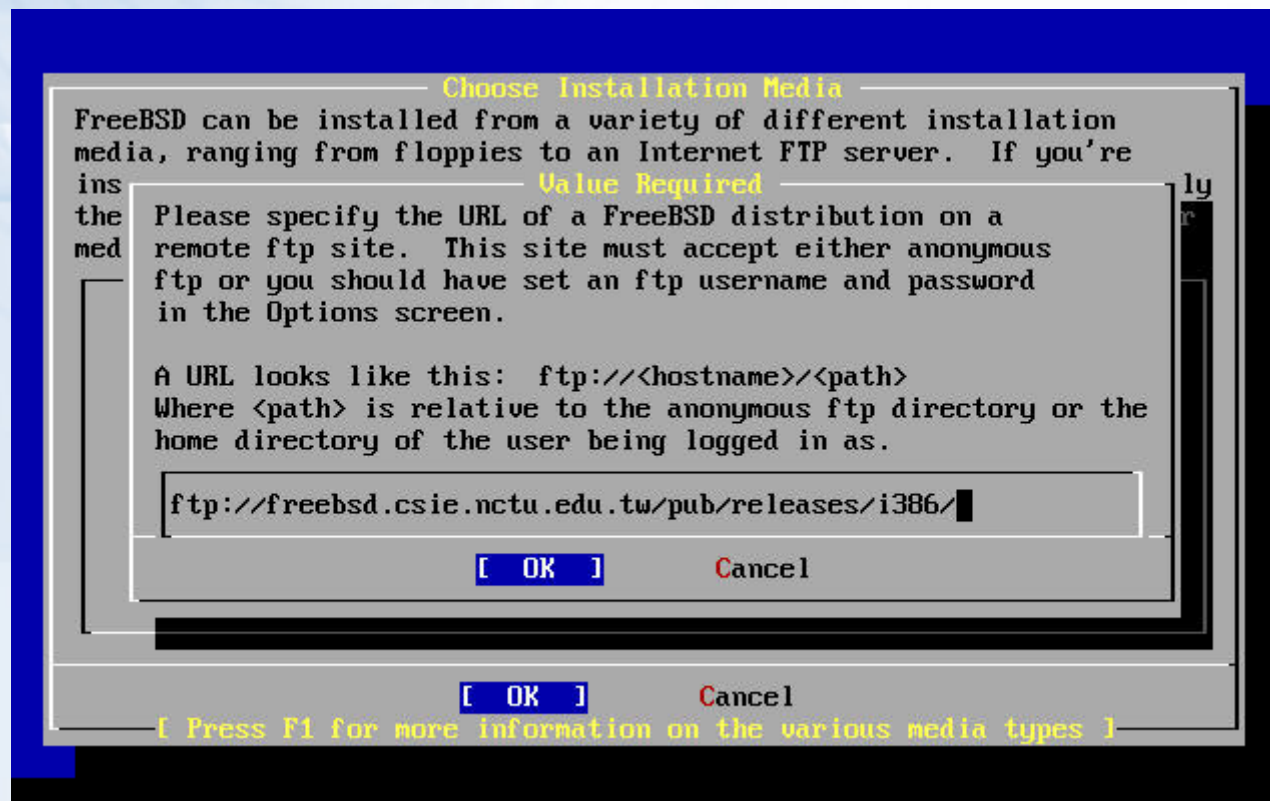


Installing FreeBSD –

6. Custom Installation – Media (2)

> Install through FTP

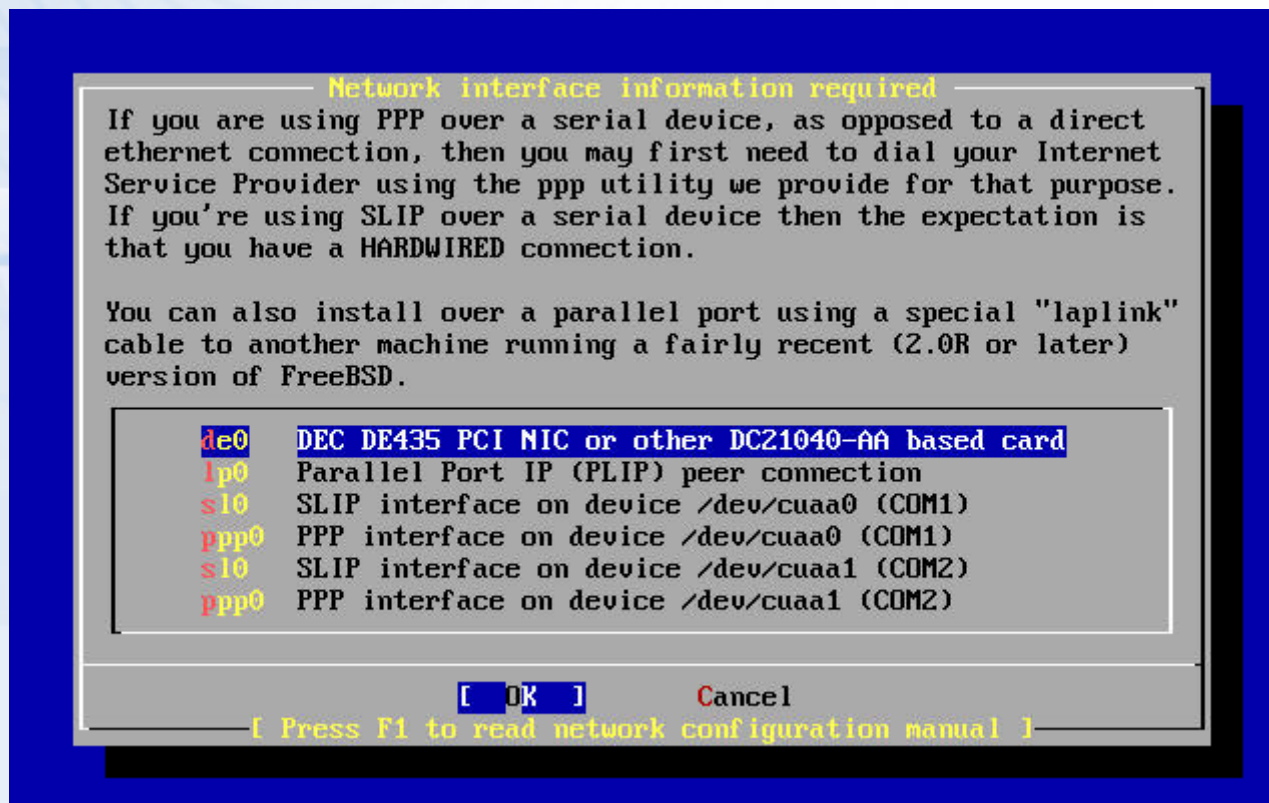
- Specify ftp server and path



Installing FreeBSD –

6. Custom Installation – Media (3)

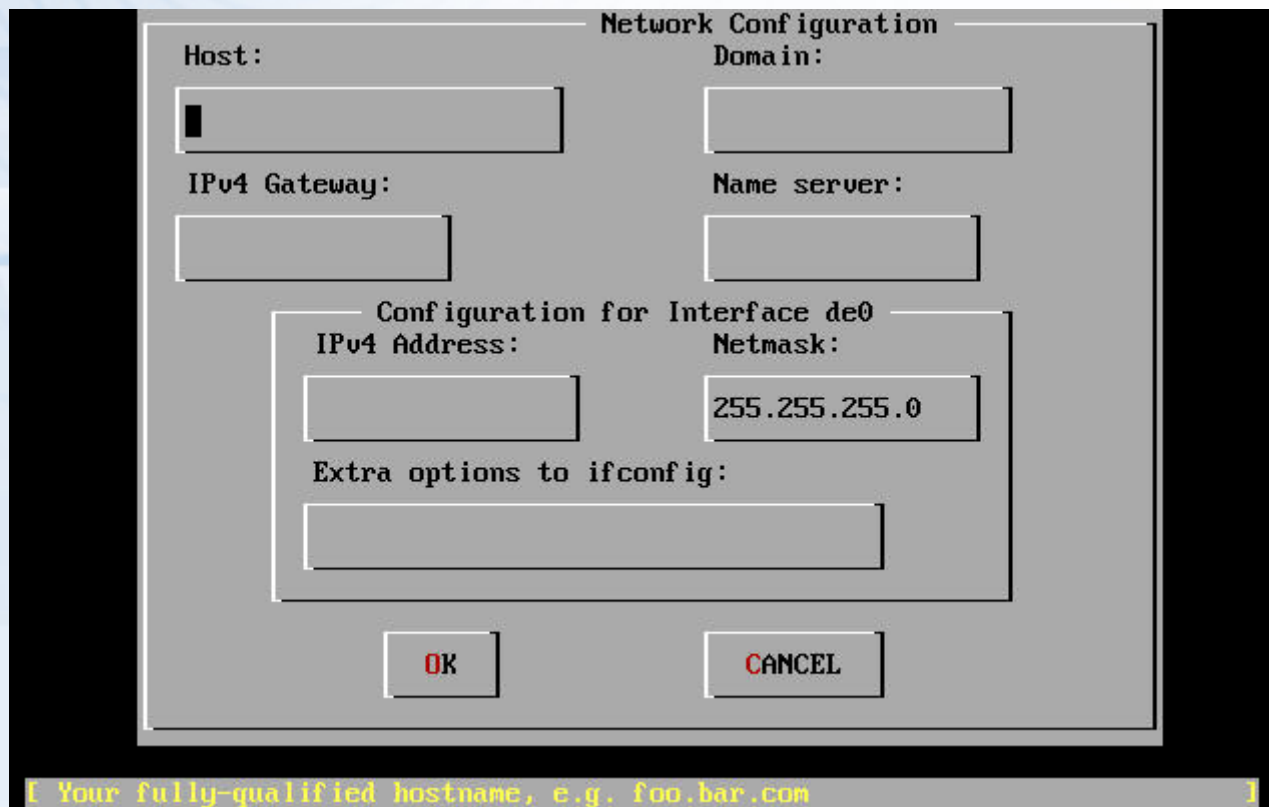
- Select NIC
- IPv6 and DHCP



Installing FreeBSD –

6. Custom Installation – Media (4)

- Specify your IP information
- Press “OK” to next step

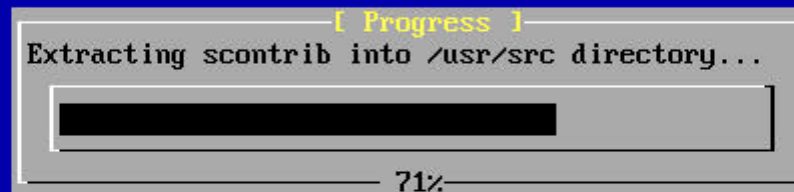


The image shows a 'Network Configuration' dialog box from the FreeBSD installer. It has a title bar 'Network Configuration'. Inside, there are four input fields: 'Host:' (with a cursor), 'Domain:', 'IPv4 Gateway:', and 'Name server:'. Below these is a sub-dialog box titled 'Configuration for Interface de0' which contains 'IPv4 Address:', 'Netmask:' (pre-filled with '255.255.255.0'), and 'Extra options to ifconfig:'. At the bottom are 'OK' and 'CANCEL' buttons. A status bar at the bottom reads: [Your fully-qualified hostname, e.g. foo.bar.com]

Installing FreeBSD –

6. Custom Installation – Commit

- > Start to format disk 、 make file system and install software
- > You can press “Alt + F2” to see the install detail

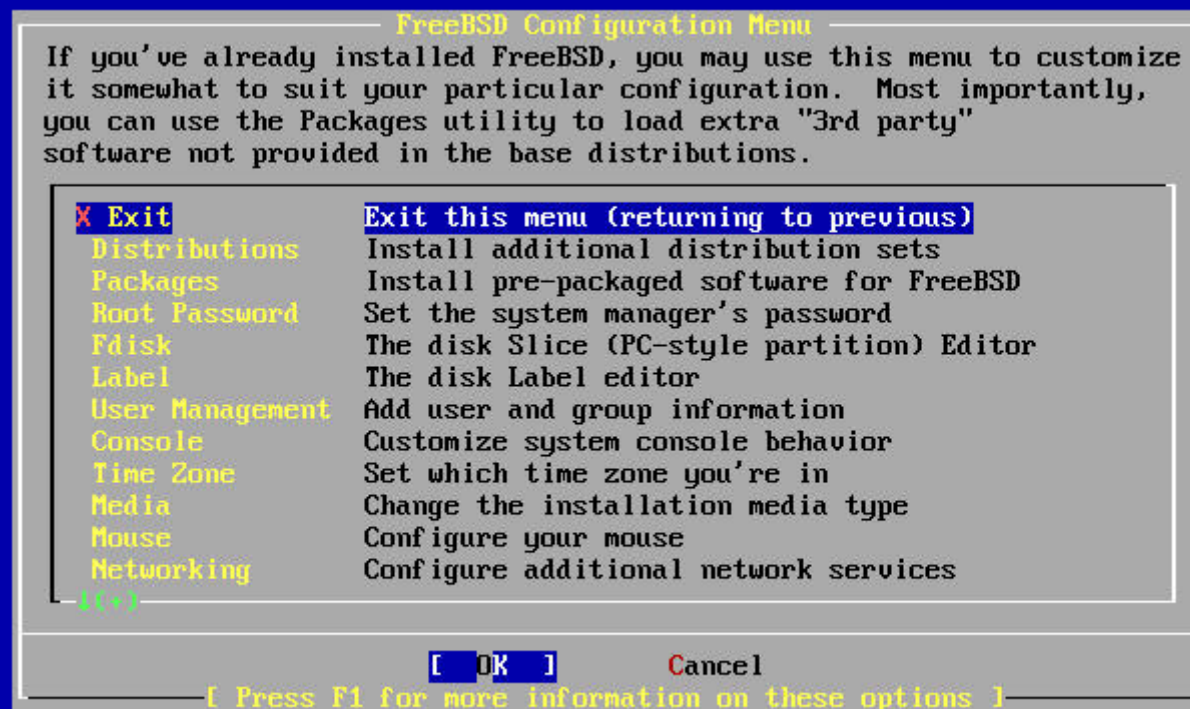
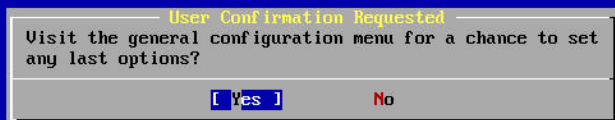
A screenshot of a FreeBSD installation progress window. The window has a title bar that says "[Progress]". The main text inside the window reads "Extracting scontrib into /usr/src directory...". Below this text is a progress bar that is approximately 71% full. At the bottom of the window, the text "71%" is displayed.

[Progress]
Extracting scontrib into /usr/src directory...
71%

25027584 bytes read from scontrib dist, chunk 105 of 146 @ 252.8 KBytes/sec.

Installing FreeBSD –

7. Post Installation (1)



Installing FreeBSD – 7. Post Installation (2)

- > Root Password
- > Time Zone → Asia → Taiwan
- > Mouse → enable



Exercise 1 – FreeBSD build world and kernel

FreeBSD source

- > Maintained in a CVS repository in California
- > We can use CVSup keep our system up-to-date with any FreeBSD mirror sites
 - Install CVSup
 - Edit CVSup supfile
 - Update source using CVSup
 - Make world to build the updated source

CVSup – CVSup Installation

> Install via pkg_add

- Package is pre-compiled application
- % pkg_add <ftp://freebsd.csie.nctu.edu.tw/pub/CVSup/cvsup-16.1e.tgz>
- **pkg_add** package-name
- **pkg_delete** package-name
- **pkg_info** package-name
- All installed package is stored in /var/db/pkg

> The cvsup binary is in /usr/local/bin/cvsup

- You can use “whereis” command to find something

CVSup –

CVSup Configuration file (1)

> Example cvsup supfile

- /usr/share/examples/cvsup/stable-supfile
- /usr/share/examples/cvsup/ports-supfile

> Create your own supfile

- Edit /usr/local/etc/cvsup-src
- Edit /usr/local/etc/cvsup-ports

CVSup –

CVSup Configuration file (2)

/usr/local/etc/cvsup-src

```
*default host=freebsd.csie.nctu.edu.tw
*default base=/usr
*default prefix=/usr
*default delete use-rel-suffix
*default release=cvs tag=RELENG_4
src-all
```

Where to get source
Where to put status file
Where to put source
Allow cvs to delete

CVSup –

CVSup Configuration file (3)

/usr/local/etc/cvsup-ports

*default host=freebsd.csie.nctu.edu.tw

*default base=/usr

*default prefix=/usr

*default delete use-rel-suffix

*default release=cvs tag=.

ports-all

CVSup –

CVSup Configuration file (4)

Or you can put them all together
/usr/local/etc/cvsup-all

*default host=freebsd.csie.nctu.edu.tw

*default base=/usr

*default prefix=/usr

*default delete use-rel-suffix

*default release=cvs tag=RELENG_4

src-all

ports-all tag=.

CVSup – update source using CVSup

> Update both src and ports

```
— % /usr/local/bin/cvsup -g -L 1 /usr/local/etc/cvsup-all  
  > /var/log/cvsup.log
```

The “-g” tells cvsup not to use its GUI

The “-L 1” tells cvsup to print out the details of
all the file updates it is doing.

from 0 (silent) to 2

It will run about 10 minutes

P4 1.8G 1GB Ram 100MB NIC

Rebuilding world

> The canonical way to update system

- make buildworld
- make buildkernel
- make installkernel
- reboot and boot in single user mode
- make installworld
- mergemaster
- reboot

Rebuilding world – Prepare make.conf

> Example make.conf

- /etc/defaults/make.conf 4.x
- /usr/share/examples/etc/make.conf 5.x

> Everything add in make.conf is used every time you run make

- KERNCONF=TYBSD

Rebuilding world – make buildworld

> Build FreeBSD entire system

- % `cd /usr/src`
- % `make -j3 buildworld >& /var/log/world.log &`

Spawn multiple (n) processes to do make.
The compiling processes of make world is I/O bound.

It will run about 30 minutes

P4 1.8G 1GB Ram 100MB NIC

Rebuilding world – make buildkernel (1)

> Why rebuild kernel?

- Fast boot time.
 - **Probe necessary device**
- Lower memory usage
 - **Smaller kernel image**
- Additional hardware support.

Rebuilding world – make buildkernel (2)

> Edit kernel config file

- `cd /usr/src/sys/i386/conf`
 - **GENERIC** may not have all for your system
 - **LINT** has every options
- `cp GENERIC "YOUR-NAME"`
 - We often use hostname to be "YOUR-NAME"
- edit config file
 - Depend on your system
 - Be attention to related options
 - Following the explanation of http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig-config.html

Rebuilding world – make buildkernel (3)

> Build kernel

- % cd /usr/src
 - % make KERNCONF=TYBSD buildkernel
-

It will run about 5 minutes depend on your configuration
P4 1.8G 1GB Ram 100MB NIC

Rebuilding world – make installkernel

> Install kernel

- % cd /usr/src
- % make KERNCONF=TYBSD installkernel

Rebuilding world – reboot in single user mode

> Boot in single user mode

- Hit any key other than “enter” when counting down
- Type “boot –s”

Or

- % shutdown now
 - **For a running system, this will drop it to single user mode**

Rebuilding world – make installworld

- > Install the built world
 - % make installworld

Rebuilding world – mergemaster

> mergemaster

- Synchronize /usr/src/etc with /etc
- Choose “i” for most case, such as
 - **/etc/defaults/rc.conf, ...**
- Press “enter” for certain file, such as
 - **master.passwd, hosts, csh.***

Reboot

- > Reboot and enjoy it
 - % reboot

If Something Goes Wrong ... (1)

> Boot with old kernel

- Hit any key other than “enter” when counting down
- Type “unload”
- Type “load /kernel.old”
- Type “boot”

```
Uncompressing ... done

BTX loader 1.00  BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
BIOS drive B: is disk1
BIOS drive C: is disk2
BIOS 639kB/129984kB available memory

FreeBSD/i386 bootstrap loader, Revision 0.8
(root@freebsd-stable.sentex.ca, Thu Apr  3 08:41:45 GMT 2003)
/kernel text=0x280131 data=0x33018+0x3311c i
-
Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [kernel] in 4 seconds...
```

If Something Goes Wrong ... (2)

> Unlock kernel

- Make sure the `securelevel` in `/etc/rc.conf` below 0, if not, change it and reboot.
- `% chflags noschg /kernel`
- `% cp kernel.old kernel`
- `sync;sync; reboot`

> Lock kernel

- `% chflags schg /kernel`

> Use `ls -lo` to check similar file

`schg` → set the immutable (永遠不變的) flag

`ls -o` → include file flags in long output

Install software

> Ports

- `cd /usr/ports`, `make search`, `make install clean`

> Package

- Pre-built ports
- `pkg_add`, `pkg_delete`, `pkg_info`

> Source

- Tar ball
- `tar xzvf certain-source.tar.gz`
- `./configure`
- `make`; `make install`

How to use ports

> Steps of install software using ports

(1) Figure out the path to the software

- % cd /usr/ports
- % make search key=mutt
- % cd /usr/ports/chinese/mutt

(2) Fetch and compile the source

- % make install

> Uninstall

- % make deinstall