

How I replace SCO OpenServer and VisionFS with FreeBSD and Samba to run FoxPro Unix

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.bsd.freebsd.misc/2004-03/1612.html>

From: Michael Green (michaelrmgreen_at_yahoo.co.uk)

Date: 03/29/04

Date: 29 Mar 2004 02:28:10 -0800

How to make a FreeBSD system.doc Revision date : 16 Feb. 2004

Export version

A note of thanks:

To Peter Elsner and Dennis Allen for all their help.

Prepare the PC

A PC with at least one ISA slot, with, ideally, an Intel Chipset and Intel CPU, a new HD, a new PSU and a new fan fitted to the existing CPU heatsink. (www.overclockers.co.uk and PCLincs). A byterunner tc800 (www.byterunner.com) serial card set to unix mode, Irq 11 and start i/o address 0x100 . FreeBSD install CD. DOS boot floppy with fdisk and format (Win98 emergency startup disk will do). IDE zip drive

What you do:

WITH THE PC OFF (unplug the cable!), reset the PC bios using the cmos jumper.

Restart

Configure the bios, set date and time, set halt on keyboard error to no (enabling boot with no keyboard), set boot sequence to CD, A, C. Set power management to off. Set PNP OS to no. Set IRQ control to manual, then reserve IRQ 11 for ISA

Save and exit

Boot from the floppy, then create a 1g dos partition with Fdisk.

Format with format c:/s. (Do not attempt to do this from FBSD the resulting partiton can't be understood by DOS format.)

Install FreeBSD

Restart with the FBSD CD in

Remove all the SCSI and PCcard card stuff in the first screen

At the colour screen "sysinstall Main Menu" choose Standard

At the Fdisk screen create two FBSD slices, both type 165 (the default), the first 5000m, the second 1000m. Then Q to exit.

At the 'Install boot manager' screen select 'none'.

At the "FreeBSD Disklabel Editor" screen select [A]uto for the first (5000m) FBSD slice, then [C]reate the second, and select type

'FileSystem', select a mount point of /data, then [Q] to finish.

At the 'Choose distribution' screen select 'Developer' (others will be autoselected).

Choose [Y]es at the install FreeBSD ports options screen, then select EXIT.

At the "Choose Installation Media" select [1] CD rom. Wait (a lot)....

Select [Y] for the user config devices screen. Select UK keyboard map.

(However this bit appears not to work and the UK keymap has to be selected in post installation configuration to work)

Set the time zone to UK

Select 'Yes' for configure inet. Enable telnet (not telnet6).

Select no for Linux compatability etc etc.....

Browse Packages to Emulators and select mtools -> install -> OK

At the 'Would you like to create initial users' -> Yes -> User and

Group management -> User, add user 'console', group 'wheel', password 'console'

Reboot, enter the BIOS setup screen and change the boot order to C: only. Continue the reboot.

If there is an invalid partition table error, reboot with the DOS boot floppy and use fdisk to inspect the partiton table. The two FBSD partitions are probably both active. Mark the large one, only, active.

If there are UDMA error messages put hw.ata.ata_dma=0 in /boot/loader.conf

Getting access to the zip drive

To access the zip drive as z: edit /usr/local/etc/mtools.conf. add line

```
drive z: file="/dev/afd0s4"
```

compare this with a: and b:

Getting the terminal controller card to work

Configure a custom kernel. Either copy a saved kernel conf file from a floppy to /usr/src/sys/i386/conf or copy GENERIC to CUSTOM (cd to /usr/src/sys/i386/conf first). Edit the kernel conf file with edit (#edit CUSTOM [Return])

Remove all the unnecessary lines. Caution, there is a line in the SCSI section 'device scbus' which is required in all kernels. Comment out COM3 and COM4. Add the lines for the Byterunner tc800:

```
options COM_MULIPORT
```

```
device sio2 at isa? port 0x100 flags 0x905
```

```
device sio3 at isa? port 0x108 flags 0x905
```

```
etc
```

```
device sio9 at isa? port 0x138 flags 0x905 irq 11
```

(See man sio and LINT)

```
Save and exit ([ESC]a a)
Check the file for errors with config...
# config CUSTOM
Change to the compile directory
# cd ../../compile/CUSTOM
#make depend
#make
#make install
#shutdown -r now (reboot)
```

If the new kernel fails to boot, reboot and press a letter key at the countdown prompt.

At the new prompt unload the faulty kernel with unload, then boot the old

kernel with boot kernel.GENERIC. Replace the faulty kernel with the generic kernel:

```
# chflags noschg /kernel
# cp kernel.GENERIC kernel
```

Edit /etc/ttys. Change std.9600 to 3wire.9600 as appropriate. Change dialup to wy60-25. Change off to on.

Add more ttydx lines upto ttyd9

Create the device special files

```
# cd /dev
# ./MAKEDEV ttyd4
etc to ttyd9
```

Restart the scheduler with ...

```
#kill -HUP 1
```

Login prompt should now come up. Hurray!

If there are error messages on the console of the form:

Date and time getty[nnnn]:login_tty /dev/ttydn: Operation not supported on device

Check /etc/ttys for errors and duplications in the ttydn lines.

Check /dev for the correct

array of ttydn files. Restart the scheduler with kill -HUP 1.

Change the lines value in the terminal setup screen to 25. ([SHIFT] [SETUP] on the terminal to access the setup system)

Check the terminal has default tabs. Turn on xon/xoff flow control.

Install ncurses...

```
#cd /usr/src/contrib/ncurses. (Read INSTALL)
```

If this dir doesn't exist the sources didn't get installed. Use /stand/sysinstall to install the sources for 'contrib'

```
#!/configure
#make
#cp /usr/src/contrib/ncurses/progs/tic /usr/local/sbin
Compile the terminfo database...
# rehash (you moved a prog)
```

```
#tic /usr/src/contrib/ncurses/misc/terminfo.src
```

Ignore the error messages.

Create a link for OpenServer programs to find terminfo. (FreeBSD tic puts terminfo files in /usr/share/terminfo. OpenServer uses /usr/lib/terminfo)

```
#ln -sf /usr/share/terminfo /usr/lib/terminfo
```

Getting iBCS2 to work

Create these directories:

```
#mkdir /usr/compat/ibcs2
```

```
#mkdir /usr/compat/ibcs2/dev
```

Create links

```
#cd /usr/compat/ibcs2/dev
```

```
#ln -sf /dev/null XOR (ecs nought arr)
```

```
#ln -sf socksyes nfsd
```

```
#ln -sf /dev/null socksyes
```

Make an empty file called null

The section above is required in spite of what the official FAQ says.

Edit /boot/loader.conf. Add lines

```
ibcs2_load="YES"
```

```
ibcs2_coff_load="YES"
```

(compare with /boot/defaults/loader.conf)

Reboot

Check that ibcs2.ko and ibcs2_coff.ko are loaded with kldstat

Move the FoxPro files on the SCO system to a zip disk

```
# cd /usr/lib/foxpro
```

```
# tar cf fox.tar *
```

```
#mv fox.tar /data/tmp
```

In My Network Places move pr_data->tmp->fox.tar to zip

Move the data files to a zip disk

```
#cd /data/usr/pacman
```

```
#tar cf prstuff.tar *
```

```
#mv prstuff.tar /data/tmp
```

In My Network Place move pr_data->tmp->prstuff.tar to zip

Move the FoxPro files to /usr/local/lib/foxpro

```
#mkdir /usr/local/lib/foxpro
```

```
# cd /usr/local/lib/foxpro (don't forget this bit)
```

```
#mcopy z:fox.tar .
```

```
#tar xf fox.tar
```

Compile the foxpro terminfo files

```
#tic /usr/local/lib/foxpro/terminfo.src
```

Move the data files

```
#mkdir /data/usr
```

```
#mkdir /data/usr/pacman  
#cd /data/usr/pacman (!)  
#mcopy z:prstuff.tar .  
#tar xf prstuff.tar
```

Alternative method of moving data files

```
#cd /tmp  
#tar cvf /tmp/all.tar /data/usr/*  
#cksum /tmp/all.tar
```

On noggin: start->run->\\pacman. Click the 'Allow' box if Zonealarm bring up a dialogue box. Click pr_tmp to access /tmp on pacman. Use the 'Copy to' function to move a copy of all.tar to c:\tmp. Use 'My Network Places' to copy c:/tmp/all.tar to /tmp on pacman2.

pacman2# cksum all.tar. Compare the results.

Change the owner, group and perms

```
# chown recep1:wheel *  
#chmod a+rw *
```

Change the directory to writable

```
#cd /data/usr  
#chmod a+w pacman
```

Locate all instances of sys(30) in the prg files. (Replace with getenv('LOGNAME'))

```
# cd /data/usr/pacman  
# fgrep -ly 'sys(30)' *.prg
```

Make a dir for the backup files

```
# mkdir /data/backup  
# cd /data  
# chmod a+w backup
```

Type in this shell script:

```
# /usr/local/bin/foxpro  
#!/bin/sh  
umask 000  
PROGDIR=/usr/local/lib/foxpro  
PATH=$PROGDIR:$PATH  
export PATH  
TZ=GMT0BST  
export TZ  
exec $PROGDIR/foxpro.pr "$@"
```

chmod the file to executable

```
# chmod a+x /usr/local/bin/foxpro
```

Create the users

```
#adduser -config_create  
#adduser -s
```

Add users: recep1, recep2, treat1, treat2, console in group wheel

The .profile files are in /usr/home/(username). Add these lines to the end of each except console.

```
FOXTERM=wy60-25fox; export FOXTERM
cd /data/usr/pacman
foxpro -eiot prstart
```

For user console add these lines :

```
cd /data/usr/pacman
N.B. the console cannot be used directly, but a telnet session can be
used for access from the console:
telnet localhost
user (console):console
password: console
$ foxpro -eiot prstart
```

Check that /bin is in the path of the users. If not printing may malfunction.

Log in as that user then:

```
$env
```

Adding a printer

Check dmesg for two lpt0 lines indicating interrupt driven printer port
dmesg | more

Create a spool dir

```
#mkdir /var/spool/lp
```

Plug in and switch on the printer. Test the link to the printer

```
#lptest > /dev/lpt0
```

Edit /etc/printcap. Add these lines:

```
lp|printer|default|HP Laserjet 4v:\
[TAB]:sh:sd=/var/spool/lp:lp=/dev/lpt0:if=/usr/local/libexec/hpif:
raw|windowsprint|Laserjet 4v via Samba:\
[TAB]:sh:sf:sd=/var/spool/raw:lp=/dev/lpt0:if=/usr/local/libexec/raw:
```

Turn on LPD by adding lpd_enable="YES" to /etc/rc.conf. Start LPD

```
#lpd
```

Type in the following input filter files: 1.For the unix printer with crlf mapping on:

```
#!/bin/sh
#/usr/local/libexec/hpif
# turns on and off lf to crlf mapping in HP printers
printf "\033&k2G" && cat && printf "\033&10H" exit 0
exit 2
```

(NB that is the letter elle and number zero in the seconf printf command, not the number ten)

2. with no mapping:

```
#!/bin/sh
```

```
#usr/local/libexec/raw  
/bin/cat && printf "\f" && exit 0  
exit 2
```

Don't forget to

```
#chmod a+rx /usr/local/libexec/hpif  
#chmod a+rx /usr/local/libexec/raw
```

Install lynx

```
# /stand/sysinstall -> Configure -> Packages -> CD -> www -> lynx  
Edit lynx.cfg (in /usr/local/etc). Change STARTFILE line to  
STARTFILE:file://localhost (see example line 96)  
Change PRINTER to: PRINTER:Local printer:lpr %s:TRUE
```

Configure network

```
/stand/sysinstall -> configure -> media -> FTP -> Primary site  
At "Running multiuser, assume network configured" select No  
Select network card (Example ed0 Novell NE1000)  
No to IPV6 configuration  
No to DHCP  
At network configuration screen  
Host: pacman.blank.blank  
IPv4 address enter: 10.0.0.1  
Netmask: 255.0.0.0  
Others blank  
Select OK, then Exit, then Exit install
```

Check TCP/IP with ping:

```
# ping 127.0.0.1 (^C to end pings)  
# ping 10.0.0.1
```

Edit /etc/hosts, add this line:

```
10.0.0.10 noggin
```

Configure the windows system (noggin) to address 10.0.0.10

```
My Computer -> Control Panel -> Network and Dial-up Connections ->  
Local Area Connection -> Properties->Internet Protocol->Properties  
->Use the following IP address. Adjust to 10.0.0.10 netmask 255.0.0.0
```

Test ping pacman and noggin

On the FBSD system use adduser -s to create user pcgust with blank password. Also add noggin/pacman.

Install Samba

```
# /stand/sysinstall -> Configure -> Packages -> CD/DVD -> Net -> Samba
```

```
Copy /usr/local/etc/smb.conf.default to /usr/local/etc/smb.conf  
Copy /usr/local/etc/rc.d/samba.sh.sample to  
/usr/local/etc/rc.d/samba.sh
```

```
Edit smb.conf
Change workgroup= to WORKGROUP.
Enable guest account = pcguest (line 49)
Enable encrypt passwords. = yes (line 75)
Add null passwords = yes
Enable preferred master = yes (line 109)
Add print command = lpr -r -P%p %s
Add public = yes to [printers] (line212)
Uncomment the [tmp] section (line 217)
Change guest ok to yes in the [printers] section (line 212)
```

```
Save and exit ([Esc]aa)
Test smb.conf with testparm
```

```
Add users pcguest and noggin to the samba password file:
Delete the default password file
# rm /usr/local/private/smbpasswd
#smbpasswd -an pcguest (-an = (a)dd (n)o password) (Ignore the 'unable
to open passwd database' message)
#smbpasswd -a noggin pacman (-a = (a)dd) (Don't forget these must be
existing users)
```

Restart the system to check that Samba starts up normally.

To start Samba without restarting use:

```
#/usr/local/etc/rc.d/samba.sh start
```

Check the daemons are running:

```
#ps -w aux | grep mbd
```

There should be two daemons running, smbd and nmbd. If only smbd is running the network is not configured correctly for either hostname or IP address. Check the 'Configure network' section above.

Add IP address (ie 10.0.0.1) of server to 'wins server' box in the advanced section of tcp/ip properties. On reflection, this is probably not necessary.

Using SWAT the Samba configurator

Edit /etc/inetd.conf. Uncomment the swat line (line 118, the last line). Save and exit. Force inetd to reread inetd.conf with

```
# killall -HUP inetd
```

Start lynx then use the (G)o command. The URL is localhost:901. The root name and password are required.

Bring the data files up to date:

Use mcopy to transfer pacman.tar to /data/usr/pacman

```
#cd /data/usr/pacman
```

```
#mcopy z:/pacman/pacman.tar . (unix slashes!)
```

Extract the tar files:

```
#tar xvf pacman.tar
```

Redo the owner and group changes:

```
#chown recep1:wheel *
```

```
# chmod a+rw *
```

Start foxpro and reindex the files ('do reindex' at the prompt), the index files are not carried over!

Change the backup routines to reflect the new paths.

Useful changes to defaults of FreeBSD:

To enable root login via telnet (Termlite etc) add the word secure to the end of the line starting tty0 (zero not oh) in /etc/ttys and restart the scheduler (#kill -HUP 1) (There are security issues here)

To enable terminal speed of 19200 edit /etc/gettytab and add an entry similar to the entry starting '3wire.9600' with 19200 substituted for 9600. These entries are right at the bottom.

Beyond 19600 xon/xoff is required. stty sets xon/xoff (See man stty and man sio):

```
#stty -f /dev/ttyid2 ixon ixoff (note the 'i', this is the device initial state file).
```

```
#stty -f /dev/ttyld2 ixon ixoff (note the 'l' this is the lock state device)
```

```
#stty -f /dev/ttyd2 ixon ixoff ( Added these to the end of /etc/rc.serial. Not sure about this)
```

To check the xon/xoff state of a line:

```
#stty -a -f /dev/ttyd2 (etc)
```

However operation at this speed is still not perfect. More research required.

** EOF How to make a FreeBSD system.doc **