



# DM&P X-Linux Developer's Manual

2008-11-24

## Table of Content

What's X-Linux .....	2
Update History .....	3
Environment Overview .....	5
Install X-Linux .....	5
Setup Files .....	7
Utilities .....	8
Using X-Linux .....	10
Disable Booting Message .....	10
Enable Disk Writable .....	10
Set Fixed IP Address and DNS .....	10
Using DHCP .....	10
Use PPP Dial-Up .....	11
Enable PPP Server .....	11
Enable Serial Console .....	12
Use USB Mass Storage .....	13
Boot from USB Mass Storage .....	13
Using NFS .....	13
Make Your Linux Kernel .....	14
Install Your Program .....	14
Demo Program .....	15
Reference Links .....	16
Technical Support .....	16

**We offer free software support resource for your reference only. Our resource might lead you to other websites. It is to save your research time and we don't have any obligation or responsibility to provide further support or answer questions on your application. If you need any special assistance, please contact your account manager.**



## What's X-Linux

Some of our customers need embedded Linux to start their development. There are too much resource about Linux and needs a lot of time to make embedded Linux. We have some projects/products using embedded Linux and our Linux programmers put it on web site. It can save money and development time for our customers about Linux application. X-Linux is maintained and improved since 2002. Bugs are fixed and customers use it as their Linux application without trouble. X-Linux does not provide full documents and friendly tools to install, but it is enough and good for most embedded Linux application.

### X-Linux feature list:

- Can run on Vortex86SX/DX series with 64M bytes memory.
- Only needs 6M bytes storage space.
- Only need 10 (or less) seconds to boot on Vortex86SX series from power on.
- Support MSTI Embeddisk.
- Support EXT3 filesystem.
- Working with read-only filesystem (using tmpfs to reduce writing Flash storage).
- Support serial console for device without VGA.
- Include FTP, TELNET and WWW server.
- Support DHCP.
- Support PPP dial-up and access PPP dial in.
- Support NFS.
- Support USB mass storage and keyboard/mouse.

If you have trouble about using X-Linux on ICOP board, please mail to [tech@dmp.com.tw](mailto:tech@dmp.com.tw). We will try to help you.



## Update History

### Release 5.51 (2008-09-12)

- Add SSH server and client.
- Add USB audio support.

### Release 5.5 (2008-08-08)

- Using Linux kernel 2.6.23.
- Update BusyBox to 1.10.4.
- Update glibc to 2.7.
- Using EXT3 filesystem.
- Add C/C++ demo program.

### Release 5.4 (2006-01-09)

- Update BusyBox to 1.01.
- Use vsftpd 2.0.3 as FTP server.
- Add NFS V2 and V3.
- Update EXT2 filesystem tools 1.38.
- Use SysLinux boot loader.
- Add domain name resolving.
- Provide three versions for M6117D, Vortex86 and Vega86 series.
- glibc updated to 2.3.3.

### Release 5.3 (2005-06-21)

- Linux kernel updated to 2.4.31.
- Use LILO 22.2 as boot loader.
- BusyBox updated.
- glibc updated to 2.3.2.
- Support USB mass storage and keyboard.
- Support Vortex86 audio.
- Support CD-ROM.
- Use tmpfs to replace RAM disk for /var and /tmp.
- Update web server to WN server 2.4.6.

### Release 5.2 (2005-01-05)

- Linux kernel updated to 2.4.28.
- SysLinux updated.
- BusyBox updated.
- glibc updated to 2.3.2.
- Support USB mass storage and keyboard.
- Support Vortex86 audio.



#### **Release 4 (2003-05-28)**

- Linux kernel updated to 2.4.20.
- SysLinux updated.
- udhcp updated.
- BusyBox updated.
- PPP server function added.
- Login shell added.
- Set root filesystem to read only.
- Link /var and /tmp to RAM disk to reduce writing of disk.
- Serial console added. (Version 4.1 only)

#### **Release 3 (2002-12-31)**

- Loadable module support enabled.
- inetd added to take some of the effort out of running services such as telnet and ftp.
- TELNET service daemon added.
- WU-FTPD service daemon added.



## Environment Overview

Software	Version	Path
Linux Kernel	2.6.23	/boot/linux
Boot Loader	SysLinux 2.13	/boot
Shell	BusyBox 1.10.4	/bin/busybox
FTP Server	vsftpd 2.0.3	/usr/sbin/ftpd
TELNET Server	BusyBox 1.10.4	/usr/sbin/telnetd
SSH Server	openssh-5.0p1	/usr/sbin/sshd
HTTP Server	WN Server 2.4.6	/usr/httpd
Share Library	glibc 2.7	/lib
DHCP Client	BusyBox 1.10.4	/sbin/udhcpd
PPP Daemon	pppd 2.4.1	/sbin
NFS	NFS-Utills 1.0.6	/sbin
Web Pages		/usr/www
<b>Size Requirement</b>		<b>&lt; 6 MB</b>

## Install X-Linux

We strongly recommend using Norton GHOST to restore X-Linux. If programmers have Linux knowledge, they can try to use tar.gz to restore. Or, it will need a lot of time for junior engineers on restoring X-Linux. We release X-Linux to help developers to start their development quickly. If developers waste too much time on restoring X-Linux, please order DOM or Embeddisk ([http://www.icop.com.tw/products\\_detail.asp?ProductID=185](http://www.icop.com.tw/products_detail.asp?ProductID=185)) from ICOP with X-Linux installed. It can save a lot of time.

We provide two solutions (GHOST image and tar.gz file) for programmer to restore X-Linux.

### Using Norton GHOST

If programmer has Norton GHOST, download GHOST file and restore X-Linux as "Disk From Image". Use GHOST to restore X-Linux is recommended.

If programmer can not use GHOST to restore X-Linux, try to use tar.gz file:

### Using tar.gz with SysLinux Boot Loader

1. Assume you have a Linux system (Linux host system can be any distribution, ex: Redhat, SUSE, Debian, etc.) and DOM is /dev/hdb in your system.
2. Use fdisk to make a primary FAT12/16 partition (/dev/hdb1) and a secondary Linux partition (/dev/hdb2).
3. Toggle the bootable flag to the primary partition.
4. Format the primary partition and Linux partition as EXT3 filesystem.



```
# mkdosfs /dev/hdb1
```

```
# mke3fs /dev/hdb2
```

5. You will have those two partitions:

/dev/hdb1	*	1	62	984	6	FAT16
/dev/hdb2		63	1000	15008	83	Linux

6. Mount /dev/hdb2 to /mnt. (Ex: "**mount -t ext3 /dev/hdb2 /mnt**").
7. Run "**tar -xzf ~/xlr551-rootfs.tar.gz -C /mnt**" to restore X-Linux root filesystem.
8. Mount /dev/hdb1 to /mnt/boot. (Ex: "**mount -t msdos /dev/hdb1 /mnt/boot**")
9. Run "**syslinux /dev/hdb1**" to install SysLinux.
10. Copy /etc/syslinux.cfg to /mnt/boot. (ex: "**cp /etc/syslinux.cfg /mnt/boot**")
11. Run "**tar -xzf ~/xlr551-kernel-2.6.23-vortex86sx.tar.gz -C /mnt/boot**".
12. Umount /dev/hdb1 and /dev/hdb2.
13. Plug DOM to ICOP/DMP boards and set it as IDE master to boot.



## Setup Files

File Name	Description
/boot/linux	Linux kernel.
/etc/exports	The file describing exported filesystems for NFS services.
/etc/fstab	Lists the filesystems mounted automatically at startup by the mount -a command in startup file.
/etc/group	Similar to /etc/passwd but for groups rather than users.
/etc/hosts	List hosts for name lookup use that are locally required.
/etc/inetd.conf	The inetd.conf file contains the list of servers that inetd invokes when it receives an Internet request over a socket.
/etc/init.d/rcS	It will be run first and you can add your initial programs into it. We add statement assign our IP in this file.
/etc/inittab	This file plays a crucial role in the boot sequence.
/etc/kernel-config	Linux kernel configuration file. Developer can use the kernel configuration to build new Linux kernel.
/etc/mtab	A list of currently mounted filesystems generated by /proc.
/etc/nsswitch.conf	Name service switch configuration file.
/etc/passwd	The user database with fields giving the username, real name, home directory, encrypted password and other information about each user.
/etc/ppp/pap-secrets	User name and password file for PPP dial-up.
/etc/profile	It work as autoexec.bat under DOS and will be run automatically.
/etc/protocols	Describes DARPA internet protocols available from the TCP/IP subsystem. Maps protocol ID numbers to protocol names.
/etc/resolv.conf	Configures the name resolver, specifying the address of your name server and your domain name.
/etc/securetty	This file allows you to specify which TTY devices the root user is allowed to login on.
/etc/services	This file contains information regarding the known services available in the DARPA Internet.
/etc/shadow	Shadow password file on systems with shadow password software installed. Shadow passwords move the encrypted password files from /etc/passwd to /etc/shadow which can only be read by root.
/etc/syslinux.cfg	SysLinux configuration file. Copy it to target device after installing SysLinux.
/etc/vsftpd.conf	Vsftpd configuration file.
/usr/www	Our web pages are here.
/usr/local/etc/ssh_config	Configuration file for SSH client.
/usr/local/etc/sshd_config	Configuration file for SSH server.



## Utilities

Run those commands via keyboard or scripts. List of important commands/scripts:

Command	Path	Description
setserial	/bin	Get/set Linux serial ports information.
fdisk	/sbin	Partition table manipulator for Linux.
fsck	/sbin	Filesystem check tool.
httpd	/sbin	Start a HTTP server.
ifconfig	/sbin	Configure a network interface.
mkdosfs	/sbin	Create an MS-DOS filesystem under Linux.
mkfs.ext3	/sbin	Utility to create an EXT2/EXT3 filesystem.
mke2fs	/sbin	Create a Linux second extended filesystem.
portmap	/sbin	Start RPC to be a NFS client.
rw	/sbin	Script to make root filesystem writable.
syslinux	/sbin	SysLinux boot loader.
tune2fs	/sbin	Adjust tunable filesystem parameters on second extended filesystems.
exportfs	/usr/sbin	Maintain list of NFS exported filesystems.
nfsd	/usr/sbin	Script to enable NFS server. Refer to below section about NFS for more.
ppp-off	/usr/sbin	Stop a ppp connection.
ppp-on	/usr/sbin	Start a ppp connection.
rpcinfo	/usr/sbin	Show port used by RPC.

Commands listed below are implemented by BusyBox:

(You can go to <http://www.busybox.net/downloads/BusyBox.html> to get more information.)

Path	Command
/bin	[, [[, addgroup, adduser, ar, arping, ash, awk, basename, bunzip2, busybox, bzip2, cal, cat, catv, chattr, chgrp, chmod, chown, chpst, chrt, chvt, cksum, cmp, comm, cp, cpio, crontab, cryptpw, cut, date, dc, dd, dealloctv, delgroup, deluser, df, diff, dirname, dmesg, dos2unix, du, dumpkmap, dumpleases, echo, ed, egrep, eject, env, envdir, ether-wake, expand, expr, false, fdformat, fetchmail, fgrep, find, fold, free, ftpget, ftpput, fuser, getopt, grep, gunzip, gzip, head, hexdump, hostname, id, install, ip, ipaddr, ipcalc, ipcrm, ipcs, iplink, iproute, iprule, iptunnel, kbd_mode, kill, killall, killall5, last, length, less, linux32, ln, loadfont, logger, login, logname, lpq, lpr, ls, lsattr, lzmacat, md5sum, msg, microcom, mkdir, mkfifo, mknod, mktemp, more, mount, mountpoint, mt, mv, netstat, nice, nohup, nslookup, od, openvt, passwd, patch, pgrep, pidof, ping, ping6, pipe_progress, pkill, printenv, printf, ps, pscan, pwd, readahead, realpath, renice, reset, resize, rm, rmdir, rpm, rpm2cpio, run-parts, runsv, runsvdir, rx, script, sed, sendmail, seq, setarch, setkeycodes, setserial, setsid, setuidgid, sh, sha1sum, sleep, softlimit, sort, split, stat, strings, stty, su, sum, sv, svlogd, sync, tac, tail, tar, taskset, tcpsvd, tee, telnet, test, tftpd, time, top, touch, tr, traceroute, true, tty, ttysize, udpsvd, umount, uname, uncompress, unexpand, uniq, unix2dos, unlzma, unzip, uptime, usleep, uudecode, uuencode, vi, vlock, watch, wc, wget, which, who, whoami, xargs, yes, zcat



/sbin	adjtimex, arp, brctl, chpasswd, chroot, crond, dhcprelay, dnssd, e2fsck, fakeidentd, fbset, fdisk, freeramdisk, fsck, fsck.ext2, fsck.ext3, fsck.minix, fsck.msos, fsck.nfs, getty, halt, hdparm, httpd, hwclock, ifconfig, ifdown, ifenslave, ifup, inetd, init, insmod, klogd, lilo, lilo.real, loadkmap, logread, losetup, lpd, lsmod, makedevs, mdev, mgetty, mii-tool, mkdosfs, mke2fs, mkfs.ext3, mkfs.minix, mkfs.msos, mkswap, modprobe, nameif, pivot_root, portmap, poweroff, raidautorun, rdate, readprofile, reboot, rmmmod, ro, route, runlevel, rw, setconsole, setlogcons, slattach, start-stop-daemon, sulogin, swapoff, swapon, switch_root, sysctl, syslinux, syslogd, telnetd, tune2fs, udhcpc, udhcpd, vconfig, watchdog, zcip
/usr/bin	[, ar, awk, basename, bunzip2, bzip2, clear, cmp, crontab, cut, dirname, dos2unix, du, env, expr, find, fold, free, head, hexdump, hostid, id, install, killall, ldd, lddlibc4, length, loadfont, logname, mattrib, mcopy, mtools, od, passwd, readlink, realpath, renice, reset, scp, seq, setkeycodes, sftp, sort, ssh, ssh-add, ssh-agent, ssh-keygen, ssh-keyscan, strings, tail, tee, telnet, test, tftp, time, top, tr, traceroute, tty, uniq, unix2dos, unzip, uptime, wc, wget, which, who, whoami, xargs, yes
/usr/sbin	brctl, callback, chat, chroot, crond, exportfs, ftpd, inetd, nfsd, nfsstat, pmap_dump, pmap_set, ppp-off, ppp-on, pppd, rpc.lockd, rpc.mountd, rpc.nfsd, rpc.statd, rpcinfo, showmount, sshd, telnetd, vsftpd



## Using X-Linux

Default setting may not meet programmer's requirement. Here are some sections to help developer to change X-Linux settings.

The default user name is **root** and password is **password**. Remember to change default password.

### Disable Booting Message

If developer wants to disable Linux kernel booting message, add "quiet" to boot loader configuration. For example, if SysLinux is used as your boot loader, edit /boot/syslinux.cfg to add "quiet":

```
APPEND root=/dev/hda2 quiet
```

And redirection messages from /etc/init.d/rcS and /etc/profile:

```
::sysinit:/etc/init.d/rcS >/dev/null 2>&1
```

### Enable Disk Writable

We only set root filesystem can be read because embedded system will power-off at any time. If any data want to save into disk, re-mount disk writable is needed. For example: user wants to use FTP to upload file, he should uses telnet to X-Linux and do those steps:

```
# mount -o remount,rw / (mount root filesystem readable/writable)
# (use FTP to upload files...)
# sync (flush filesystem buffers)
# mount -o remount,ro / (mount root filesystem read only)
```

Or, using "rw" script in /sbin to do those commands.

### Set Fixed IP Address and DNS

If user wants to use fixed IP address, change this line in /etc/init.d/rcS:

```
echo $Linux_string: Set IP=192.168.0.222
ifconfig eth0 192.168.0.222 netmask 255.255.255.0
```

If DNS is needed, edit /etc/resolv.conf to add your DNS server:

```
nameserver 192.168.0.1
```

### Using DHCP

Remove the "#" in /etc/init.d/rcS to enable DHCP:

```
#echo $Linux_string: Getting IP from DHCP server
#udhcpc >/dev/null 2>&1
```



## Use PPP Dial-Up

Modify configuration files with information from your ISP to enable PPP dial-up. We recommend using FTP to download it, modify it, and then upload it. Assume you connect modem to COM1.

### 1. `/etc/ppp/pap-secrets`

This file contain user's account, you can see first line:

```
my_username ppp0 my_password
```

"my\_username" is your dial-up account user name, "my\_password" is password. You should change them with yours.

### 2. `/usr/sbin/ppp-on`

```
#!/bin/sh
pppd nodetach lock modem crtscts /dev/ttyS0 38400 noipdefault defaultroute noauth \
name my_username connect "chat -v ' ' ATDT123456789 CONNECT" &
```

You should replace "my\_username" with your user name as `/etc/ppp/pap-secrets`. \ "ATDT123456789" is AT command to dial to ISP. You should replace "123456789" with your ISP's phone number.

Now, you can type "ppp-on" to dial-up to your ISP and "ppp-off" to hang-up. Type "ifconfig" to display network configuration, you can find device "eth0" and "ppp0". Any problem, mail us to help you to dial-up.

## Enable PPP Server

Follow those steps to enable PPP server function.

### 1. `/etc/inittab`

You can find those two lines on `/etc/inittab`:

```
# remove '#' of next line to enable PPP server function, but ppp client function will
not work
#ttyS0::respawn:/sbin/mgetty -D -a modem ttyS0
```

Remember to remount root filesystem readable/writable and remove '#' of "#ttyS0".

### 2. `/etc/ppp/options.ttyS0`

Use mv to restore `/etc/ppp/options.ttyS0`:

```
# mv /etc/ppp/options.ttyS0.pppd /etc/ppp/options.ttyS0
```



/etc/ppp/options.ttyS0 is needed for PPP server.

### 3. PPP Account

Default use name and password for PPP client to dial-up are "pppuser" and "xlinux". If you want to add other user account, just duplicate line 2 of "/etc/ppp/pap-secrets" and replace "pppuser" with new user name and "xlinux" with new password:

```
my_username ppp0 my_password *
pppuser      *      xlinux      *
```

After doing that, login in as root and do those steps:

```
# adduser new_user_name (add new user)
# passwd new_user_name (change user's password)
```

Because accepting PPP dial-up will run mgetty which read account data from system to verify user, we have to add user account to accord with /etc/ppp/pap-secrets.

### 4. Run "sync" command and reboot xLinux.

If you enable PPP server function, PPP client dial-up function of xLinux will not work.

## Enable Serial Console

Flow those steps to add serial console into X-Linux R5:

1. Run "**mount -o remount,rw /**" to make root filesystem can be read/wrote.
2. Edit **/etc/inittab** to add this line or remote "#" at start of this line:

```
# remove '#' of next line to support serial console logins
ttyS0::respawn:/sbin/getty -L 9600 ttyS0
```

3. Edit **/etc/secrtyty** to add those lines:

```
ttyS0
ttyS1
```

4. Edit **/boot/syslinux.cfg** to add this line:

```
APPEND "console=tty1 console=ttyS0,9600n8"
```

5. Run "sync" and reboot the device.
6. Use a cross RS-232 cable to connect to X-Linux and your PC.
7. Run terminal program and set COM parameters:

```
Baud Rate      = 9600
Parity Bit     = No
Data Bits      = 8
Stop Bits      = 1
Flow Control   = OFF
```

8. Turn on X-Linux device. You can get message from serial port and login now.



When /dev/ttyS0 is used by serial console, PPP client or server function will not work. You can enable one of them or use another serial port (ex: /dev/ttyS1) for serial console or PPP.

## Use USB Mass Storage

USB function is enabled in X-Linux kernel. After enabling USB in BIOS, X-Linux can find USB mass storage device. When USB storage is plugged, Linux will find SCSI devices:

```
hub.c: new USB device 00:01.2-1, assigned address 2
scsi0 : SCSI emulation for USB Mass Storage devices
  Vendor: Usb      Model: Flash Disk      Rev: 1.11
  Type:   Direct-Access          ANSI SCSI revision: 02
Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
SCSI device sda: 129024 512-byte hdwr sectors (66 MB)
sda: Write Protect is off
sda: sda1 sda2 < >
```

Mount USB storage:

```
~ # mount /dev/sda1 /mnt
```

Now, you can read/write USB storage. Linux kernel in X-Linux only support EXT2 and FAT16/32 filesystem. If you other filesystem on USB storage, X-Linux can not recognize it. Beside, before removing your USB storage, please unmount USB device first. Also, USB keyboard can work with X-Linux.

## Boot from USB Mass Storage

If developer wants to boot X-Linux from USB mass storage, follow the installation steps in previous chapter and do those changes:

1. Edit /boot/syslinux.cfg to modify APPEND line as

```
APPEND root=/dev/sda2 rootdelay=5
```

2. Edit /etc/init.d/rcS, replace "/bin/mount -a" to

```
/bin/mount -t msdos /dev/sda1 /boot
```

## Using NFS

NFS is helpful for development phase. We add NFS support into X-Linux and here are steps to enable NFS:

### X-Linux to be NFS server

1. Run "portmap" and "nfsd".



2. Modify **"/etc/exports"** to add share path. Share path is root by default in X-Linux.
3. Run **"exportfs -vr"** to make settings in **"/etc/exports"** active.

### X-Linux to be NFS client

1. Run **"portmap"**.
2. To get sharing status of NFS server, run **"showmount -e (hostname or ip)"**.
3. To mount a NFS share: **"mount -t nfs host:/ /mnt/nfs"**
4. To umount a NFS share: **"umount /mnt/nfs"**

### Make Your Linux Kernel

If developer wants to extend Linux kernel function, copy the kernel configuration file from X-Linux (for example, the **/etc/kernel-config/2.6.23-vortex86sx.config** is kernel configuration for Linux kernel 2.6.23 for Vortex86SX series) to your Linux kernel source code directory and rename it to **".config"**. Then, you can make your kernel based on X-Linux kernel. Configuration file for BusyBox 1.10.4 is also at the same directory. Developer can use BusyBox configuration file to make his own BusyBox.

### Install Your Program

This is an example form our Mity-Mite demo box Linux version. We wrote simple LCM library under Linux.

FTP server is enabled on our Linux DOC/MDM that can use FTP to upload programs. The default user name is "root" and password is "password". Install Linux (Red Hat, Mandrake, SuSE...) on a powerful PC to develop applications is recommended. Write and compile programs then upload it to Mity-Mite/Tiny module via FTP to test. Kill the process if program is not correct. Repeat those steps until programs are okay. There is an example:

1. We create a directory **"/demo"**.
2. Put lcd (LCD driver), dmp.bmp (DM&P logo, will loaded by lcd) and lcd\_time (send time to display every 0.5 second) into **"/demo"**.
3. Write a script **"demo"** to load lcd and lcd\_time.

```
#!/bin/sh ./lcd&
sleep 5
./lcd_time&
```

4. Remember to change mode of upload files to executable.

```
chmod +x ./lcd
chmod +x ./lcd_time
chmod +x ./demo
```

5. Run **"demo"** to test. If programs are not correct, use **"ps"** to find process ID of program to kill it. Go to step 2 until programs are okay.

```
# ps
```



```
PID  Uid    Stat Command
   1  0      S    init
   2  0      S    [keventd]
   3  0      S    [ksoftirqd_CPU0]
   4  0      S    [kswapd]
   5  0      S    [bdflush]
   6  0      S    [kupdated]
  11  0      S    -sh
  14  0      S    ./ftpd
  17  0      S    /usr/httpd/wnsd -p 80
  19  0      S    ./lcd
  21  0      S    ./lcd_time
  22  0      R    ps
# kill 21
# kill 22
```

Also, you can lunch "demo" script at /etc/profile to run it automatically when program is okay.

### Hint

Before uploading your programs, use "ldd" to check share library dependency. Upload share library to /lib first if need.

```
[root@/root/grlcd]# ldd lcd
        libstdc++-libc6.2-2.so.3      =>      /usr/lib/libstdc++-libc6.2-2.so.3
(0x4002c000)
        libm.so.6 => /lib/libm.so.6 (0x4006e000)
        libc.so.6 => /lib/libc.so.6 (0x40090000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
[root@/root/grlcd]#
```

### Demo Program

There are some simple C/C++ example code at /demo directory. Developers can copy those source code to desktop PC to compile. Here are compiling commands:

```
# gcc hello.c -o hello_c -march=i486
# gcc float_demo.c -o float_demo_c -lm -march=i486
# gcc thread.c -o thread_c -lpthread -march=i486
# g++ hello.cpp -o hello_cpp -march=i486
# g++ float_demo.cpp -o float_demo_cpp -march=i486
# g++ thread_demo.cpp -o thread_demo_cpp -lpthread -march=i486
```



## Reference Links

Linux Kernel	<a href="http://www.kernel.org/">http://www.kernel.org/</a>
SysLinux	<a href="http://syslinux.zytor.com/">http://syslinux.zytor.com/</a>
BusyBox	<a href="http://www.busybox.net/">http://www.busybox.net/</a>
Linux NetKit	<a href="ftp://ftp.uk.linux.org/pub/linux/Networking">ftp://ftp.uk.linux.org/pub/linux/Networking</a>
vsftpd	<a href="http://vsftpd.beasts.org/">http://vsftpd.beasts.org/</a>
WN Server	<a href="http://hopf.math.nwu.edu/">http://hopf.math.nwu.edu/</a>
LILO	<a href="http://en.wikipedia.org/wiki/LILO_(boot_loader)">http://en.wikipedia.org/wiki/LILO_(boot_loader)</a>

## Technical Support

For more technical support, please visit <http://www.dmp.com.tw/tech> or mail to [tech@dmp.com.tw](mailto:tech@dmp.com.tw).