DM&P X-Linux Developer's Manual

2010-04-28

Table of Content

1. What's X-Linux

We have some projects/products using embedded Linux and engineers make it as X-Linux. It is for our customers need embedded Linux to start their development. X-Linux is maintained and improved since 2002. Bugs are fixed and customers can use it as their Linux application without embedded Linux platform setup. Because it does not provide full documents and tool-chain, developers needs to modify it manually.

X-Linux feature list:

- Can run on Vortex86SX/DX/MX series with 64M bytes memory.
- Only need 10M bytes storage space.
- Only need 10 seconds to boot on Vortex86SX/DX/MX series after POST.
- Support EXT2/EXT3 file system.
- Working with read-only file system (using tmpfs to reduce writing Flash storage).
- Support serial console for device without VGA.
- Include FTP, TELNET and WWW server.
- Support DHCP client.
- Support NFS.
- Support SSH.
- Support USB mass storage and USB keyboard/mouse.
- Support NTP client.

2. Update History

Version 5.7 (2010-04-13)

- PPP is removed.
- Add install scripts.
- Using FTP/Telnet/HTTP server in BusyBox.
- Update BusyBox to 1.16.1.
- Fix frame buffer bug.

Version 5.63 (2009-09-01)

- Fix USB audio.

Version 5.62 (2009-07-10)

- Support MC35 GPRS modem.
- Add setterm to disable blank screen.
- Add NTP client.

Version 5.61 (2009-05-15)

- Support frame buffer in kernel.

Version 5.6 (2009-05-05)

- Support Vortex86DX.
- Using Linux kernel 2.6.29.
- Update glibc to 2.8.90.
- Update BusyBox to 1.13.2.
- Add Dropbear 0.52.

Version 5.51 (2008-09-12)

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

Version 5.5 (2008-08-08)

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

Version 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 file system 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.

Version 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.

Version 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.

Version 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 file system to read only.
- Link /var and /tmp to RAM disk to reduce writing of disk.
- Serial console added. (Version 4.1 only)

Version 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.

3. Environment Overview

Software	**Version**	**Path**
Linux Kernel	2.6.29	/boot/bzImage
Boot Loader	SysLinux 2.13	/boot
Shell	BusyBox 1.16.1	/bin/busybox
FTP Server	BusyBox 1.16.1	/sbin/ftpd
TELNET Server	BusyBox 1.13.2	/sbin/telnetd
SSH Server	Dropbear 0.52	/sbin/dropbear
HTTP Server	BusyBox 1.16.1	/sbin/httpd
NFS	NFS-Utils 1.0.6	/usr/sbin/nfsd
Web Pages		/www
Size Requirement		**< 10 MB**

4. X-Linux Installation

X-Linux has installation scripts to install itself onto RAM disk, USB mass storage or IDE device. Download X-Linux RAM disk image from web site and make a bootable USB to boot into DOS (or, boot DOS from DOM) to run it. Here are steps to make bootable USB mass storage to install X-Linux.

4.1. Run X-Linux Installation Script

4.1.1. Install X-Linux from Windows Machine

Here are steps to using X-Linux (from DOS via RAM disk image) to install X-Linux:

1. Extract X-Linux RAM disk ZIP file onto your Windows system.
2. Assume the target directory is "**xlinux-5.7-makebootfat**".
3. Ensure your Windows system only has one USB mass storage plugged.
4. Before runing "**xlinux-5.7-makebootfat\make.bat"** (or **make_lba.bat**), format your USB mass storage with FAT or FAT32 format.
5. Run the batch file and it will search the only USB mass storage to make it FreeDOS bootable.
6. If your USB mass storage can not boot properly, run make_lba.bat to try again.

After above steps, your USB mass storage is FreeDOS bootable and has those files:

File Name	Description
bzimage	Linux kernel image.
loadlin.exe	DOS tool to load Linux.
autoexec.bat	DOS batch file to launch loadlin.exe with Linux kernel and RAM disk image after boot.
ramdisk.gz	X-Linux RAM disk image.

Plug USB mass storage onto Vortex86SX/DX/MX boards and it will boot into FreeDOS and load X-Linux RAM disk image. After booting into X-Linux, enter directory "**/xlinux**" to run "**./install-xlinux.sh**" script. Check next section for more detail.

If above steps still can not make your USB mass storage boot on Vortex86SX/DX/MX boards, please search "**hpusbf.exe download**" from Google to get hpusbf.exe. Put it onto the directory where make.bat (or make_lba.bat) is. Run below command to make a FreeDOS bootable USB mass storage by HP USB utility:

```
hpusbf hd1 -fs:fat -q -b:.\freedos -y -s:.\image
```

4.1.2. Install X-Linux from Linux Machine

The other way to install X-Linux in Linux is to download X-Linux installation source from our web site. Extract it onto your Linux machine. Ensure utility mcopy and mattrib are available in your Linux system. They are member of

MS-DOS tool (mtools) and needed by SysLinux. Assume the target directory is "**/xlinux-5.7-src**". Enter "**/xlinux-5.7-src/xlinux**" and run "**./install-xlinux.sh"** script. Check next section for more detail.

4.1.3. Installation Script

Here is example to run install script from USB mass storage boot:

```
[root@X-Linux]:~ # cd xlinux/
[root@X-Linux]:/xlinux # ./install-xlinux.sh

Usage: ./install-xlinux.sh DEVICE

       DEVICE=/dev/hd[a-d] -> install X-Linux on to hard disk
       DEVICE=/dev/sd[a-b] -> install X-Linux on to USB mass storage
       DEVICE=img          -> make X-Linux RAM disk image
       DEVICE=imgbb        -> make X-Linux RAM disk image with BusyBox only
       vsx                 -> using Linux kernl with FPU emulation for
                              Vortex86SX with /dev/hdx and /dev/sdx

  Ex: ./install-xlinux.sh /dev/hdc
      ./install-xlinux.sh /dev/sda
      ./install-xlinux.sh /dev/hdc
      ./install-xlinux.sh /dev/sda
      ./install-xlinux.sh img
      ./install-xlinux.sh imgbb

[root@X-Linux]:/xlinux #
```

Note: X-Linux install script does not check error return. It can work properly for most case. Please run it carefully.

4.2. Install X-Linux onto IDE device

```
[root@X-Linux]:/xlinux # ./install-xlinux.sh /dev/hda

Install X-Linux onto /dev/hda

Make partitions
Fromat /dev/hda1
Install boot loader
Format /dev/hda2
Mount /dev/hda2
Make basic directories
Install BusyBox
Make device nodes
Copy directories
Sync
Umount /dev/hda2
Done!

[root@X-Linux]:/xlinux #
```

After this step, remove USB mass storage and boot X-Linux from your IDE device.

If you want more storage space and do not need to install X-Linux from your IDE device, please mount /dev/hd?2 to remove “**bzImage-sx**” and “**bzImage-dx**” at **xlinux** directory.

4.3. Install X-Linux onto USB mass storage

```
[root@X-Linux]:/xlinux # ./install-xlinux.sh /dev/sda

Install X-Linux onto /dev/sda

Make partitions
Fromat /dev/sda1
Install boot loader
Format /dev/sda2
Mount /dev/sda2
Make basic directories
Install BusyBox
Make device nodes
Copy directories
Sync
Umount /dev/sda2
Done!

[root@X-Linux]:/xlinux #
```

For boot X-Linux from USB mass storage, X-Linux will add "**rootdelay=10**" in syslinux.cfg. Just plug USB mass storage onto Vorex86SX/DX/MX boards to boot X-Linux.

Note (1): if you are using IDE-to-USB cable to convert IDE device as USB mass storage in X-Linux, run "**./sda-patch.sh**". It will patch **syslinux.cfg** and **/etc/fsteb** for IDE device boot (using **/dev/hdax** to replace **/dev/sdax**) .

Note (2): if your USB pen driver is < 512Mbytes, Vortex86SX/DX/MX BIOS will not recognize it as "Hard Disk" mode. Please select "Hard Disk" in BIOS "**Advance → USB Mass Storage Device Configuation → Emulation Type → Hard Disk**" for USB pen driver boot.

Note (3): if you are booting from USB pen driver, also can run install script to install X-Linux on it (/dev/sda) to boot without FreeDOS.

If you want more storage space and do not need to install X-Linux from your IDE device, please mount /dev/sd?2 to remove "**bzImage-sx**" and "**bzImage-dx**" at **xlinux** directory.

4.4. Make X-Linux RAM Disk

```
[root@X-Linux]:/xlinux # ./install-xlinux.sh img

Install X-Linux onto RAM disk image
Create RAM disk image
16384+0 records in
16384+0 records out
16777216 bytes (16.0MB) copied, 0.361298 seconds, 44.3MB/s
Mount RAM disk image
Make basic directories
Install BusyBox
Make device nodes
Copy directories
Sync
Umount RAM disk image
X-Linux RAM disk image path = /tmp/ramdisk.gz
Done!

[root@X-Linux]:/xlinux #
```

After running "**install-x-linux.sh**" with **img** or **imgbb**, a RAM disk image will be generated on /tmp. Developers can load X-Linux from DOS by loadlin.exe or from FAT16 file system by syslinux.

If you want boot RAM disk image faster and do not need to install X-Linux from your IDE device, remove "**bzImage-sx**" and "**bzImage-dx**" at /**xlinux** directory before running install-xlinux.sh.

5. Setup Files

File Name	Description
/etc/dropbear	Configuration files for dropbear SSH tool.
/etc/exports	The file describing exported file systems for NFS services.
/etc/fstab	Lists the file systems 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/ld.so.conf	File containing a list of colon, space, tab, newline, or comma-separated directories in which to search for libraries for lddconfig.
/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/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/rpc	rpc program number data base.
/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.
/www	Web pages for HTTP server.

6. BusyBox Commands

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, bbconfig, beep, bunzip2, busybox, bzcat, bzip2, cal, cat, catv, chat, chattr, chgrp, chmod, chown, chpst, chrt, chvt, cksum, clear, cmp, comm, cp, cpio, crontab, cryptpw, cttyhack, cut, date, dc, dd, deallocvt, delgroup, deluser, df, diff, dirname, dmesg, dnsdomainname, dos2unix, dpkg, dpkg-deb, du, dumpkmap, echo, ed, egrep, eject, env, envdir, envuidgid, ether-wake, expand, expr, false, fdflush, fdformat, fgrep, find, fold, free, fsync, ftpget, ftpput, fuser, getopt, grep, gunzip, gzip, hd, head, hexdump, hostid, hostname, hush, id, ifplugd, install, ionice, ip, ipaddr, ipcalc, ipcrm, ipcs, iplink, iproute, iprule, iptunnel, kbd_mode, kill, killall, killall5, last, length, less, linux32, linux64, ln, logger, login, logname, lpq, lpr, ls, lsattr, lzmacat, lzop, lzopcat, md5sum, mesg, microcom, mkdir, mkfifo, mknod, mkpasswd, mktemp, more, mount, mountpoint, msh, mt, mv, nc, netstat, nice, nmeter, nohup, nslookup, od, openvt, passwd, patch, pgrep, pidof, ping, ping6, pipe_progress, pkill, printenv, printf, ps, pscan, pwd, readlink, realpath, renice, reset, resize, rm, rmdir, rpm, rpm2cpio, rtcwake, run-parts, runsv, runsvdir, rx, script, scriptreplay, sed, seq, setarch, setkeycodes, setsid, setuidgid, sh, sha1sum, sha256sum, sha512sum, showkey, sleep, softlimit, sort, split, stat, strings, stty, su, sum, sv, sync, tac, tail, tar, taskset, tcpsvd, tee, telnet, test, time, timeout, top, touch, tr, traceroute, traceroute6, true, tty, ttysize, udpsvd, umount, uname, uncompress, unexpand, uniq, unix2dos, unlzma, unlzop, unzip, uptime, usleep, uudecode, uuencode, vi, vlock, volname, wall, watch, wc, wget, which, who, whoami, xargs, yes, zcat
/sbin	adjtimex, arp, blkid, brctl, chpasswd, chroot, crond, depmod, devfsd, devmem, dnsd, fakeidentd, fbset, fbsplash, fdisk, findfs, freeramdisk, fsck, fsck.minix, ftpd, getty, halt, hdparm, httpd, hwclock, ifconfig, ifdown, ifenslave, ifup, inetd, init, inotifyd, insmod, klogd, loadfont, loadkmap, logread, losetup, lpd, lsmod, lspci, lsusb, makedevs, man, mdev, mke2fs, mkfs.ext2, mkfs.minix, mkfs.vfat, mkswap, modprobe, nameif, ntpd, pivot_root, poweroff, raidautorun, rdate, rdev, readprofile, reboot, rmmod, route, runlevel, setconsole, setfont, setlogcons, slattach, start-stop-daemon, sulogin, svlogd, swapoff, swapon, switch_root, sysctl, syslogd, telnetd, tunctl, tune2fs, udhcpc, vconfig, watchdog, zcip

7. Non-BusyBox Utilities

Command	Path	Description
setserial	/bin	Get/set Linux serial ports information.
ldconfig	/sbin	Configure dynamic linker run time bindings
mkdosfs	/sbin	Utility to create an MS-DOS file system.
portmap	/sbin	Start RPC to be a NFS client.
ro	/sbin	Script to make root file system read only.
rw	/sbin	Script to make root file system writable.
syslinux	/sbin	SysLinux boot loader.
dbclient	/usr/bin	Dropbear SSH client.
ldd	/usr/bin	Print shared library dependencies.
dropbear	/usr/sbin	Deopbear SSH server.
e2fsck	/sbin	e2fsck is used to check a Linux second extended file system (ext2fs).
exportfs	/usr/sbin	Maintain list of NFS exported file systems.
nfsd	/usr/sbin	Script to enable NFS server. Refer to below section about NFS for more.
ntpdate	/usr/sbin	NTP client tool.
rpcinfo	/usr/sbin	Show port used by RPC.
setterm	/usr/sbin	Set terminal attributes.
showmount	/usr/sbin	Show mount information for an NFS server.

8. Using X-Linux

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

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

8.1. 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
```

8.2. Enable Disk Writable

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

Or, using "**rw**" script in /sbin to do those commands and "**ro**" to make file system read only again.

8.3. Enable Hard Disk DMA

User can run "hdparm -d0 /dev/hda" to disable DMA and run "hdparm -d1 /dev/hda" to enable DMA after X-Linux boot. Or, add them into /etc/profile. The other way to disable DMA is to add "ide=nodma" in /boot/syslinux.cfg to pass parameter to Linux kernel.

8.4. Set Fixed IP Address and DNS

If user wants to use fixed IP address, change/add those lines 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
```

8.5. 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
```

8.6. Enable Serial Console

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

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

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

3. Edit **/etc/securetty** 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.

8.7. 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/EXT3 and FAT16/32 file system. If you have other file system on USB storage, X-Linux can not recognize it. Beside, before removing your USB storage, please unmount USB device first.

8.8. Modify RAM Disk Image

1. Decompress ramdisk.gz on your desktop Linux (assume /tmp).
2. # losetup /dev/loop0 /tmp/ramdisk
3. # mount /dev/loop0 /mnt
4. Do you job on /mnt.
5. # sync
6. # umount /mnt
7. # losetup -d /dev/loop0
8. # gzip -v9 /tmp/ramdisk
9. You get a new ramdisk.gz with your changes.

8.9. 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**"

8.10. Using NTP

Run "ntpdate 0.uk.pool.ntp.org" to test, where 0.uk.pool.ntp.org is NTP server. Run “hwclock -w" to write new time into RTC.

8.11. Disable Blank Screen

Run "setterm -blank 0".

8.12. Using Frame Buffer

Frame buffer is built-in Linux kernel in X-Linux. User just has to edit syslinux.cfg to add vga parameter.

VGA mode list:

```
Mode: Resolution:  Type: Mode: Resolution:  Type: Mode: Resolution:  Type:
0 F00   80x25      VGA   1 F01   80x50      VGA   2 F02   80x43      VGA
3 F03   80x28      VGA   4 F05   80x30      VGA   5 F06   80x34      VGA
6 F07   80x60      VGA   7 300  640x400x8   VESA  8 301  640x480x8   VESA
9 303  800x600x8   VESA  a 305 1024x768x8   VESA  b 307 1280x1024x8  VESA
c 30E  320x200x16  VESA  d 30F  320x200x32  VESA  e 311  640x480x16  VESA
f 312  640x480x32  VESA  g 314  800x600x16  VESA  h 315  800x600x32  VESA
i 317 1024x768x16  VESA  j 318 1024x768x32  VESA  k 31A 1280x1024x16 VESA
l 31B 1280x1024x32 VESA  m 330  320x200x8   VESA  n 331  320x400x8   VESA
o 332  320x400x16  VESA  p 333  320x400x32  VESA  q 334  320x240x8   VESA
r 335  320x240x16  VESA  s 336  320x240x32  VESA  t 33D  640x400x16  VESA
u 33E  640x400x32  VESA  v 310  640x480x15  VESA  w 316  1024x768x15 VESA
x 313  800x600x15  VESA  y 319 1280x1024x15 VESA  z 30D  320x200x15  VESA
```

For example, set VGA as 80x60 frame buffer mode:

```
APPEND root=/dev/hda2 vga=0xf07  (or using vga=ask to get list like above table)
```

Running graphic application based on frame buffer can work properly after changing APPEND line. Programmer also can visit http://tldp.org/HOWTO/Framebuffer-HOWTO.html for more VGA modes.

8.13. 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/kernel-2.6.x-vertex86dx.config is kernel configuration file for Vortex86DX Linux kernel 2.6.x) 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 is also at the same directory. Developer can use BusyBox configuration file to make his own BusyBox.

8.14. Develop Application

X-Linux is a run-time environment for developers and does not provide tool-chain. Developers have to make their programs on desktop PC and put them onto X-Linux to run. Refer to below section for more.

8.15. Install Application

Developers can put their program onto X-Linux device via FTP or NFS. Before running it, use **ldd** command on development workstation to check dependency files. Also put relative files onto X-Linux to ensure program can run properly. Here is an example when we put "syslinux" onto X-Linux:

```
[root@X-Linux]:/sbin # ldd syslinux
        linux-gate.so.1 =>  (0xb80a0000)
        libc.so.6 => /lib/libc.so.6 (0xb7f60000)
        /lib/ld-linux.so.2 (0xb80a1000)
[root@X-Linux]:/sbin #
```

From above messages, **/lib/libc.so.6** and **/lib/ld-linux.so.2** are needed by syslinux. Put those two files onto X-Linux to ensure syslinux can work properly.

For **linux-gate.so.1**, refer to http://www.trilithium.com/johan/2005/08/linux-gate/.

9. Graphic Solution

We will not provide X-Linux with X-Window for our embedded products now. This is because the X-Window is complexity and we reduce function/size to add it into X-Linux will have compatible problems any X-Window system. If you need tiny graphic solution in your OEM/ODM project, you can contact your DMP sales for customized technical support. For general graphic solution, you can try Puppy Linux (http://www.puppylinux.org/) or other popular Linux (ex: Debina with graphic interface). For that, maybe you need 512MB to 1GB storage.

10. Reference Links

Linux Kernel	http://www.kernel.org/
SysLinux	http://syslinux.zytor.com/
BusyBox	http://www.busybox.net/
DropBear	http://matt.ucc.asn.au/dropbear/dropbear.html

11. Technical Support

For more technical support, please visit http://www.dmp.com.tw/tech or mail to soc@dmp.com.tw.