V2416A Linux User's Manual

Edition 1.0, September 2015

www.moxa.com/product

V2416A Linux User's Manual

Copyright Notice

Trademarks

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas
Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa China (Shanghai office)
Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-21-5258-5505

Moxa Europe
Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa Asia-Pacific
Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

1

Introduction

Thank you for purchasing the Moxa V2416A series of x86 ready-to-run embedded computers. This manual introduces the software configuration and management of the V2416A, which runs the Linux operating system. For hardware installation, connector interfaces, setup, and upgrading the BIOS, please refer to the "V2416A Hardware User's Manual."

Linux is an open, scalable operating system that allows you to build a wide range of innovative, small footprint devices. Software written for desktop PCs can be easily ported to the embedded computer with a GNU cross compiler and a minimum of source code modifications. A typical Linux-based device is designed for a specific use, and is often not connected to other computers, or a number of such devices connect to a centralized, front-end host. Examples include enterprise tools such as industrial controllers, communications hubs, point-of-sale terminals, and display devices, which include HMIs, advertisement appliances, and interactive panels.

The following topics are covered in this chapter:

- **Overview**
- **Software Specifications**
- **Software Components**

Overview

V2416A series EN 50155-certified embedded computers are based on the Intel® Celeron® 1047UE Processor or Intel® Core™ i7-3517UE Processor. The V2416A computers feature four serial ports, dual 10/100 Mbps or 10/100/1000 LAN ports, three USB 2.0 hosts, and come with two DVI-I outputs. with EN 50155 certification, the computers are robust enough for railway and industrial applications.

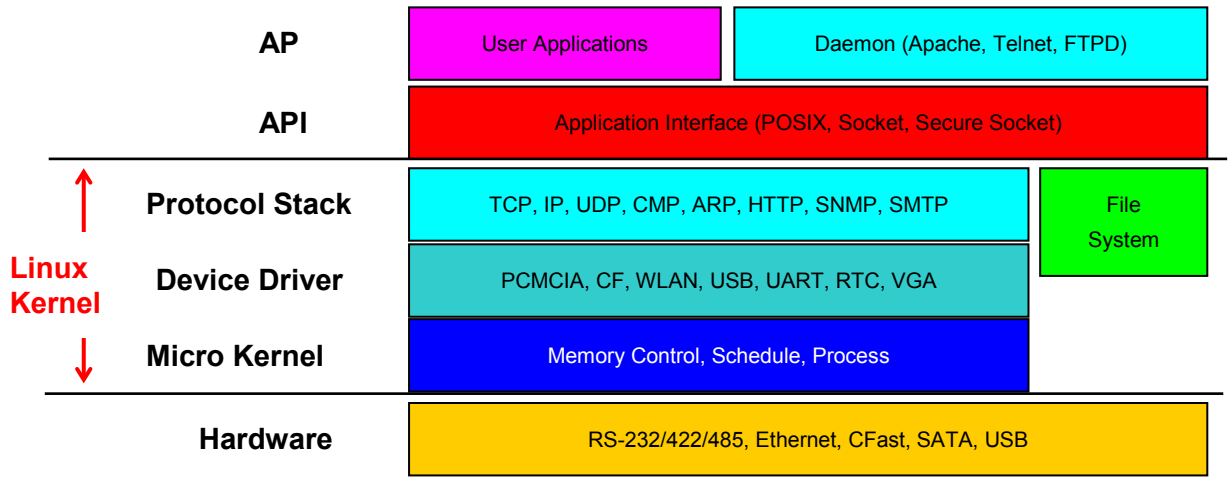
The V2416A's four serial ports make it ideal for connecting a wide range of serial devices, and the dual 10/100 Mbps or 10/100/1000 Ethernet ports offer a reliable solution for network redundancy, which taken together promise continuous data communication and management operations. For added convenience, the V2416A computers have 6 DIs and 2 Dos. In addition, the USB ports provide V2416A computers with data buffering and storage expansion, which provide the necessary reliability for industrial applications.

Pre-installed with Linux, the V2416A series provides programmers with a friendly environment for developing sophisticated, bug-free application software at a lower cost.

All V2416A models support a wide operating temperature range of -40 to 70°C for use in harsh industrial environments.

Software Specifications

The Linux operating system pre-installed on the V2416A embedded computers (CTO models) is the **Debian Wheezy 7.8** distribution. The Debian project involves a worldwide group of volunteers who endeavor to produce an operating system distribution composed entirely of free software. The Debian GNU/Linux follows the standard Linux architecture, making it easy to use programs that meet the POSIX standard. Program porting can be done with the GNU Tool Chain provided by Moxa. In addition to Standard POSIX APIs, device drivers for Moxa UART and other special peripherals are also included. An example software architecture is shown below:



ATTENTION

Refer to http://www.debian.org/ and http://www.gnu.org/ for information and documentation related to Debian GNU/Linux and the free software concept.

ATTENTION

The above software architecture is only an example. Different models or different build revisions of the Linux operating system may include components not shown in the above graphic.

Software Components

The V2416A Linux models are pre-installed with the Debian Wheezy7.8 Linux distribution. For the software components, see "Appendix A."

2

Software Configuration

In this chapter, we explain how to operate a V2416A-LX computer directly from your desktop. There are two ways to connect to the V2416A-LX computer: through a DVI monitor, or via an SSH console from a Windows or Linux machine. This chapter describes basic Linux operating system configurations. Advanced network management and configuration instructions will be described in the next chapter, "Managing Communications."

The following topics are covered in this chapter:

- **Account Management**
- **Starting from a DVI Console**
- **Setting up a Desktop Environment**
- **Connecting from an SSH Console**
  - Windows Users
  - Linux Users
- **Adjusting the System Time**
  - Setting the Time Manually
  - NTP Client
  - Updating the Time Automatically
- **Enabling and Disabling Daemons**
- **Cron—Daemon for Executing Scheduled Commands**
- **Inserting a USB Storage Device into the Computer**
- **Audio Playback and Recording**
- **Checking the Linux Version**
- **APT—Installing and Removing Packages**
- **Wake on LAN**

Account Management

Connect the V2416A to a display, turn on the computer, and then enter the following information to log in the computer.

Login: moxa
Password: moxa

For security reasons, the root account is already disabled. We strongly suggest changing the password during the first login. After successfully logging in, you can set up a new password.

```
login as: moxa
moxa@192.168.27.42's password:
You are required to change your password immediately (root enforced)
Linux Moxa 3.16.0-0.bpo.4-amd64 #1 SMP Debian 3.16.7-ckt4-3~bpo70+1 (2015-02-12) x86_64

    ####       ####    ######    ####### ######     ##
     ###      ####   ###    ###    ####   ####     ###
      ###     ###   ###     ###    ###    ##      ###
      ###    ####   ##       ##    ###   #       ####
      ####    # ##  ###       ###    ### ##       ## ##
     ## ##    # ##  ###        ##     ####        #  ##
     ## ###  ## ##  ##         ##     ####        #  ###
     ##  ##  #  ##  ##         ##      ###       #######
     ##  ##  #  ##  ###        ###     #####      #    ##
     ##   ###   ##  ###        ###    ##  ###     #    ###
     ##   ###   ##   ##        ##     ##   ###    ##    ##
     ##   ###   ##    ##      ##     #    ###    #     ##
   ######  #  ######   ########   ####### ##########  ######

For further information check:
http://www.moxa.com/

moxa@Moxa:~#
```

When you finish changing the password, remember to type **sudo** each time you want to run commands with privilege as the root. For example, typing **sudo ifconfig eth0 192.168.100.100** will allow you to configure the IP address of the LAN 1 port.

Starting from a DVI Console

Connect the display monitor to the V2416A-LX DVI connector, and then power it up by connecting it to the power adaptor. It takes approximately 30 to 60 seconds for the system to boot up. Once the system is ready, a login screen will appear on your monitor.

To log in, type the login name and password as requested. The default values are both **moxa**.

Login: moxa
Password: moxa

```
Moxa login: moxa
Password:
Linux Moxa 3.16.0-0.bpo.4-amd64 #1 SMP Debian 3.16.7-ckt4-3~bpo70+1 (2015-02-12)
x86_64

    ####       ####    ######    ####### ######     ##
     ###      ####   ###    ###    ####   ####     ###
      ###     ###   ###     ###    ###    ##      ###
      ###    ####   ##       ##    ###   #       ####
      ####    # ##  ###       ###    ### ##       ## ##
     ## ##    # ##  ###        ##     ####        #  ##
     ## ###  ## ##  ##         ##     ####        #  ###
     ##  ##  #  ##  ##         ##      ###       #######
     ##  ##  #  ##  ###        ###     #####      #    ##
     ##   ###   ##  ###        ###    ##  ###     #    ###
     ##   ###   ##   ##        ##     ##   ###    ##    ##
     ##   ###   ##    ##      ##     #    ###    #     ##
    ######  #  ######   ########   ####### ##########  ######

For further information check:
http://www.moxa.com/
```

Setting up a Desktop Environment

This section introduces the desktop environment for the V2416A series. By default, the V2416A Linux operating system models do not install a desktop environment. Debian supports all kinds of fully-featured graphical environments, such as Gnome, KDE, and lighter environments like Xfce and LXDE. You can choose to install one of these desktop systems on the V2416A. To do this, use the following commands:

To install Gnome:

```
moxa@MOXA:~# sudo apt-get install gnome-core
```

To install KDE:

```
moxa@MOXA:~# sudo apt-get install kde-standard
```

To install Xfce:

```
moxa@MOXA:~# apt-get install xfce4 xfce4-goodies thunar-archive-plugin
```

To install the minimum LXDE:

```
moxa@MOXA:~# sudo apt-get install lxde-core
```

Connecting from an SSH Console

The V2416A computers support the SSH console to offer users better network security compared to Telnet. The default IP addresses and netmasks of the network interfaces are as follows:

	Default IP Address	Netmask
LAN 1	192.168.3.127	255.255.255.0
LAN 2	192.168.4.127	255.255.255.0

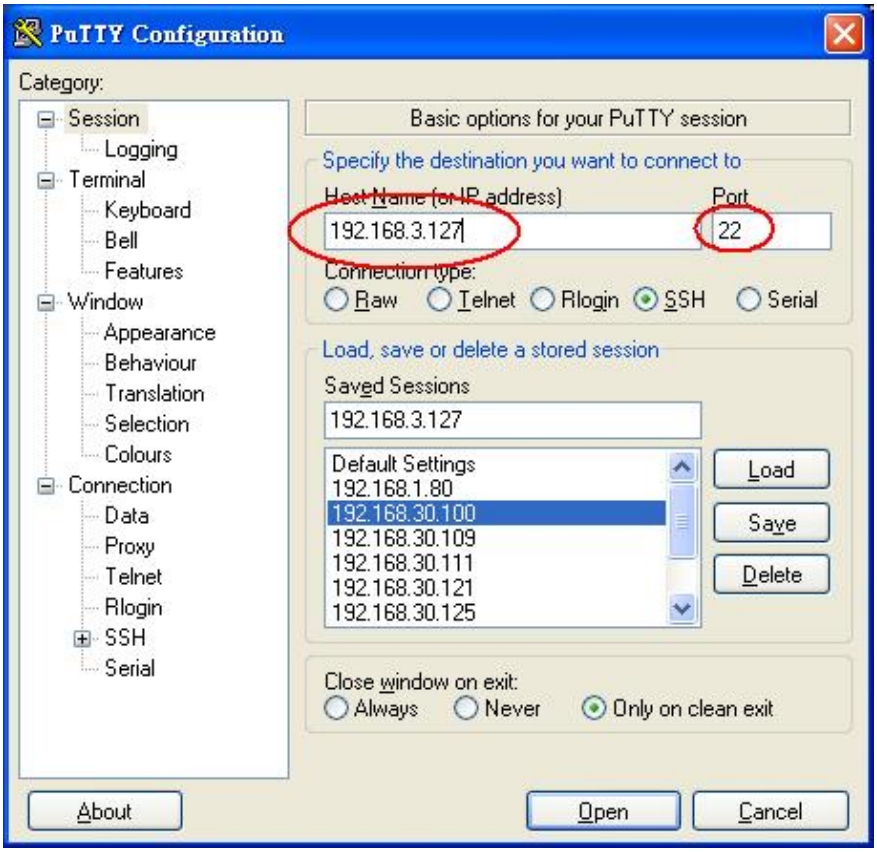
Before using the ssh client, you should change the IP address of your development workstation so that the network ports are on the same subnet as the IP address for the LAN port that you will connect to. For example, if you connect to LAN1, you could set your PC's IP address to 192.168.3.126, and the netmask to 255.255.255.0. If you connect to LAN2, you could set your PC's IP address to 192.168.4.126, and the netmask to 255.255.255.0.

Use a cross-over Ethernet cable to connect your development workstation directly to the target computer, or use a straight-through Ethernet cable to connect the computer to a LAN hub or switch. Next, use a Telnet client on your development workstation to connect to the target computer. After a connection has been established, type the login name and password as requested to log on to the computer. The default values are both **moxa**.

Login: moxa
Password: moxa

Windows Users

Click on the link http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html to download **PuTTY** (free software) to set up an SSH console for the V2416A in a Windows environment. The following screen shows an example of the configuration that is required.



Linux Users

From a Linux machine, use the **ssh** command to access the V2416A-LX's console utility via SSH.

 # ssh 192.168.3.127

Select **yes** to open the connection.

```
[root@bee_notebook root]# ssh 192.168.3.127
The authenticity of host '192.168.3.127 (192.168.3.127)' can't be established.
RSA key fingerprint is 8b:ee:ff:84:41:25:fc:cd:2a:f2:92:8f:cb:1f:6b:2f.
Are you sure you want to continue connection (yes/no)? yes_
```

Adjusting the System Time

The V2416A-LX has two time settings. One is the system time, and the other is provided by an RTC (Real Time Clock) built into the V2416A-LX's hardware.

Setting the Time Manually

Use the **date** command to query the current system time or to set a new system time. Use **hwclock** to query the current RTC time or to set a new RTC time.

Use the following command to set the system time.
 # date MMDDhhmmYYYY

MM: Month
DD: Date
hhmm: Hour and Minute
YYYY: Year

Use the following command to write the current system time to the RTC.
 # hwclock –w

```
root@Moxa:~# date
Wed Dec 16 03:34:46 CST 2009
root@Moxa:~# hwclock
Wed 16 Dec 2009 03:35:16 AM CST  -0.017600 seconds
root@Moxa:~# date 121616352009
Wed Dec 16 16:35:00 CST 2009
root@Moxa:~# hwclock -w
root@Moxa:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2009
Wed 16 Dec 2009 04:36:13 PM CST -0.016751 seconds
root@Moxa:~#
```

NTP Client

The V2416A-LX has a built-in NTP (Network Time Protocol) client that is used to initialize a time request to a remote NTP server. Use **ntpdate** to update the system time.

#ntpdate time.stdtime.gov.tw
#hwclock –w

Visit http://www.ntp.org for more information about NTP and NTP server addresses.

```
root@Moxa:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2009
Wed 16 Dec 2009 04:36:13 PM CST -0.016751 seconds
root@Moxa:~#
root@Moxa:~# ntpdate time.stdtime.gov.tw
 16 Dec 03:49:48 ntpdate[2510]: step time server 220.130.158.52 offset 155905087.9
84256 sec
root@Moxa:~#
root@Moxa:~# hwclock -w
root@Moxa:~# date ; hwclock
Wed Dec 16 03:51:07 CST 2009
Wed 16 Dec 2009 03:51:07 AM CST   -0.016771 seconds
root@Moxa:~#
```

ATTENTION

Before using the NTP client utility, check your IP address and network settings (gateway and DNS) to make sure an Internet connection is available.

Updating the Time Automatically

This section describes how to use a shell script to update the time automatically.

Example shell script for updating the system time periodically

```
#!/bin/sh
ntpdate time.stdtime.gov.tw
# You can use the time server's ip address or domain
# name directly. If you use domain name, you must
# enable the domain client on the system by updating
# /etc/resolv.conf file.
hwclock –w
sleep 100
# Updates every 100 seconds. The min. time is 100 seconds.
# Change 100 to a larger number to update RTC less often.
```

Save the shell script using any file name. For example, **fixtime**.

How to run the shell script automatically when the kernel boots up

Copy the example shell script **fixtime** to the directory **/etc/init.d**, and then use **chmod 755 fixtime** to change the shell script mode.

 # chmod 755 fixtime

Next, use **vi** editor to edit the file **/etc/inittab**.

 # vi /etc/inittab

Add the following line to the bottom of the file:

ntp : 2345 : respawn : /etc/init.d/fixtime

After you finish writing or modifying the code, remember to execute **umount /** to change the root directory back to Read-only mode.

 # umount /

Use the command **#init q** to re-initialize the kernel.

 # init q

Enabling and Disabling Daemons

Only the following daemons are enabled in the V2416A-LX by default:

sftpd SFTP Server / Client daemon

sshd Secure Shell Server daemon

You may manage what services to run in the background by the command insserv. Below example shows how to add the apache daemon in current runlevel.

```
moxa@Moxa:~$ sudo insserv –d apache2
```

Apache will not activate in the current boot session, but will be running in the background from the next boot session. To disable the apache daemon, use the following command:

```
moxa@Moxa:~$ sudo insserv -r apache2
```

Linux daemons can be started or stopped in the current boot session by using of the scripts in /etc/init.d. To start the apache daemon, use:

```
moxa@Moxa:~$ sudo /etc/init.d/apache2 start
```

To stop the apache daemon, use:

```
moxa@Moxa:~$ sudo /etc/init.d/apache2 stop
```

1. To run a private daemon, you can edit the file **rc.loca**l, as shown below (type cd /etc/ to change directories):

   ```
   root@Moxa:~# cd /etc/
   ```

2. Type **vi rc.local** to edit the configuration file with vi editor.

   ```
   root@Moxa:/etc/# vi rc.local
   ```

3. Next, add the application daemon that you want to run. The following example shows how to add a daemon that is placed in the **/home/** directory. The daemon will run in the background during system boot.

   ```
   # !/bin/sh
   # Add you want to run daemon
   home/daemon &~
   ```

Cron—Daemon for Executing Scheduled Commands

The Cron daemon will search **/etc/crontab** for crontab files. Cron wakes up every minute and checks each command to see if it should be run at that time. When executing commands, output is mailed to the owner of the **crontab** (or to the user named in the MAILTO environment variable in the **crontab**, if such a user exists).

Modify the file **/etc/crontab** to set up your scheduled applications. **Crontab** files have the following format:

mm	h	dom	mon	dow	user	command
minute	hour	date	month	week	user	command
0-59	0-23	1-31	1-12	0-6 (0 is Sunday)		

For example, issue the following command if you want to launch a program at 8:00 every day:

```
#minute hour  date  month  week  user   command
*        8    *     *      *     root   /path/to/your/program
```

The following example demonstrates how to use **Cron** to update the system time and RTC time every day at 8:00.

1. Write a shell script named fixtime.sh and save it to **/home/**.
   #!/bin/sh
   ntpdate time.stdtime.gov.tw
   hwclock –w
   exit 0
2. Change the mode of **fixtime.sh**
   # chmod 755 fixtime.sh
3. Modify the **/etc/crontab** file to run fixtime.sh at 8:00 every day.
   Add the following line to the end of crontab:
   *** 8 * * *root /home/fixtime.sh**

Inserting a USB Storage Device into the Computer

Since mounting USB storage devices manually can be difficult, a Debian package named **usbmount** is used to mount the USB drivers automatically. **usbmount** relies on **udev** to mount USB storage devices automatically at certain mount points. The USB storage devices will be mounted on **/media/usb0, /media/usb1**, etc.

```
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=494659,mode=755)
devpts on /dev/pts type devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=397292k,mode=755)
/dev/disk/by-label/V2400A_MOXA on / type ext4
(rw,noatime,errors=remount-ro,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=794560k)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
(rw,noexec,nodev,sync,noatime,gid=25,dmask=0007,fmask=0117)
/dev/sdb1 on /media/usb1 type vfat
```

```
(rw,noexec,nodev,sync,noatime,gid=25,dmask=0007,fmask=0117)
root@MOXA:~#
```

Note that usbmount is a light-weight solution for text mode, and does not fully support the gnome desktop environment. For better supportability, install gnome-volume-manager instead of usbmount:

```
root@MOXA:~# apt-get remove usbmount
root@MOXA:~# apt-get install gnome-volume-manager
```

ATTENTION

Remember to type the command # sync before you disconnect the USB storage device. If you do not issue the command, you may lose data.

ATTENTION

Remember to exit the /media/usb0 or /media/usb1 directory when you disconnect the USB storage device. If you stay in /media/usb0 or /media/usb1, the automatic un-mount process will fail. If that happens, type # umount /media/usb0 to un-mount the USB device manually.

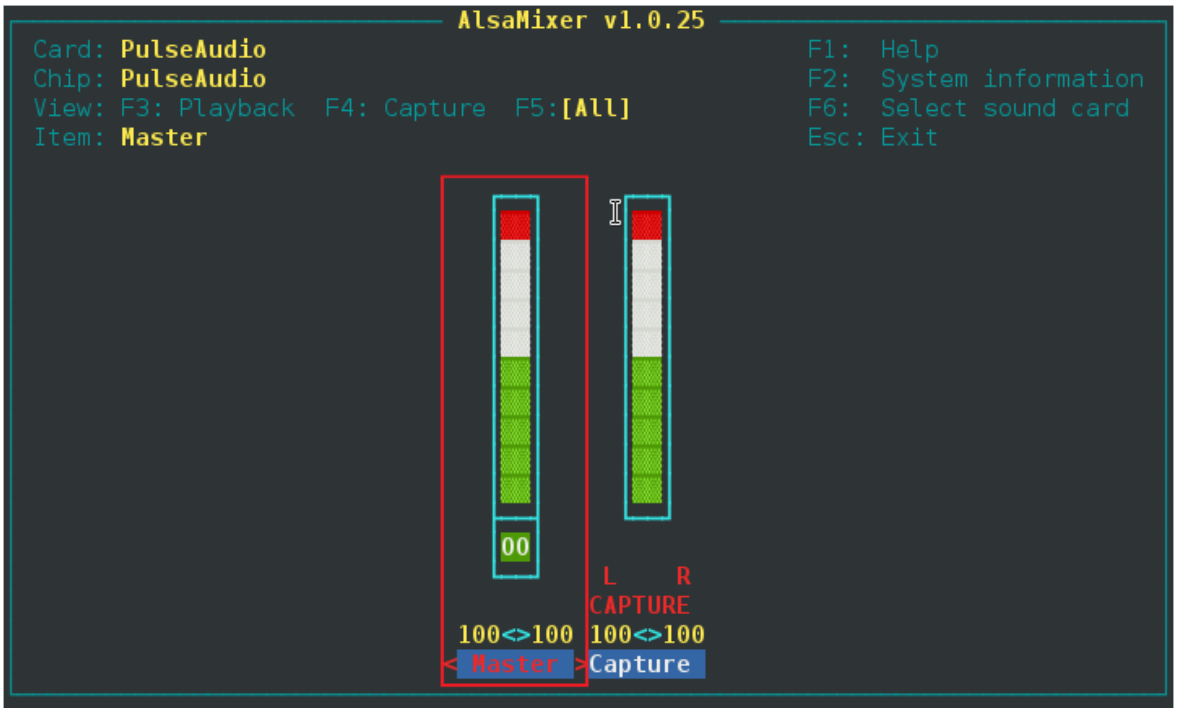
Audio Playback and Recording

The V2416A has a built-in audio system that provides the Line-in, line-out interface in M12 format for audio playback or recording. Use the following commands to play back audio on the V2416A.

Control the volumn of the Master with alsamixer

```
root@Moxa:~# alsamixer
```

Press ***<TAB>*** to select the Master as the playback source and press <***UP***> and <***DOWN***> to adjust the volume.



Play a wave file

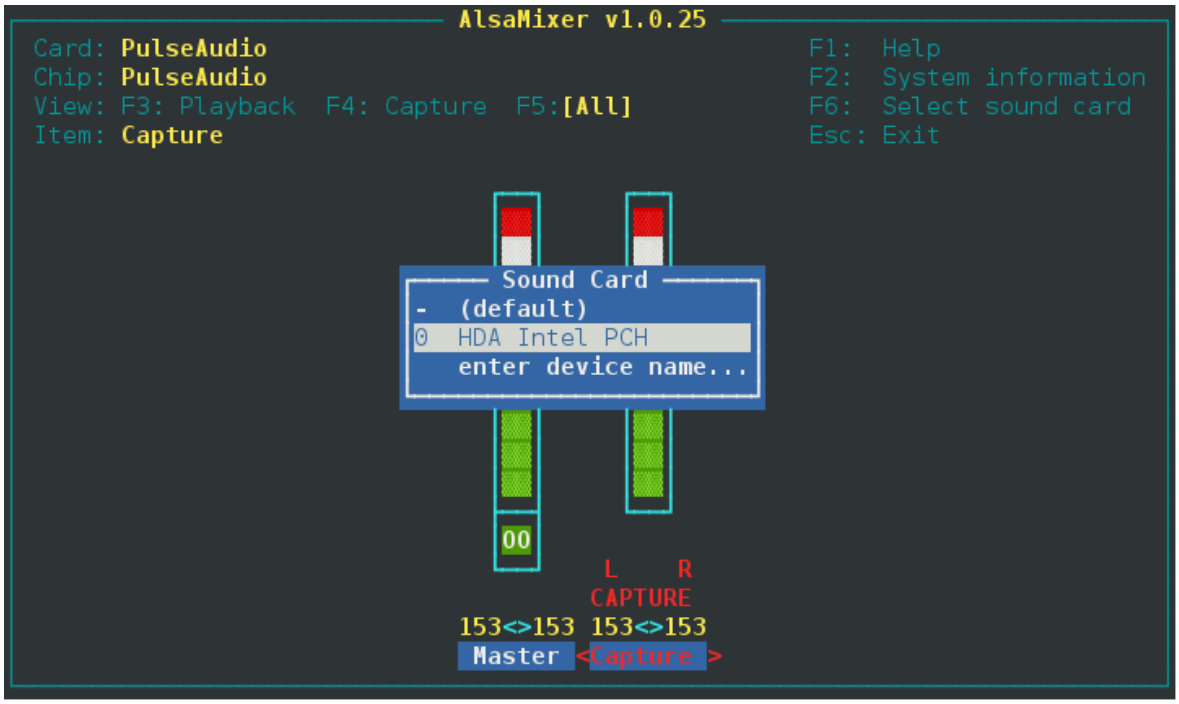
```
root@Moxa:~# aplay -t wav /PATH/TO/test.wav
```

The V2416A has a Line-in interface for recording audio. Use the following commands to record audio on the V2416A.

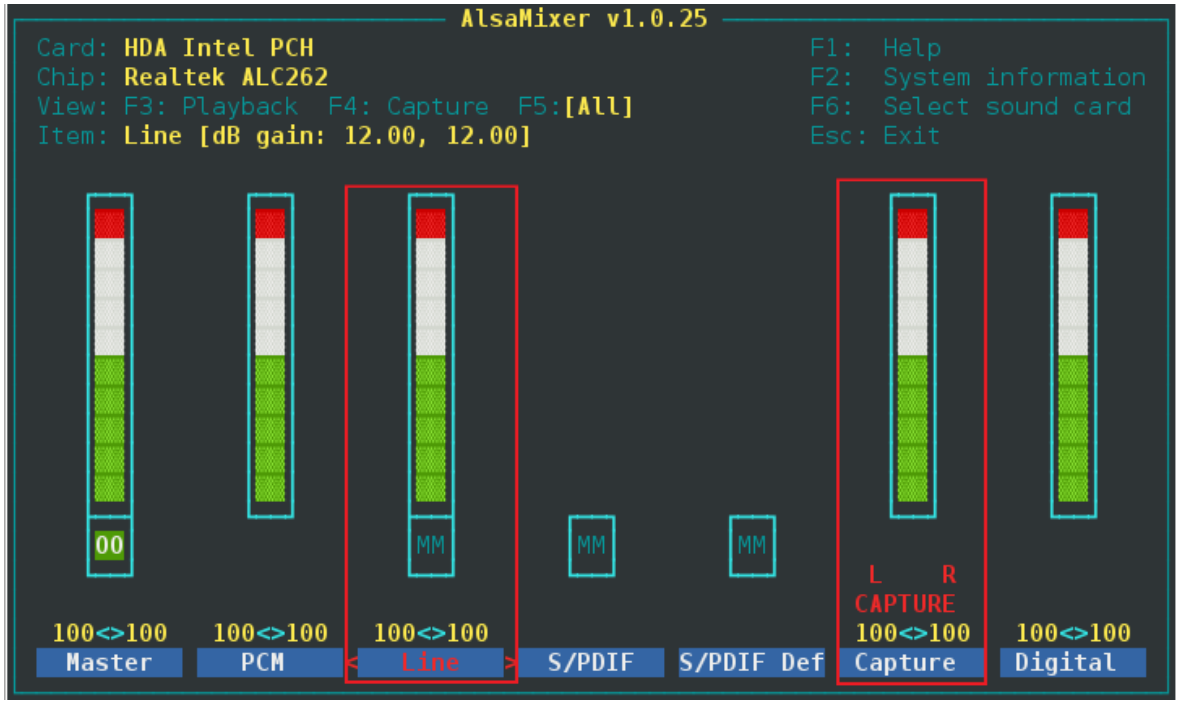
Control the volume of Capture with alsamixer

```
root@Moxa:~# alsamixer
```

Press <***F6***> to Select the device - "HDA Intel PCH"



Press ***<TAB>*** to select the Line and Capture as capture and press <***UP***> and <***DOWN***> to adjust the volume.



Record the audio in wave format

```
moxa@MOXA:~# arecord -t wav -f cd -c 2 /dev/shm/aaa.wav
```

Checking the Linux Version

The program **uname**, which stands for "Unix Name" and is part of the Unix operating system, prints the name, version, and other details about the operating system running on the computer. Use the -**a** option to generate a response similar to the one shown below:

```
root@Moxa:~# uname -a
Linux Moxa 3.16.0-0.bpo.4-amd64 #1 SMP Debian 3.16.7-ckt4-3~bpo70+1 (2015-02-12)
x86_64 GNU/Linux
root@Moxa:~#
```

APT—Installing and Removing Packages

APT is the Debian tool used to install and remove packages. Before installing a package, you need to configure the apt source file, **/etc/apt/sources.list**, which is not a read only partition.

1. Use vi editor to configure **/etc/apt/sources.list**.

```
root@Moxa:~# vi /etc/apt/sources.list

deb http://debian.moxa.com/debian wheezy main

deb http://ftp.us.debian.org/debian/ wheezy main contrib non-free
deb-src http://ftp.us.debian.org/debian/ wheezy main contrib non-free

deb http://ftp.us.debian.org/debian/ wheezy-updates main contrib non-free
deb-src http://ftp.us.debian.org/debian/ wheezy-updates main contrib non-free

deb http://security.debian.org/ wheezy/updates main contrib non-free
deb-src http://security.debian.org/ wheezy/updates main contrib non-free

deb http://ftp.debian.org/debian wheezy-backports main contrib non-free
deb-src http://ftp.debian.org/debian wheezy-backports main contrib non-free
```

2. Add Moxa's apt repository
   For adding or updating the drivers, libraries, and utilities provided by Moxa.
   "deb http://debian.moxa.com/debian wheezy main" is added to the source list by default. If you delete it, You should add "deb http://debian.moxa.com/debian wheezy main" to the source list.

```
root@Moxa:~# sudo vi /etc/apt/sources.list
deb http://debian.moxa.com/debian wheezy main
```

Moxa has encrypted the packages with a GPG key that allows you to check if the package was verified by us. The GPG key should be added before starting the installation. You can find the GPG key on the CD or on Moxa's website. Upload the key to the Moxa embedded computer and then add it.

Check your GPG key list to verify that "MOXA SYS" is in the list.

```
root@Moxa:~# apt-key list
/etc/apt/trusted.gpg
--------------------
pub   2048R/62B24532 2014-05-28 [expires: 2024-05-25]
uid                  MOXA SYS <sys.support@moxa.com>
sub   2048R/F7F3CD9E 2014-05-28 [expires: 2024-05-25]
```

If "MOXA SYS" is not in your list, can add it into the GPG key list.

```
root@Moxa:~# apt-key add NEW-MOXA-SYS-DEBIAN-KEY
```

Next, update the package list to display the most up-to-date package list.

```
root@Moxa:~# apt-get update
Ign http://debian.moxa.com wheezy Release.gpg
Ign http://debian.moxa.com/debian/ wheezy/main Translation-en
Ign http://debian.moxa.com/debian/ wheezy/main Translation-en_HK
Get:1 http://debian.moxa.com wheezy Release [1,633 B]
Ign http://debian.moxa.com wheezy/main i386 Packages
Get:2 http://debian.moxa.com wheezy/main i386 Packages [1,585 B]
Fetched 3,218 B in 0s (47.2 kB/s)
Reading package lists... Done
```

After updating the package list, use apt-get to install or upgrade the packages from Moxa's apt repository.

1. Update the source list after you configure it.

```
moxa@MOXA:~# sudo apt-get update
moxa@MOXA:~#
```

2. Once you indicate which package you want to install (**ipsec-tools**, for example), type:

```
moxa@MOXA:~# sudo apt-get install ipsec-tools
moxa@MOXA:~#
```

3. Use one of the following commands to remove a package:
   a. For a simple package removal:

```
moxa@MOXA:~# sudo apt-get remove ipsec-tools
moxa@MOXA:~#
```

   b. For a complete package removal:

```
moxa@MOXA:~# sudo apt-get remove ipsec-tools --purge
moxa@MOXA:~#
```

ATTENTION

The APT cache space /var/cache/apt is located in tmpfs. If you need to install a huge package, link /var/cache/apt to a USB mass storage device or mount it to an NFS space to generate more free space. Use df –h to check how much free space is available in tmpfs.

```
moxa@MOXA:~# df -h
Filesystem          Size  Used Avail Use% Mounted on
rootfs                     7.3G  923M  6.2G  13% /
udev                        10M     0   10M   0% /dev
tmpfs                      388M  184K  388M   1% /run
/dev/disk/by-label/V2400A_MOXA  7.3G  923M  6.2G  13% /
tmpfs                      5.0M     0  5.0M   0% /run/lock
tmpfs                      776M     0  776M   0% /run/shm
moxa@MOXA:~#
```

ATTENTION

Use the command **# apt-get clean** to free up the cache space.

```
moxa@MOXA:~# apt-get clean
moxa@MOXA:~#
```

Wake on LAN

The V2416A-LX supports wake on LAN, a feature used to wake up a device for suspend (S3) and shutdown (S5).

To check the WOL support on Ethernet port x, type **ethtool ethx**, where "x" is the port number.

```
root@Moxa:/# ethtool eth0
Settings for eth0:
       Supported ports: [ TP ]
       Supported link modes:   10baseT/Half 10baseT/Full
                               100baseT/Half 100baseT/Full
                               1000baseT/Full
       Supports auto-negotiation: Yes
       Advertised link modes:  10baseT/Half 10baseT/Full
                               100baseT/Half 100baseT/Full
                               1000baseT/Full
       Advertised auto-negotiation: Yes
       Speed: 100Mb/s
       Duplex: Full
       Port: Twisted Pair
       PHYAD: 0
       Transceiver: internal
       Auto-negotiation: on
       Supports Wake-on: pumbg
       Wake-on: ug
       Current message level: 0x00000033 (51)
       Link detected: yes
Moxa:/#
```

As you can see, the default WOL support is g (wake on Magic packet).

The following example illustrates how to wake up on suspend (S3):

1. Moxa's embedded computer
   Get its MAC by issuing "ifconfig ethx" (x is the port number)
   Suspend to RAM with command "pm-suspend --quirk-s3-bios"
   Remote computer
   Issue the command **etherwake -b mac_of_this_device** to wake it up. For example:

```
etherwake -b 00:90:e8:00:d7:07
```

The following example illustrates how to wake up on shutdown (S5):

1. Moxa's embedded computer
   Shut down your computer with "shutdown -h now"
2. Remote computer
   Issue the command **etherwake -b mac_of_this_device** to wake it up. For example:

```
etherwake -b 00:90:e8:00:d7:07
```

3

Managing Communications

The V2416A-LX ready-to-run embedded computer is a network-centric platform designed to serve as a front-end for data acquisition and industrial control applications. This chapter describes how to configure the various communication functions supported by the Linux operating system.

The following topics are covered in this chapter:

- **Detecting Network Interfaces**
- **Changing the Network Settings**
  - Changing the "interfaces" Configuration File
  - Adjusting IP Addresses with "ifconfig"
- **Serial Port Operation Mode**
- **DNS Client**
  - /etc/hostname
  - /etc/resolv.conf
  - /etc/nsswitch.conf
- **Configuring Ethernet Bonding**
- **Apache Web Server**
  - Default Homepage
- **IPTABLES**
  - IPTABLES Hierarchy
  - IPTABLES Modules
  - Observe and Erase Chain Rules
  - Define Policy for Chain Rules
  - Append or Delete Rules
- **NAT (Network Address Translation)**
  - NAT Example
  - Enabling NAT at Bootup
- **PPP (Point to Point Protocol)**
  - Connecting to a PPP Server over a Simple Dial-up Connection
  - Connecting to a PPP Server over a Hard-wired Link
  - Checking the Connection
  - Setting up a Machine for Incoming PPP Connections
- **PPPoE**
- **NFS (Network File System) Client**
- **SNMP**
- **OpenVPN**
  - Ethernet Bridging for Private Networks on Different Subnets
  - Ethernet Bridging for Private Networks on the Same Subnet
  - Routed IP

Detecting Network Interfaces

Debian Linux systems use **udevd** to detect new network interfaces, including Ethernet interfaces and wireless interfaces. One of the rules is **/lib/udev/rules.d/75-persistent-net-generator.rules** for creating a persistent network interface naming order. The content in **/etc/udev/rules.d/70-persistent-net.rules** is similar to the following:

```
# PCI device 0x10ec:/sys/devices/pci0000:00/0000:00:1c.1/0000:02:00.0 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:90:e8:00:00:20", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"

# PCI device 0x10ec:/sys/devices/pci0000:00/0000:00:1c.0/0000:01:00.0 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:90:e8:00:00:21", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth1"
```

The above example indicates that the system has detected two Ethernet interfaces.

ATTENTION

When replacing or connecting a network interface, the system may keep the old record in **/etc/udev/rules.d/70-persistent-net.rules**, which could cause network interfaces to be detected abnormally. To avoid this problem, delete the content of the file /etc/udev/rules.d/70-persistent-net.rules and reboot the system.

Changing the Network Settings

The V2416A-LX computer has two 10/100 or 10/100/1000 Ethernet ports named LAN1 and LAN2. The default IP addresses and netmasks of these network interfaces are:

	Default IP Address	Netmask
LAN1	192.168.3.127	255.255.255.0
LAN2	192.168.4.127	255.255.255.0

These network settings can be modified by changing the **interfaces** configuration file, or they can be adjusted temporarily with the **ifconfig** command.

Changing the "interfaces" Configuration File

1. Type **cd /etc/network** to change directories.

```
root@MOXA:~# cd /etc/network
```

2. Type **vi interfaces** to edit the network configuration file with **vi** editor. You can configure the V2416A-LX's Ethernet ports for static or dynamic (DHCP) IP addresses.

```
root@MOXA:/etc/network# vi interfaces
```

Static IP Address

As shown in the following example, the default static IP addresses can be modified.

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
       address 192.168.3.127
       netmask 255.255.255.0
       broadcast 192.168.3.255

auto eth1
iface eth1 inet static
       address 192.168.4.127
       netmask 255.255.255.0
       broadcast 192.168.4.255
```

Dynamic IP Address using DHCP

To configure one or both LAN ports to request an IP address dynamically, replace **static** with **dhcp** and then delete the rest of the lines.

```
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

After modifying the boot settings of the LAN interface, issue the following command to activate the LAN settings immediately.

 # /etc/init.d/networking restart

```
root@MOXA:~# /etc/init.d/networking restart
```

Adjusting IP Addresses with "ifconfig"

IP settings can be adjusted during run-time, but the new settings will not be saved to the flash ROM without modifying the file **/etc/network/interfaces**. For example, type the command **# ifconfig eth0 192.168.1.1** to change the IP address of LAN1 to 192.168.1.1.

```
root@MOXA:~# ifconfig eth0 192.168.1.1
root@MOXA:~#
```

Serial Port Operation Mode

The V2416A-LX computer has 4 serial ports named COM1, COM2, COM3, and COM4. The ports support RS-232, RS-422, 2-wire RS-485, and 4-wire RS-485 operation modes with baudrate settings up to 921600 bps.

By default, the serial interface is set to RS-232. You can use the **muestty** command to change the serial port operation mode, as indicated below:

```
muestty <operation> devicedevice-node: /dev/ttyMUEn; n = 0,1,2,...

<operation> [see following table]:

-h          Help
-g          Get information
-i intf     Set interface type
            intf     RS232        RS-232 mode
                     RS422        RS-422 mode
                     RS4852W      RS-485 2 wire mode
                     RS4854W      RS-485 4 wire mode
```

For example, use the following commands to set **/dev/ttyMUE0** to RS-422:

```
root@Moxa:~# muestty -i RS422 /dev/ttyMUE0
muestty: Set interface of /dev/ttyMUE0 ok.
root@Moxa:~# muestty -g /dev/ttyMUE0
muestty: /dev/ttyMUE0 is set to RS-422 mode.
root@Moxa:~#
```

DNS Client

The V2416A-LX supports DNS client (but not DNS server). To set up DNS client, you need to edit three configuration files: **/etc/hostname, /etc/resolv.conf,** and **/etc/nsswitch.conf**.

/etc/hostname

1. Edit **/etc/hostname**:

```
moxa@MOXA:~# sudo vi /etc/hostname
Moxa
```

2. Re-configure the hostname.

```
root@Moxa:~# /etc/init.d/hostname.sh start
```

3. Check the new hostname.

```
root@Moxa:~# hostname
```

/etc/resolv.conf

This is the most important file that you need to edit when using DNS. For example, before using **# ntpdate time.stdtime.gov.tw** to update the system time, you will need to add the DNS server address to the file. Ask your network administrator which DNS server address you should use. The DNS server's IP address is specified with the **nameserver** command. For example, add the following line to /etc/resolv.conf (assuming the DNS server's IP address is 168.95.1.1):

nameserver 168.95.1.1

```
root@MOXA:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16
nameserver 168.95.1.1
nameserver 140.115.1.31
nameserver 140.115.236.10
MOXA:/etc#
```

/etc/nsswitch.conf

This file defines the sequence of files, **/etc/hosts** or **/etc/resolv.conf**, to be read to resolve the IP address. The **hosts** line in **/etc/nsswitch.conf** means use **/etc/host** first and DNS service to resolve the address.

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Configuring Ethernet Bonding

The Linux bonding driver provides a method for aggregating multiple network interfaces into a single logical "bonded" interface. To use the bonding feature, load the bonding driver. Then use ifenslave to add the Ethernet interface into the bond0 interface. The following script bonds eth1 and eth2 together; you can place the script in **/etc/init.d/bonding**.

```
#! /bin/bash

#### BEGIN INIT INFO
# Provides:        bonding
# Short-Description: Start the bonding service, bond eth1 and eth2.
# Required-Start:   $all
# Required-Stop:    $all
# Should-Start:
```

```
# Should-Stop:
# Default-Start:    2 3 4 5
# Default-Stop:     0 1 6
### END INIT INFO

NAME=bonding
PATH=/bin:/usr/bin:/sbin:/usr/sbin

case "$1" in
  start)
    # to set ethX interfaces as slave the bond0 must have an ip
    if [ "$2" == "" ]; then
      $0
      exit 1
    fi
    echo "Starting bonding service: $NAME."
    modprobe bonding mode=1 miimon=100      # load bonding module

    ifdown eth0                 # putting down eth0
    ifdown eth1                 # putting down eth1

    ifconfig bond0 hw ether 00:90:E8:00:00:60   # change mac address
    ifconfig bond0 $2 netmask 255.255.255.0 up  # set ip address

    ifenslave bond0 eth0        # set eth0 in slave for bond0
    ifenslave bond0 eth1        # set eth1 in slave for bond0
    ;;

  stop)
    echo "Stopping bonding service: $NAME"
    ifenslave -d bond0 eth0         # release eth0 from bond0
    ifenslave -d bond0 eth1         # release eth1 from bond0

    ifconfig bond0 down             # putting down bond0
    modprobe -r bonding             # unload bonding module

    ifup eth0
    ifup eth1
    ;;

  restart)
    $0 stop
    $0 start $2
    ;;

  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|restart} [ip address]"
    exit 1
    ;;
esac

exit 0
```

You can use insserv to add this to run level.

```
moxa@MOXA:~# sudo insserv -v -d bonding
```

To remove it from run level, use the following command:

```
moxa@MOXA:~# sudo insserv -r bonding
```

Apache Web Server

Default Homepage

The Apache web server's main configuration file is **/etc/apache2/sites-enabled/000-default**, with the default homepage located at **/var/www/index.html**.

Save your own homepage to the following directory:

/var/www

Before you modify the homepage, use a browser (such as Microsoft Internet Explore or Mozilla Firefox) from your PC to test if the Apache web server is working. Type the LAN1 IP address in the browser's address box to open the homepage. For example, if the default IP address 192.168.3.127 is still active, type:

http://192.168.3.127/

ATTENTION

Visit the Apache website at http://httpd.apache.org/docs/ for more information about setting up Apache servers.

IPTABLES

IPTABLES is an administrative tool for setting up, maintaining, and inspecting the Linux kernel's IP packet filter rule tables. Several different tables are defined, with each table containing built-in chains and user-defined chains.

Each chain is a list of rules that apply to a certain type of packet. Each rule specifies what to do with a matching packet. A rule (such as a jump to a user-defined chain in the same table) is called a **target**.

The V2416A-LX supports three types of IPTABLES: Filter tables, NAT tables, and Mangle tables.

Filter Table—includes three chains:

- **INPUT chain**
- **OUTPUT chain**
- **FORWARD chain**

NAT Table—includes three chains:

- **PREROUTING chain**—transfers the destination IP address (DNAT).
- **POSTROUTING chain**—works after the routing process and before the Ethernet device process to transfer the source IP address (SNAT).
- **OUTPUT chain**—produces local packets.

Sub-tables

- **Source NAT (SNAT)**—changes the first source IP address of the packet.
- **Destination NAT (DNAT)**—changes the first destination IP address of the packet.

- **MASQUERADE**—a special form for SNAT. If one host can connect to the Internet, then the other computers that connect to this host can connect to the Internet when the computer does not have an actual IP address.
- **REDIRECT**—a special form of DNAT that re-sends packets to a local host independent of the destination IP address.

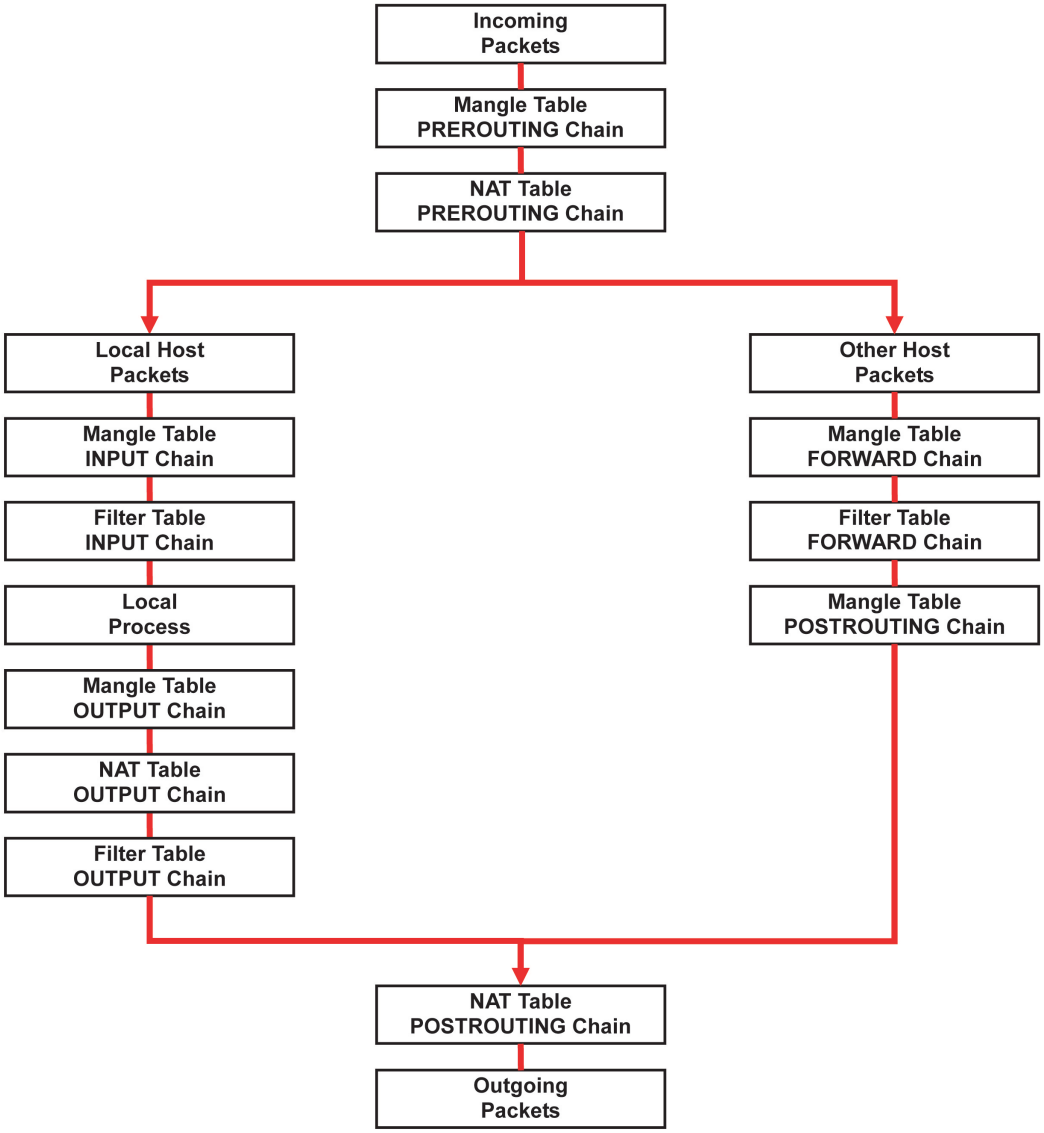
Mangle Table—includes two chains

- **PREROUTING chain**—pre-processes packets before the routing process.
- **OUTPUT chain**—processes packets after the routing process.

Mangle tables can have one of three extensions—TTL, MARK, TOS.

IPTABLES Hierarchy

The following figure shows the IPTABLES hierarchy.



IPTABLES Modules

The V2416A-LX supports the following sub-modules. Use the module that matches your application.

arptable_filter.ko	arp_tables.ko	arpt_mangle.ko	ip_conntrack_amanda.ko
ip_conntrack_ftp.ko	ip_conntrack_h323.ko	ip_conntrack_irc.ko	ip_conntrack.ko
ip_conntrack_netbios_ns.ko	ip_conntrack_netlink.ko	ip_conntrack_pptp.ko	ip_conntrack_proto_sctp.ko
ip_conntrack_sip.ko	ip_conntrack_tftp.ko	ip_nat_amanda.ko	ip_nat_ftp.ko
ip_nat_h323.ko	ip_nat_irc.ko	ip_nat.ko	ip_nat_pptp.ko
ip_nat_sip.ko	ip_nat_snmp_basic.ko	ip_nat_tftp.ko	ip_queue.ko
iptable_filter.ko	iptable_mangle.ko	iptable_nat.ko	iptable_raw.ko
ip_tables.ko	ipt_addrtype.ko	ipt_ah.ko	ipt_CLUSTERIP.ko
ipt_dscp.ko	ipt_DSCP.ko	ipt_ecn.ko	ipt_ECN.ko
ipt_hashlimit.ko	ipt_iprange.ko	ipt_LOG.ko	ipt_MASQUERADE.ko
ipt_NETMAP.ko	ipt_owner.ko	ipt_recent.ko	ipt_REDIRECT.ko
ipt_REJECT.ko	ipt_SAME.ko	ipt_TCPMSS.ko	ipt_tos.ko
ipt_TOS.ko	ipt_ttl.ko	ipt_TTL.ko	ipt_ULOG.ko

The basic syntax to enable and load an IPTABLES module is as follows:

```
# lsmod
# modprobe ip_tables
# modprobe iptable_filter
# modprobe iptable_mangle
# modprobe iptable_nat
```

Use **lsmod** to check if the **ip_tables** module has already been loaded in the V2416A-LX. Use **modprobe** to insert and enable the module.

Use **iptables**, **iptables-restore**, and **iptables-save** to maintain the database.

ATTENTION

IPTABLES plays the role of packet filtering or NAT. Be careful when setting up the IPTABLES rules. If the rules are not correct, remote hosts that connect via a LAN or PPP may be denied. We recommend using the VGA console to set up the IPTABLES. Click on the following links for more information about IPTABLES.

http://www.linuxguruz.com/iptables/
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

Since the IPTABLES command is very complex, to illustrate the IPTABLES syntax we have divided our discussion of the various rules into three categories: **Observe and erase chain rules**, **Define policy rules**, and **Append or delete rules**.

Observe and Erase Chain Rules

Usage:

 # iptables [-t tables] [-L] [-n]

-t tables: Table to manipulate (default: 'filter'); example: nat or filter.
-L [chain]: List List all rules in selected chains. If no chain is selected, all chains are listed.
-n: Numeric output of addresses and ports.

 # iptables [-t tables] [-FXZ]

-F: Flush the selected chain (all the chains in the table if none is listed).
-X: Delete the specified user-defined chain.
-Z: Set the packet and byte counters in all chains to zero.

Example:

 # iptables -L -n

In this example, since we do not use the -t parameter, the system uses the default "filter" table. Three chains are included: INPUT, OUTPUT, and FORWARD. INPUT chains are accepted automatically, and all connections are accepted without being filtered.

 # iptables –F
 # iptables –X
 # iptables –Z

Define Policy for Chain Rules

Usage:

 # iptables [-t tables] [-P] [INPUT, OUTPUT, FORWARD, PREROUTING, OUTPUT, POSTROUTING] [ACCEPT, DROP]

-P: Set the policy for the chain to the given target.
INPUT: For packets coming into the V2416A-LX.
OUTPUT: For locally-generated packets.
FORWARD: For packets routed out through the V2416A-LX.
PREROUTING: To alter packets as soon as they come in.
POSTROUTING: To alter packets as they are about to be sent out.

Example:

#iptables –P INPUT DROP
#iptables –P OUTPUT ACCEPT
#iptables –P FORWARD ACCEPT
#iptables –t nat –P PREROUTING ACCEPT
#iptables –t nat –P OUTPUT ACCEPT
#iptables -t nat –P POSTROUTING ACCEPT

In this example, the policy accepts outgoing packets and denies incoming packets.

Append or Delete Rules

Usage:

 # iptables [-t table] [-AI] [INPUT, OUTPUT, FORWARD] [-io interface] [-p tcp, udp, icmp, all] [-s IP/network] [--sport ports] [-d IP/network] [--dport ports] –j [ACCEPT. DROP]

-A: Append one or more rules to the end of the selected chain.
-I: Insert one or more rules in the selected chain as the given rule number.
-i: Name of an interface via which a packet is going to be received.
-o: Name of an interface via which a packet is going to be sent.
-p: The protocol of the rule or of the packet to check.
-s: Source address (network name, host name, network IP address, or plain IP address).
--sport: Source port number.
-d: Destination address.
--dport: Destination port number.
-j: Jump target. Specifies the target of the rules; i.e., how to handle matched packets.

For example, ACCEPT the packet, DROP the packet, or LOG the packet.

Examples:

Example 1: Accept all packets from the lo interface.

 # iptables –A INPUT –i lo –j ACCEPT

Example 2: Accept TCP packets from 192.168.0.1.

 # iptables –A INPUT –i eth0 –p tcp –s 192.168.0.1 –j ACCEPT

Example 3: Accept TCP packets from Class C network 192.168.1.0/24.

 # iptables –A INPUT –i eth0 –p tcp –s 192.168.1.0/24 –j ACCEPT

Example 4: Drop TCP packets from 192.168.1.25.

 # iptables –A INPUT –i eth0 –p tcp –s 192.168.1.25 –j DROP

Example 5: Drop TCP packets addressed for port 21.

 # iptables –A INPUT –i eth0 –p tcp --dport 21 –j DROP

Example 6: Accept TCP packets from 192.168.0.24 to V2416A-LX's port 137, 138, 139

 # iptables –A INPUT –i eth0 –p tcp –s 192.168.0.24 --dport 137:139 –j ACCEPT

Example 7: Log TCP packets that visit V2416A-LX's port 25.

 # iptables –A INPUT –i eth0 –p tcp --dport 25 –j LOG

Example 8: Drop all packets from MAC address 01:02:03:04:05:06.

 # iptables –A INPUT –i eth0 –p all –m mac --mac-source 01:02:03:04:05:06 –j DROP

ATTENTION

In Example 8, remember to issue the command **# modprobe ipt_mac** first to load the module ipt_mac.

NAT (Network Address Translation)

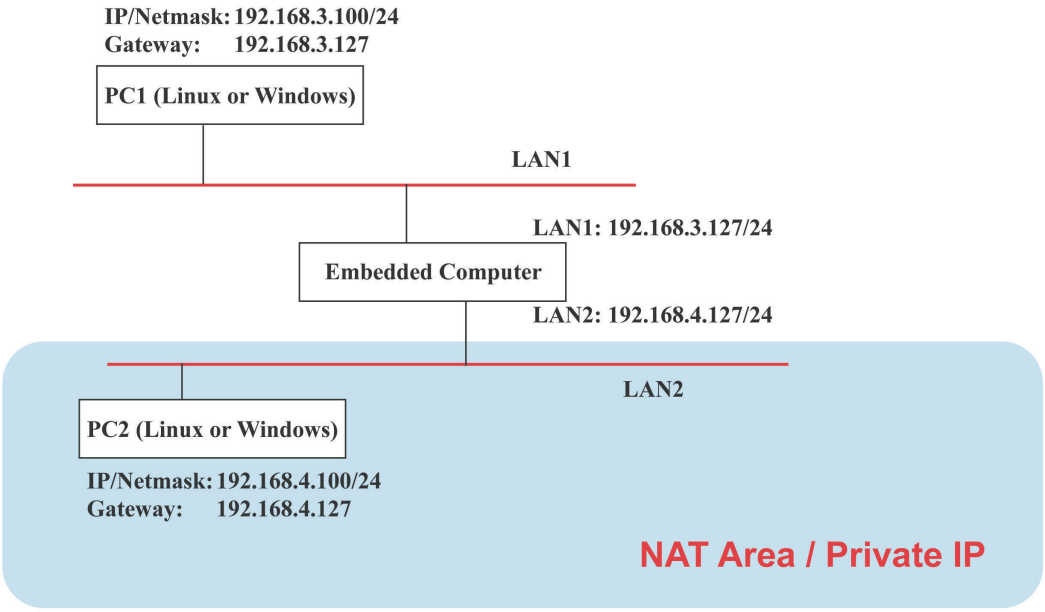
The NAT (Network Address Translation) protocol translates IP addresses used on one network into IP addresses used on a connecting network. One network is designated the inside network and the other is the outside network. Typically, the V2416A-LX connects several devices on a network and maps local inside network addresses to one or more global outside IP addresses, and un-maps the global IP addresses on incoming packets back into local IP addresses.

ATTENTION

Click on the following link for more information about NAT:
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

NAT Example

The IP address of all packets leaving LAN1 are changed to **192.168.3.127** (you will need to load the module **ipt_MASQUERADE**):



Enabling NAT at Bootup

In most real world situations, you will want to use a simple shell script to enable NAT when the V2416A-LX boots up. The following script is an example.

```
#!/bin/bash
# If you put this shell script in the /home/nat.sh
# Remember to chmod 744 /home/nat.sh
# Edit the rc.local file to make this shell startup automatically.
# vi /etc/rc.local
# Add a line in the end of rc.local /home/nat.sh
EXIF= "eth0"  #This is an external interface for setting up a valid IP address.
EXNET= "192.168.4.0/24"  #This is an internal network address.
# Step 1. Insert modules.
# Here 2> /dev/null means the standard error messages will be dump to null device.
modprobe ip_tables  2> /dev/null
modprobe ip_nat_ftp  2> /dev/null
modprobe ip_nat_irc  2> /dev/null
modprobe ip_conntrack  2> /dev/null
modprobe ip_conntrack_ftp  2> /dev/null
modprobe ip_conntrack_irc  2> /dev/null
# Step 2. Define variables, enable routing and erase default rules.
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
export PATH
echo "1" > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -F
/sbin/iptables -X
/sbin/iptables -Z
/sbin/iptables -F -t nat
/sbin/iptables -X -t nat
```

```
/sbin/iptables -Z -t nat
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT
# Step 3. Enable IP masquerade.
#ehco 1 > /proc/sys/net/ipv4/ip_forward#modprobe ipt_MASQUERADE#iptables -t nat -
A POSTROUTING -o eth0 -j MASQUERADE
```

PPP (Point to Point Protocol)

PPP (Point to Point Protocol) is used to run IP (Internet Protocol) and other network protocols over a serial link. PPP can be used for direct serial connections (using a null-modem cable) over a Telnet link, and links established using a modem over a telephone line.

Modem/PPP access is almost identical to connecting directly to a network through the V2416A-LX Ethernet port. Since PPP is a peer-to-peer system, the V2416A-LX can also use PPP to link two networks (or a local network to the Internet) to create a Wide Area Network (WAN).

ATTENTION

Click on the following links for more information about PPP:

http://tldp.org/HOWTO/PPP-HOWTO/index.html
http://axion.physics.ubc.ca/ppp-linux.html

Connecting to a PPP Server over a Simple Dial-up Connection

The following command is used to connect to a PPP server by modem. Use this command for old ppp servers that prompt for a login name (replace "username" with the correct name) and password (replace "password" with the correct password). Note that "debug crtscts" and "defaultroute 192.1.1.17" are optional.

#pppd connect 'chat -v "" ATDT5551212 CONNECT ""' ogin: username word: password' /dev/ttyMUE0 115200 debug crtscts modem defaultroute 192.1.1.17

If the PPP server does not prompt for the username and password, the command should be entered as follows (replace "username" with the correct username and replace "password" with the correct password):

#pppd connect 'chat -v "" ATDT5551212 CONNECT ""' user username password password /dev/ttyMUE0 115200 crtscts modem

The pppd options are described below:

connect 'chat etc...'	This option gives the command to contact the PPP server. The **chat** program is used to dial a remote computer. The entire command is enclosed in single quotes because pppd expects a one-word argument for the **connect** option. The options for **chat** are given below:
-v	verbose mode; log what we do to syslog
" "	Double quotes—don't wait for a prompt, but instead do ... (note that you must include a space after the second quotation mark)
ATDT5551212	Dial the modem, and then ...
CONNECT	Wait for an answer.

" " Send a return (null text followed by the usual return)

ogin: username word: password
Log in with username and password.

Note: Refer to the chat man page, chat.8, for more information about the **chat** utility.

/dev/ Specify the callout serial port.

115200 The baud rate.

debug Log status in syslog.

crtscts Use hardware flow control between the computer and modem (at baudrate of 115200 this is a must).

modem Indicates that this is a modem device; pppd will hang up the phone before and after making the call.

defaultroute Once the PPP link is established, make it the default route; if you have a PPP link to the Internet, this is probably what you want.

192.1.1.17 This is a degenerate case of a general option of the form x.x.x.x:y.y.y.y. Here x.x.x.x is the local IP address and y.y.y.y is the IP address of the remote end of the PPP connection. If this option is not specified, or if just one side is specified, then x.x.x.x defaults to the IP address associated with the local machine's hostname (located in /**etc/hosts**), and y.y.y.y is determined by the remote machine.

Connecting to a PPP Server over a Hard-wired Link

If a username and password are not required, use the following command (note that **noipdefault** is optional):

#pppd connect 'chat –v" " " " ' noipdefault /dev/ttyMUE0 19200 crtscts

If a username and password is required, use the following command (note that **noipdefault** is optional, and the username and password are both "root"):

#pppd connect 'chat –v" " " " ' user root password root noipdefault /dev/ttyMUE0 19200 crtscts

Checking the Connection

Once you have set up a PPP connection, there are some steps you can take to test the connection. First, type:

#/sbin/ifconfig

Depending on your distribution, the command might be located elsewhere. After executing the command, you should be able to see all of the network interfaces that are UP.

ppp0 should be one of the network interfaces. You should recognize the first IP address as the IP address of the computer, and **P-t-P address** is the IP address of the server. The output should be similar to the following:

```
lo      Link encap Local Loopback
        inet addr 127.0.0.1   Bcast 127.255.255.255 Mask 255.0.0.0
        UP LOOPBACK RUNNING   MTU 2000   Metric 1
        RX packets 0 errors 0 dropped 0 overrun 0

ppp0    Link encap Point-to-Point Protocol
        inet addr 192.76.32.3   P-t-P 129.67.1.165 Mask 255.255.255.0
        UP POINTOPOINT RUNNING   MTU 1500   Metric 1
        RX packets 33 errors 0 dropped 0 overrun 0
        TX packets 42 errors 0 dropped 0 overrun 0
```

Now, type:

#ping z.z.z.z

where z.z.z.z is the address of your name server. The output should be similar to the following:

```
root@MOXA:~# ping 129.67.1.165
PING 129.67.1.165 (129.67.1.165): 56 data bytes
64 bytes from 129.67.1.165: icmp_seq=0 ttl=225 time=268 ms
64 bytes from 129.67.1.165: icmp_seq=1 ttl=225 time=247 ms
64 bytes from 129.67.1.165: icmp_seq=2 ttl=225 time=266 ms
^C
--- 129.67.1.165 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 247/260/268 ms
MOXA:~#
```

Try typing:

#netstat -nr

You should see three routes similar to the following:

```
Kernel routing table
Destination   Gateway    Genmask         Flags   Metric  Ref Use
iface
129.67.1.165  0.0.0.0    255.255.255.255 UH      0       0   6
ppp0
127.0.0.0     0.0.0.0    255.0.0.0       U       0       0   0 lo
0.0.0.0       129.67.1.165  0.0.0.0     UG      0       0   6298
ppp0
```

If your output looks similar but does not have the "destination 0.0.0.0" line (which refers to the default route used for connections), you may have run pppd without the **defaultroute** option. At this point, you can try using Telnet, ftp, or finger, bearing in mind that you will have to use numeric IP addresses unless you have configured **/etc/resolv.conf** correctly.

Setting up a Machine for Incoming PPP Connections

Method 1: pppd dial-in with pppd commands

This first example applies to using a modem, and requiring authorization with a username and password.

#pppd /dev/ttyMUE0 115200 crtscts modem 192.168.16.1:192.168.16.2 login auth

You should also add the following line to the file **/etc/ppp/pap-secrets**:

*** * "" ***

The first star (*) lets everyone login. The second star (*) lets every host connect. The pair of double quotation marks ("") indicates that the file **/etc/passwd** can be used to check the password. The last star (*) is to let any IP connect.

The following example does not check the username and password:

 # pppd/dev/ttyMUE0 115200 crtscts modem 192.168.16.1:192.168.16.2

Method 2: pppd dial-in with pppd script

Configure a dial-in script **/etc/ppp/peer/dialin**

```
# You usually need this if there is no PAP authentication
noauth
#auth
#login

# The chat script (be sure to edit that file, too!)
init "/usr/sbin/chat -v -f /etc/ppp/ppp-ttyM0.chat"

# Set up routing to go through this PPP link
defaultroute

# Default modem (you better replace this with /dev/ttySx!)
/dev/ttyM0

# Speed
115200

# Keep modem up even if connection fails
persist
crtscts
modem
192.168.16.1:192.168.16.2
debug
-detach
```

Configure the chat script **/etc/ppp/ppp-ttyM0.chat**

```
SAY     'Auto Answer ON\n'
''      ATS0=1
```

Start the **pppd** dial-in service.

```
# pppd call dialin
```

ATTENTION

If you would like to have auto dial-in service, you can launch the dial-in service in **/etc/inittab** with the respawn command.

```
root@MOXA:~# mount -o remount,rw /dev/hda1 /
root@MOXA:~# echo "p0:2345:respawn:pppd call dialin" >> /etc/inittab
root@MOXA:~# umount /
```

PPPoE

Use the following procedure to configure PPPoE:

1. Connect the V2416A-LX's LAN port to an ADSL modem with a cross-over cable, HUB, or switch.
2. Log in to the V2416A-LX as the root user.
3. Edit the file **/etc/ppp/chap-secrets** and add the following:
   "username@hinet.net" * "password" *

```
# Secrets for authentication using CHAP
# client        server  secret                  IP addresses

# PPPOE example, if you want to use it, you need to unmark it and modify it
 "username@hinet.net"   *      "password"      *
```

username@hinet.net is the username obtained from the ISP to log in to the ISP account. **password** is the corresponding password for the account.

4. Edit the file **/etc/ppp/pap-secrets** and add the following:
   "username@hinet.net" * "password" *

```
# ATTENTION: The definitions here can allow users to login without a
# password if you don't use the login option of pppd! The mgetty Debian
# package already provides this option; make sure you don't change that.

# INBOUND connections

# Every regular user can use PPP and has to use passwords from /etc/passwd
*       hostname        ""      *
 "username@hinet.net"   *      "password"      *

# UserIDs that cannot use PPP at all. Check your /etc/passwd and add any
# other accounts that should not be able to use pppd!
guest   hostname        "*"     -
master  hostname        "*"     -
root    hostname        "*"     -
support hostname        "*"     -
stats   hostname        "*"     -

# OUTBOUND connections
```

username@hinet.net is the username obtained from the ISP to log in to the ISP account. **password** is the corresponding password for the account.

5. Edit the file **/etc/ppp/options** and add the following line:
   plugin rp-pppoe

```
# received.  Note: it is not advisable to use this option with the persist
# option without the demand option.  If the active-filter option is given,
# data packets which are rejected by the specified activity filter also
# count as the link being idle.
#idle <n>

# Specifies how many seconds to wait before re-initiating the link after
# it terminates.  This option only has any effect if the persist or demand
# option is used.  The holdoff period is not applied if the link was
# terminated because it was idle.
#holdoff <n>

# Wait for up n milliseconds after the connect script finishes for a valid
# PPP packet from the peer.  At the end of this time, or when a valid PPP
# packet is received from the peer, pppd will commence negotiation by
# sending its first LCP packet.  The default value is 1000 (1 second).
# This wait period only applies if the connect or pty option is used.
#connect-delay <n>

# Load the pppoe plugin
```

```
plugin rp-pppoe.so

# ---<End of File>---
```

6. If you use LAN1 to connect to the ADSL modem, add the file **/etc/ppp/options.eth0**, if you use LAN2 to connect to the ADSL modem, add **/etc/ppp/options.eth1**, etc.

```
name username@hinet.net
mtu 1492
mru 1492
defaultroute
noipdefault
~
~
 "/etc/ppp/options.eth0" 5 lines, 67 characters
```

Type your username (the one you set in the **/etc/ppp/pap-secrets and /etc/ppp/chap-secrets** files) after the **name** option. You may add other options as needed.

7. Set up DNS.
If you are using DNS servers supplied by your ISP, edit the file **/etc/resolv.conf** by adding the following lines of code:
nameserver ip_addr_of_first_dns_server
nameserver ip_addr_of_second_dns_server

For example:
nameserver 168.95.1.1
nameserver 139.175.10.20

```
root@Moxa:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
nameserver 168.95.1.1
nameserver 139.175.10.20
root@Moxa:/etc#
```

Use the following command to create a **pppoe** connection:
#pppd eth0

8. The ADSL modem is connected to the **LAN1** port, which is named **eth0**. If the ADSL modem is connected to **LAN2**, use **eth1**, etc.
9. Type **#ifconfig ppp0** to check if the connection is OK. If the connection is OK, you should see the IP address of ppp0. Use **#ping** to test the IP address.

```
ppp0    Link encap Point-to-Point Protocol
        inet addr 192.76.32.3   P-t-P 129.67.1.165 Mask 255.255.255.0
        UP POINTOPOINT RUNNING   MTU 1500   Metric 1
        RX packets 33 errors 0 dropped 0 overrun 0
        TX packets 42 errors 0 dropped 0 overrun 0
```

10. If you want to disconnect the connection, use the kill command to kill the **pppd** process.

NFS (Network File System) Client

The Network File System (NFS) is used to mount a disk partition on a remote machine (as if it were on a local hard drive), allowing fast, seamless sharing of files across a network. NFS allows users to develop applications for the V2416A-LX without worrying about the amount of disk space that will be available. The V2416A-LX only supports NFS client protocol.

ATTENTION

Click on the following links for more information about NFS.
http://www.ietf.org/rfc/rfc1213.txt
http://www.faqs.org/rfcs/rfc1317.html

The following procedures illustrate how to mount a remote NFS Server.

1. Scan the NFS Server's shared directory:
   #showmount -e HOST
   showmount: Shows the mount information of an NFS Server
   -e: Shows the NFS Server's export list.
   HOST: IP address or DNS address
2. Establish a mount point on the NFS Client site:
   #mkdir -p /home/nfs/public
3. Mount the remote directory to a local directory:
   # mount -t nfs -o nolock 192.168.3.100:/home/public /home/nfs/public
   (This is where 192.168.3.100 is the example IP address of the NFS server.)

SNMP

The V2416A-LX comes with the SNMP (Simple Network Management Protocol) agent software pre-installed. It supports **RFC 1213 MIB-II**.

By default, snmpd listens for connections from the local system. To prevent listening, remove the configuration text **agentAddress udp:127.0.0.1:161** from the system file: **/etc/snmp/snmpd.conf**.

Example:

```
root@Moxa:~# vi /etc/snmp/snmpd.conf
################################################################################
#
# EXAMPLE.conf:
#   An example configuration file for configuring the Net-SNMP agent ('snmpd')
#   See the 'snmpd.conf(5)' man page for details
#
#  Some entries are deliberately commented out, and will need to be explicitly activated
#
################################################################################
#
#  AGENT BEHAVIOUR
#

#  Listen for connections from the local system only
#  agentAddress  udp:127.0.0.1:161
#  Listen for connections on all interfaces (both IPv4 *and* Ipv6)
agentAddress udp:161,udp6:[::1]:161
……
```

After editing the config file, restart the snmpd daemon.

```
root@Moxa:~# /etc/init.d/snmpd restart
```

The following example shows an SNMP agent responding to a query from the SNMP browser on the host site:

```
root@Moxa:~# snmpwalk -v 1 -c public 192.168.3.127
iso.3.6.1.2.1.1.1.0 = STRING: "Linux Moxa 3.16.0-0.bpo.4-amd64 #1 SMP Debian
```

```
3.16.7-ckt4-3~bpo70+1 (2015-02-12) x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8691.12.2400
iso.3.6.1.2.1.1.3.0 = Timeticks: (400) 0:00:04.00
iso.3.6.1.2.1.1.4.0 = STRING: "Moxa Inc., Embedded Computing Business.
<www.moxa.com>"
iso.3.6.1.2.1.1.5.0 = STRING: "Moxa"
iso.3.6.1.2.1.1.6.0 = STRING: "Fl.4, No.135, Lane 235, Baoquao Rd., Xindian Dist.,
New Taipei City, Taiwan, R.O.C.\""
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for the
SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP
implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.25.1.1.0 = Timeticks: (737940) 2:02:59.40
iso.3.6.1.2.1.25.1.2.0 = Hex-STRING: 07 DF 08 05 0F 14 01 00 2B 08 00
iso.3.6.1.2.1.25.1.3.0 = INTEGER: 1536
iso.3.6.1.2.1.25.1.4.0 = STRING: "BOOT_IMAGE=/boot/vmlinuz-3.16.0-0.bpo.4-amd64
root=LABEL=V2400A_MOXA ro quiet
"
iso.3.6.1.2.1.25.1.5.0 = Gauge32: 2
iso.3.6.1.2.1.25.1.6.0 = Gauge32: 93
iso.3.6.1.2.1.25.1.7.0 = INTEGER: 0
End of MIB
```

ATTENTION

Click on the following links for more information about RFC1317 RS-232 like groups and RFC 1213 MIB-II:
http://www.faqs.org/rfcs/rfc1317.html
http://www.ietf.org/rfc/rfc1213.txt

OpenVPN

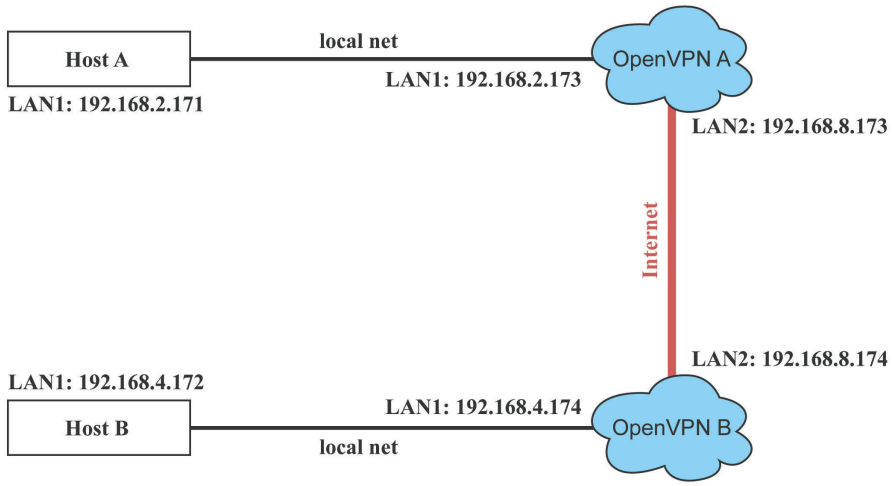
OpenVPN provides two types of tunnels for users to implement VPNS: **Routed IP Tunnels** and **Bridged Ethernet Tunnels**.

An Ethernet bridge is used to connect different Ethernet networks together. The Ethernets are bundled into one bigger, "logical" Ethernet. Each Ethernet corresponds to one physical interface (or port) that is connected to the bridge.

On each OpenVPN machine, you should carry out configurations in the **/etc/openvpn** directory, where script files and key files reside. Once established, all operations will be performed in that directory.

Ethernet Bridging for Private Networks on Different Subnets

1. Set up four machines, as shown in the following diagram.



Host A represents the machine that belongs to OpenVPN A, and Host B represents the machine that belongs to OpenVPN B. The two remote subnets are configured for a different range of IP addresses. When this configuration is moved to a public network, the external interfaces of the OpenVPN machines should be configured for static IPs, or connected to another device (such as a firewall or DSL box) first.

2. Generate a preset shared key by typing the following command:
   `# openvpn --genkey --secret secrouter.key`

3. Copy the file that is generated to the OpenVPN machine:
   `# scp /etc/openvpn/secrouter.key 192.168.8.174:/etc/openvpn`

ATTENTION

A preshared key is located at **/etc/openvpn/secrouter.key**. You can use it for testing purposes. We suggest creating a new key for non-testing purposes.

4. On machine OpenVPN A, modify the remote address in configuration file **/etc/openvpn/tap0-br.conf**.

```
# point to the peer
remote 192.168.8.174
dev tap0
port 1194
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
```

```
ping 40
up /etc/openvpn/tap0-br.sh
#comp-lzo
```

5. Next, modify the routing table in **/etc/openvpn/tap0-br.sh script**.

```
#--------------------------Start---------------------------
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.4.0 netmask 255.255.255.0 dev br0
#----------------------------end---------------------------
```

And then configure the bridge interface in **/etc/openvpn/bridge**.

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"
eth_ip="192.168.8.173"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.174"
...
```

Start the bridge script file to configure the bridge interface:
 # /etc/openvpn/bridge restart

On machine OpenVPN B, modify the remote address in configuration file **/etc/openvpn/tap0-br.conf**.

```
# point to the peer
remote 192.168.8.173
dev tap0
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
up /etc/openvpn/tap0-br.sh
#comp-lzo
```

6. Next modify the routing table in **/etc/openvpn/tap0-br.sh script** file.

```
#----------------------------------Start-----------------------------
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 dev br0
#---------------------------------- end ---------------------------
```

And then configure the bridge interface in **/etc/openvpn/bridge**.

```
#!/bin/bash
# Create global variables
# Define Bridge Interface
br="br0"
# Define list of TAP interfaces to be bridged,
# for example tap="tap0 tap1 tap2".
tap="tap0"
# Define physical ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"
eth_ip="192.168.8.174"
eth_netmask="255.255.255.0"
eth_broadcast="192.168.8.255"
#gw="192.168.8.173"
...
```

Start the bridge script file to configure the bridge interface.
 # /etc/openvpn/bridge restart

ATTENTION

Select cipher and authentication algorithms by specifying cipher and auth. To see which algorithms are available, type:
 # openvpn --show-ciphers
 # openvpn --show-auths

7. Start both OpenVPN peers on machine OpenVPN A and OpenVPN B.
   # openvpn --config /etc/openvpn/tap0-br.conf&

   If you see the line **Peer Connection Initiated with 192.168.8.173:5000**on each machine, the connection between OpenVPN machines has been established successfully on UDP port 5000.

ATTENTION

You can create link symbols to start the OpenVPN service at boot time:
 # ln -sf /etc/init.d/openvpn /etc/rc2.d/S16openvpn

To stop the service, you should create these links:
 # ln -sf /etc/init.d/openvpn /etc/rc0.d/K80openvpn
 # ln -sf /etc/init.d/openvpn /etc/rc6.d/K80openvpn

8. On each OpenVPN machine, check the routing table by typing the command **# route**

```
Destination     Gateway Genmsk          Flags   Metric  Ref Use Iface
192.168.5.0     0.0.0.0 255.255.255.0   U       0       0   0   eth2
192.168.4.0     0.0.0.0 255.255.255.0   U       0       0   0   br0
192.168.3.0     0.0.0.0 255.255.255.0   U       0       0   0   eth0
192.168.30.0    0.0.0.0 255.255.255.0   U       0       0   0   eth3
192.168.8.0     0.0.0.0 255.255.255.0   U       0       0   0   br0
```

Interface **eth1** and device **tap0** both connect to the bridging interface, and the virtual device **tun** sits on top of **tap0**. This ensures that all traffic coming to this bridge from internal networks connected to interface eth1 write to the TAP/TUN device that the OpenVPN program monitors. Once the OpenVPN program detects traffic on the virtual device, it sends the traffic to its peer.

9. To create an indirect connection to Host B from Host A, you need to add the following routing item:
   # route add –net 192.168.4.0 netmask 255.255.255.0 dev eth0

   To create an indirect connection to Host A from Host B, you need to add the following routing item:
   # route add –net 192.168.2.0 netmask 255.255.255.0 dev eth0

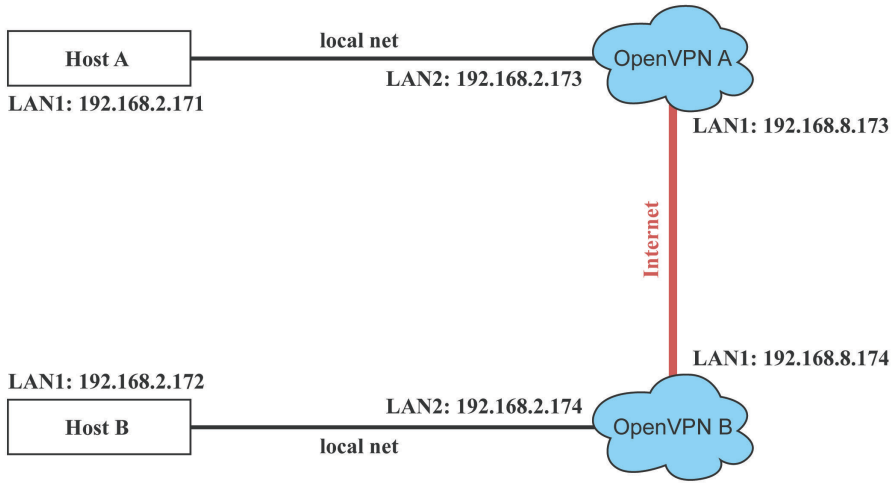
   Now ping Host B from Host A by typing:
   # ping 192.168.4.174

   A successful ping indicates that you have created a VPN system that only allows authorized users from one internal network to access users at the remote site. For this system, all data is transmitted by UDP packets on port 5000 between OpenVPN peers.

10. To shut down OpenVPN programs, type the command:
    # killall -TERM openvpn

Ethernet Bridging for Private Networks on the Same Subnet

1. Set up four machines, as shown in the following diagram.

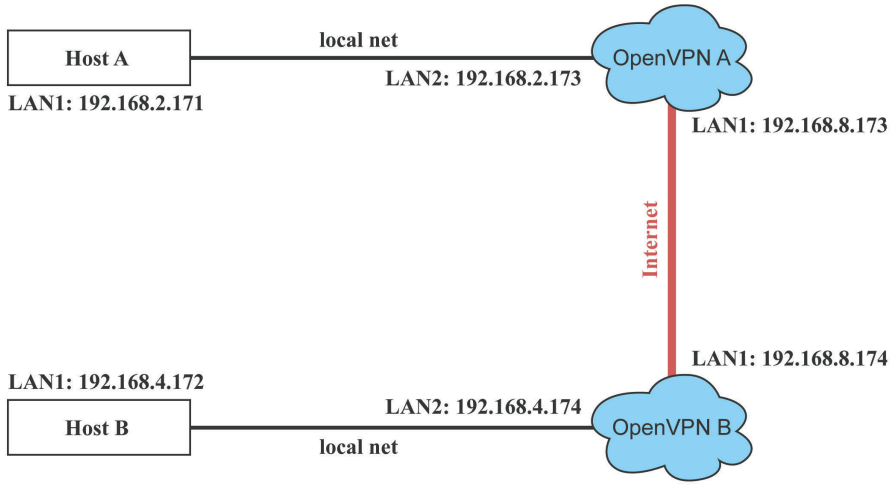


2. The configuration procedure is almost the same as for the previous example. The only difference is that you will need to comment out the parameter **up** in **/etc/openvpn/tap0-br.conf** of OpenVPN A and **/etc/openvpn/tap0-br.conf** of OpenVPN B.

```
# point to the peer
remote 192.168.8.174
dev tap0
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
#up /etc/openvpn/tap0-br.sh
#comp-lzo
```

Routed IP

1. Set up four machines, as shown in the following diagram.



2. On machine OpenVPN A, modify the remote address in configuration file **/etc/openvpn/tun.conf**.

```
# point to the peer
remote 192.168.8.174
dev tun
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
ifconfig 192.168.2.173 192.168.4.174
up /etc/openvpn/tun.sh
-----
```

3. Next, modify the routing table in script file **/etc/openvpn/tun.sh**.

```
#---------------------------Start----------------------------
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 gw $5
#----------------------------end-----------------------------
```

4. On machine OpenVPN B, modify the remote address in configuration file **/etc/openvpn/tun.conf**.

```
# point to the peer
remote 192.168.8.173
dev tun
secret /etc/openvpn/secrouter.key
cipher DES-EDE3-CBC
auth MD5
tun-mtu 1500
tun-mtu-extra 64
ping 40
ifconfig 192.168.4.174 192.168.2.173
up /etc/openvpn/tun.sh
```

And then modify the routing table in script file **/etc/openvpn/tun.sh**.

```
#---------------------------Start-----------------------------
#!/bin/sh
# value after "-net" is the subnet behind the remote peer
route add -net 192.168.2.0 netmask 255.255.255.0 gw $5
#---------------------------end-----------------------------
```

The first argument of parameter **ifconfig** is the local internal interface and the second argument is the internal interface at the remote peer.

$5 is the argument that the OpenVPN program passes to the script file. Its value is the second argument of **ifconfig** in the configuration file.

5. Check the routing table after you run the OpenVPN programs, by typing the command **# route**.

```
Destination     Gateway         Genmsk          Flags   Metric  Ref Use Iface
192.168.4.174   *               255.255.255.255 UH      0       0   0   tun0
192.168.4.0     192.168.4.174   255.255.255.0   UG      0       0   0   tun0
192.168.2.0     *               255.255.255.0   U       0       0   0   eth1
192.168.8.0     *               255.255.255.0   U       0       0   0   eth0
```

4

Programming Guide

The following topics are covered in this chapter:

- **Device API**
- **Getting the Product Serial Number**
- **RTC (Real Time Clock)**
- **Digital I/O**
  - Special Note
  - Examples
- **WDT (Watch Dog Timer)**
  - Introduction
  - Watchdog Usage
  - How the WDT Works
  - Watchdog Device IOCTL Commands
  - Examples

Device API

The V2416A-LX supports control devices with the **ioctl** system API. The interface is shown below:

```
int ioctl(int d, int request,…);
Input:
    <d> open device node return file handle
    <request> argument in or out
```

Refer to desktop Linux's man page for detailed documentation:

#man ioctl

Getting the Product Serial Number

Use dmidecode to ready the product information. The command is:

```
moxa@Moxa:~$ sudo dmidecode -s "baseboard-manufacturer"
MOXA
```

Refer to the following keywords to get other product information.

```
  bios-vendor
  bios-version
  bios-release-date
  system-manufacturer
  system-product-name
  system-version
  system-serial-number
  system-uuid
  baseboard-manufacturer
  baseboard-product-name
  baseboard-version
  baseboard-serial-number
  baseboard-asset-tag
  chassis-manufacturer
  chassis-type
  chassis-version
  chassis-serial-number
  chassis-asset-tag
  processor-family
  processor-manufacturer
  processor-version
  processor-frequency
```

RTC (Real Time Clock)

The device node is located at **/dev/rtc**. The V2416A-LX supports standard Linux simple RTC control. You must include **<linux/rtc.h>**.

1. Function: RTC_RD_TIME

```
int ioctl(fd, RTC_RD_TIME, struct rtc_time *time);
Description: read time information from the RTC. It will return the value on argument
3.
```

2. Function: RTC_SET_TIME

```
int ioctl(fd, RTC_SET_TIME, struct rtc_time *time);
Description: set RTC time. Argument 3 will be passed to RTC.
```

Digital I/O

Digital Output channels can be set to high or low. The channels are controlled by the function call **set_dout_state( )**. Use the digital input channels to detect the state change of the digital input signal. The DI channels can also be used to detect whether or not the state of a digital signal changes during a fixed period of time. This can be done with the function call **set_din_event( )**.

Return error code definitions:

```
#define DIO_ERROR_PORT -1 // no such port
#define DIO_ERROR_MODE -2 // no such mode or state
#define DIO_ERROR_CONTROL -3 // open or ioctl fail
#define DIO_ERROR_DURATION -4 // The value of duration is not 0 or not in the range,
40 <= duration <= 3600000 milliseconds (1 hour)
#define DIO_ERROR_DURATION_20MS -5 // The value of duration must be a multiple of 20
ms
#define DIO_OK 0
```

DIN and DOUT definitions:

```
#define DIO_HIGH 1
#define DIO_LOW 0
```

Moxa functions for DI/DO

Function	**int set_dout_state(int doport, int state)**
Description	Set the DOUT port to high or low state.
Input	<doport> The DOUT port you want to set. Port starts from 0 to 1 <state> Set high or low state; DIO_HIGH (1) for high, DIO_LOW (0) for low.
Output	None
Return	refer to the error code

Function	**int get_din_state(int diport, int *state)**
Description	Get the DIN port state
Input	<diport> The DIN port to get the state of. Port numbering is from 0 to 5 <state> Save the current state
Output	<state> DIO_HIGH (1) for high, DIO_LOW (0) for low
Return	Refer to the error code

Function	**int get_dout_state(int doport, int *state)**
Description	Get the DOUT port state
Input	<doport> The DOUT port to get the state of. <state> Save the current state.
Output	<state> DIO_HIGH (1) for high, DIO_LOW (0) for low
Return	Refer to the error code

Function	**int set_din_event(int diport, void (*func)(int diport), int mode, long int duration)**
Description	Set the DIN event when the state is changed from high to low or from low to high
Input	<diport> The port that will be used to detect the DIN event. Port numbering is from 0 to 5. This value depends on your device. <(*func) (int diport)> Not NULL: Returns the call back function. When the event occurs, the call back function will be invoked. NULL: Clear this event <mode> DIN_EVENT_HIGH_TO_LOW (1): From high to low DIN_EVENT_LOW_TO_HIGH (0): From low to high DIN_EVENT_CLEAR (-1): Clear this event <duration> 0: Detect the din event DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH without duration Not 0: Detect the din event DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH with duration. Note: The value of "duration" must be a multiple of 20 milliseconds. The range of "duration" is 0, or 40 <= duration <= 3600000 milliseconds. The error of the measurement is 24 ms. For example, if the DIN duration is 200 ms, this event will be generated when the DIN pin stays in the same state for a time between 176 ms and 200 ms.
Output	None
Return	Refer to the error code

Function	**int get_din_event(int diport, int *mode, long int *duration)**
Description	To retrieve the DIN event configuration, including mode (DIN_EVENT_HIGH_TO_LOW or DIN_EVENT_LOW_TO_HIGH), and the value of "duration."
Input	<diport> Which DIN port you want to retrieve <mode> Save the set event. <duration> The duration the DIN port is kept in high or low state. - return to the current duration value of diport
Output	<mode> DIN_EVENT_HIGH_TO_LOW (1): From high to low DIN_EVENT_LOW_TO_HIGH(0): From low to high DIN_EVENT_CLEAR(-1): Clear this event <duration> The value of duration should be 0 or 40 <= duration <= 3600000 milliseconds.
Return	Refer to the error code

Special Note

1. You need to build the moxalib in advance for DI/DO. The moxalib is included in the folder **\example\V2400A_LX_V1.0_example** on the CD.
2. Make sure to link the library **libmoxalib** for DI/DO programming, and include the header file **moxadevice.h**. Only one program at a time can use the DI/DO library.
3. Due to hardware limitations, you need to modify MIN_DURATION as 60 for V2416A-LX computers.

Examples

Example files **tdio.c** and **Makefile** are located in the folder **\example\V2400A_LX_V1.0_example\dio** on the CD.

WDT (Watch Dog Timer)

Introduction

The WDT works like a watchdog function, and can be enabled or disabled. When the WDT function is enabled and the application does not acknowledge it, the system will reboot.

Watchdog Usage

Users can set the ack time from a minimum of 1 sec to a maximum of 1 day.The default timer is 60 seconds and NO WAY OUT is enabled by default; there is no way of disabling the watchdog once it has been started, so if the watchdog daemon crashes, the system will reboot after the timeout has passed. If the NO WAY OUT is disabled, the user can stop the timer.

Example of setting the default timer

Edit the /etc/modprobe.d/watchdog.conf file to set the default timer. The following commands set the default timer to 60 seconds:

```
moxa@Moxa:~$ vi /etc/modprobe.d/watchdog.conf
options moxa_v2100_wdt timer_margin=60
moxa@Moxa:~$
```

Enable or disable NO WAY OUT

Edit the /etc/modprobe.d/watchdog.conf file to enable or disable NO WAY OUT.

Enable NO WAY OUT:

```
moxa@Moxa:~$ vi /etc/modprobe.d/watchdog.conf
options moxa_v2100_wdt nowayout=1
moxa@Moxa:~$
```

Disable NO WAY OUT:

```
moxa@Moxa:~$ vi /etc/modprobe.d/watchdog.conf
options moxa_v2100_wdt nowayout=0
moxa@Moxa:~$
```

Magic close

If NO WAY OUT is disabled, you can stop the timer using magic close. Use the following commands to do this:

```
root@Moxa:~$ echo V > /dev/watchdog
root@Moxa:~$
```

How the WDT Works

The Debian project supports a watchdog daemon. The watchdog daemon checks if your system is still working. If programs are no longer executing, it will perform a hard reset of the system. The standard watchdog driver and package have been installed in the V2416A. If you need to run the watchdog once the system boots up, you can use **insserv** to enable the watchdog function.

```
moxa@Moxa:~$ sudo insserv -v -d watchdog
[sudo] password for moxa:
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc0.d/K01watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc1.d/K01watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc2.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc3.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc4.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc5.d/S23watchdog
insserv: enable service ../init.d/watchdog -> /etc/init.d/../rc6.d/K01watchdog
insserv: creating .depend.boot
insserv: creating .depend.start
insserv: creating .depend.stop
moxa@Moxa:~$
```

Check the run level:

```
moxa@Moxa:~$ ls -l /etc/rc?.d/*watchdog*
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc0.d/K01watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc1.d/K01watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc2.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc3.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc4.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc5.d/S23watchdog -> ../init.d/watchdog
lrwxrwxrwx 1 root root 18 Nov  8 15:48 /etc/rc6.d/K01watchdog -> ../init.d/watchdog
moxa@Moxa:~$
```

The watchdog configuration file is located in **/etc/watchdog.conf**. Currently, we configure the watchdog daemon to acknowledge the watchdog device in 30 seconds. The realtime is to lock itself into memory, so it is never swapped out to prevent the delay of watchdog acknowledge. You can configure this file to enable the watchdog as needed by your application.

```
…
watchdog-device = /dev/watchdog
…
interval             = 30
realtime             = yes
priority             = -10
…
```

Use the following command to remove it from run-level:

```
moxa@Moxa:~# sudo insserv -r watchdog
```

Check the run level removement.

```
moxa@Moxa:~# ls -l /etc/rc?.d/*watchdog*
ls: cannot access /etc/rc?.d/*watchdog*: No such file or directory
moxa@Moxa:~#
```

Watchdog Device IOCTL Commands

IOCTL	WDIOC_GETSUPPORT
Description	Returns the support of the card itself
Input	None
Output	(struct watchdog_info *) arg
Return	On success, returns 0. Otherwise, returns a value < 0.

IOCTL	WDIOC_GETSTATUS
Description	Returns the status of the card
Input	None
Output	(int *)arg
Return	On success, returns 0. Otherwise, returns a value < 0.

IOCTL	WDIOC_GETBOOTSTATUS
Description	Returns the status of the card that was reported at bootup.
Input	None
Output	(int *)arg)
Return	On success, returns 0. Otherwise, returns a value < 0.

IOCTL	WDIOC_SETOPTIONS
Description	Lets you set the options of the card. You can either enable or disable the card.
Input	None
Output	(int *)arg)
Return	On success, returns 0. Otherwise, returns a value < 0.

IOCTL	WDIOC_KEEPALIVE
Description	Pings the card to tell it not to reset your computer.
Input	None
Output	None
Return	On success, returns 0. Otherwise, returns a value < 0.

IOCTL	WDIOC_SETTIMEOUT
Description	Sets the watchdog timeout
Input	arg: 1 to 255 seconds
Output	None
Return	On success, returns 0. Otherwise, returns a value < 0.

IOCTL	WDIOC_GETTIMEOUT
Description	Gets the current watchdog timeout.
Input	None
Output	arg: 1 to 255 seconds
Return	On success, returns 0. Otherwise, returns a value < 0.

Examples

The example file **watchdog-simple.c** acks the watchdog every 10 seconds.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(void)
{
	int fd = open("/dev/watchdog", O_WRONLY);
	int ret = 0;
	if (fd == -1) {
		perror("watchdog");
		exit(EXIT_FAILURE);
	}
	while (1) {
		ret = write(fd, "\0", 1);
		if (ret != 1) {
			ret = -1;
			break;
		}
		sleep(10);
	}
	close(fd);
	return ret;
}
```

5

Managing the Disk

The V2416A embedded computer is a RAID management platform designed to serve as a front-end for data acquisition and industrial control applications. In this chapter, we describe how to configure the volume supported by the Linux operating system.

The following topics are covered in this chapter:

- **Hot-Swapping Function**
  - Installing the hotswap driver and daemon
  - File Overview
  - Configuring the Hot-Swapping Daemon
  - Handling an Event with the Hot-Swapping Daemon
  - Logging the Hot-Swapping Daemon Message
- **Software RAID—mdadm**
  - Create Software RAID Volume
  - Check Software RAID Status
  - Replacing a Failed Disk
  - Removing RAID

Hot-Swapping Function

The V2416A-LX computers come with two removable, hot-swappable slots for inserting additional storage media such as hard disks or SSD drives. It also supports hot swapping for convenient, fast, and easy storage expansion, and provides user-defined programmable LEDs and the related API for storage management. Storage plug/unplug functionality, automatic storage removal, and storage status display are all supported.

Installing the hotswap driver and daemon

Use the following command to install the hotswap driver and daemon:

```
root@Moxa:/# dpkg -i v2400a-hotswap_1.0.0_amd64.deb
```

When you restart your computer, and the hotswap driver and daemon will execute automatically.

File Overview

The main files are listed below:

- **mxhtspd:** a daemon for monitoring hot-swap events
- **mxhtspd-setled:** a command to set up LED signals
- **/etc/mxhtspd/scripts:** scripts executed when an event occurs; the following files are included:
  action-btn-pressed
  action-disk-plugged
  action-disk-unplugged
- **libmxhtsp.so:** library

Configuring the Hot-Swapping Daemon

An **mxhtspd** daemon is provided for the V2416A hot-swapping function. It is launched with the **/etc/init.d/mxhtspd.sh** script at startup and will detect the disk status in the background.

You can configure **mxhtspd** with the following options:

- **-l facility_num:** log daemon's message by rsyslogd with LOCAL[facility_num]
- **-v:** run in verbose mode
- **-h:** print usage

The following example shows how to use the –v option to modify **/etc/init.d/mxhtspd.sh**:

```
…
start)
       echo "Starting mxhtspd daemon..."
       sleep 1
       mxhtspd -v &
```

Handling an Event with the Hot-Swapping Daemon

mxhtspd will be triggered when the following events occur:

1. **A disk is plugged in**
   When an ext2/ext3/ext4/FAT32 disk n with m partitions is been plugged in, the system will automatically mount its partitions on **/media/diskpm**, where m ranges from 1 to m. For example, if disk 1 has two partitions, they would be mounted on **/media/disk1p1** and **/media/disk1p2**.

   The action-disk-plugged script will be triggered with the disk number as argument. By default it will scan the disk on the SATA bus. You can add some operations here when necessary.

Execute the mount command to check the disk mounting situation. The following example shows the status of 2 disks with 2 partitions separately.

```
root@Moxa:/# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=236357,mode=755)
devpts on /dev/pts type devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=190648k,mode=755)
/dev/disk/by-label/V2400A_MOXA on / type ext4
(rw,noatime,errors=remount-ro,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=381280k)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
/dev/sdb1 on /media/disk1p1 type ext4 (rw,relatime,data=ordered)
/dev/sdc1 on /media/disk2p1 type ext4 (rw,relatime,data=ordered)
```

2. **A button is pressed**
   When button n is pressed, the **action-btn-pressed** script will be executed with argument n. It will unmount all partitions on disk n and the LED will blink 3 times at 1 second intervals. Finally, the LED will turn off to indicate that the disk has been successfully unmounted, and that users can remove that hard disk from the storage tray. If the disk is busy, it will show a warning message and the LED will remain green after blinking 3 times at 1 second intervals

3. **A disk is unplugged**
   When disk n is unplugged, the **action-disk-unplugged** script will be triggered with argument n. It will check if all partitions on disk n have been unmounted before they were unplugged and warn the user if they weren't. The correct procedure is to first press the button to unmount the partitions and then unplug the disk. The only purpose of this script is to warn of misuse or incorrect operation.

ATTENTION

Make sure a disk is unmounted before unplugging it. After pressing the button for the disk, the LED will blink 3 times at 1 second intervals, and then will turn off to indicate that the disk has been successfully unmounted.

Logging the Hot-Swapping Daemon Message

Take the following steps to log the **mxhtspd** message with the **rsyslogd** daemon:

1. Check your run level in the **/etc/inittab** file; the default is 2:

```
# The default runlevel.
id:2:initdefault:
```

2. Enable rsyslogd at startup:

```
root@Moxa:/# insserv -d rsyslog
```

3. Edit **/etc/init.d/mxhtspd.sh** to add **-l 0** options to enable the mxhtspd logging function with local 0.

```
#Add parameter if necessary
mxhtspd -l 0&
```

4. Edit the configuration file **/etc/rsysload.conf** and add the following setting.

```
#Uncomment below lines for mxhtspd with local 0
local0.*                    -/var/log/mxhtspd.log
```

mxhtspd will use the local0 facility to log a message. The destination file is /var/log/mxhtspd.log. The minus '-' sign indicates to omit syncing the file after every logging.

5. Restart your computer to activate the settings.

ATTENTION

When you run the **rsyslogd** daemon to log messages at startup, take care to prevent excessive disk usage.

Software RAID—mdadm

mdadm is a Linux utility used to manage software RAID devices. The name is derived from the "md" (multiple device) device nodes it "administers or manages" (it replaces the utility **mdctl**). The original name was "Mirror Disk" but was changed as the functionality increased. The RAID volume is built in Linux, not in the BIOS.

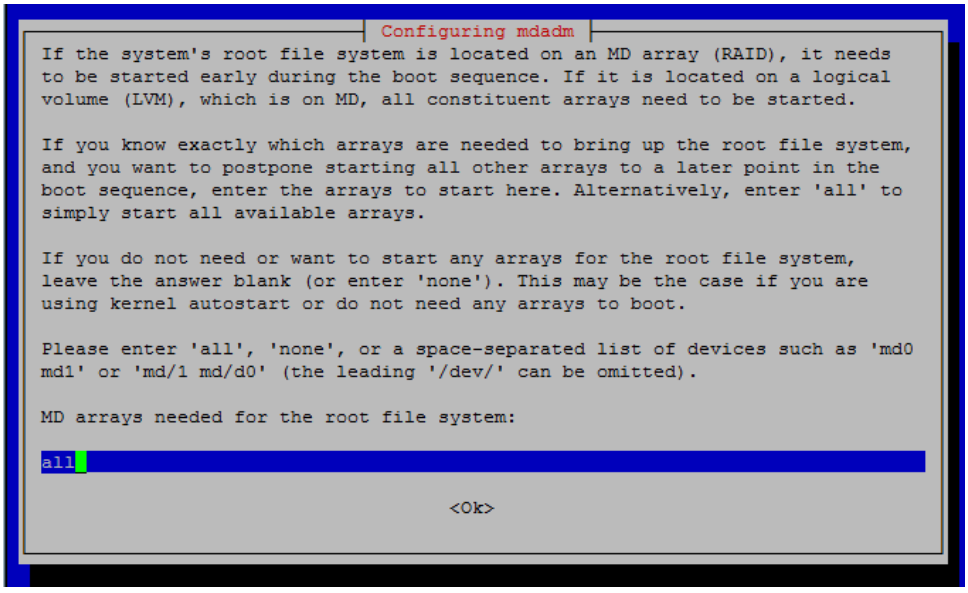
Create Software RAID Volume

The V2416A comes with 2 SATA disk slots; it can manage linear, RAID0, and RAID1 volumes on these two SATA disks.

1. Type the following command to install the **mdadm** utility.

```
root@Moxa:~# apt-get install mdadm
```

Press enter to continue.



2. Use the **/proc/mdstat** command to check the RAID device information.

```
root@Moxa:~# cat /proc/mdstat
Personalities :
Unused devices: <none>
root@Moxa:~#
```

3. Use the command **mxhtspd** to unmont the disks if they are mounted, stop the mxhtspd service, and remove the run level.

```
root@Moxa:~# umount /media/disk1p1
root@Moxa:~# umount /media/disk2p1
root@Moxa:~# /etc/init.d/mxhtspd.sh stop
```

Because the **mxhtspd** hotswap daemon only supports general disk hotswap management, the RAID management features is not included. We suggest disabling it with RAID management features. You can use this command to remove the mxhtspd service from run-level:

```
moxa@Moxa:~# insserv -r mxhtspd.sh
```

4. Use the following commands to create partitions on the disk. This action will auto mount the disk.

```
root@Moxa:~# fdisk /dev/sdb1
root@Moxa:~# fdisk /dev/sdc1
```

5. Because of the auto mount feature, you will need to unmount the disks again.

```
root@Moxa:~# umount /media/disk1p1
root@Moxa:~# umount /media/disk2p1
```

6. Create the RAID volume.
   The mdadm options shown below are used for the RAID volume creation. You can choose to create a linear mode, striping mode, or mirror mode RAID volume.

```
-C: create
-v: verbose
-l: RAID level, options are: linear, raid0, 0, stripe, raid1, 1, mirror, raid4,
4, raid5, 5, raid6, 6, raid10, 10, multipath, mp, faulty. Obviously some of these
are synonymous.
-n: the number of disks
```

7. You may create different types of software RAID:
   A. Create a linear mode software RAID.

```
root@Moxa:~# mdadm -Cv -llinear -n2 /dev/md0 /dev/sd{b,c}1
```

   B. Create a striping mode software RAID 0.

```
root@Moxa:~# mdadm -Cv -l0 -n2 /dev/md0 /dev/sd{b,c}1
```

   C. Create a mirror mode software RAID 1.

```
root@Moxa:~# mdadm -Cv -l1 -n2 /dev/md0 /dev/sd{b,c}1
```

   Next, use /proc/mdstat to check the RAID device information.

```
root@Moxa:~# cat /proc/mdstat
Personalities: [raid0]
md0: active raid0 sdb1[1] sdc1[0]
     195369504 blocks super 1.2 OK rounding

Unused devices: <none>
```

8. Format the RAID.

```
root@Moxa:~# mkfs.ext4 /dev/md0
```

9. Mount the RAID device manually.

```
root@Moxa:~# mount /dev/md0 /mnt/raid
```

10. Start the RAID volume automatically at the next boot up.
    If you need to start the array automatically, edit **/etc/mdadm/mdadm.conf**.

```
root@Moxa:~# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

    Edit **/etc/mdadm/mdadm.conf**.

```
DEVICE /dev/sdb1 /dev/sdc1
CREATE owner=root group=disk mode=0660 auto=yes
HOMEHOST <system>
```

```
MAILADDR your_email@xxx.com
ARRAY /dev/md0 metadata=1.2 name=Moxa:0 UUID=45ae9dbe:f30741ec:b22eff98:2dadb12d
```

Add the following line to /etc/fstab to mount the RAID volume.

```
/dev/md0    /mnt/raid   ext4    defaults    0   2
```

Unmount the root file system and reboot. The array should be started and mounted at /mnt/raid.

```
root@Moxa:~# umount /
```

Check Software RAID Status

Use **reading /proc/mdstat** to check the software RAID status. If the array is running, the status will be as indicated below.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear]

md0 : active linear sdb1[1] sdc1[0]
    23436724 blocks super 1.2 0k rounding

unused devices: <none>
```

If the array is not running, the status will be.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [linear]
unused devices: <none>
```

Replacing a Failed Disk

If the array is running in mirror mode and one of the disks fails, you should remove the failed disk and replace it with a new one. In this case, **sdb1[0](F)** means the sdb disk has failed.

```
md1 : active raid1 sdb1[1] sdc1[0](F)
17920384 blocks [2/1] [_U]
```

You can use the following commands to simulate the failure of one disk.

```
root@Moxa:~# mdadm --manage /dev/md0 --fail /dev/sdb1
mdadm: set /dev/sdb1 faulty in /dev/md0
root@Moxa:~# sync
```

Use the following command to remove a failed disk from RAID.

```
root@Moxa:~# mdadm -r /dev/md0 /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
```

Use the following command to replace the first drive with a new disk and add it into the array.

```
root@Moxa:~# mdadm -a /dev/md0 /dev/sdb1
```

If you check mdstat, you will find that the array is in the process of auto recovering.

```
root@Moxa:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdb1[0] sdc1[1]
     7806522 blocks super 1.2 [2/1] [_U]
     [==>..................]  recovery = 10.6% (831488/7806522) finish=0.9min
speed=118784K/sec

unused devices: <none>
```

Removing RAID

Unmont the disks if they are already mounted.

```
root@Moxa:~# umount /mnt/raid
```

Use the following command to remove a failed disk from RAID.

```
root@Moxa:~# mdadm --stop /dev/md0
```

Use the following command to remove mdadm service from run-level.

```
root@Moxa:~# insserv -r mdadm
```

6

System Recovery

The V2416A are installed with the Embedded Linux operating system, which is located in the Flash DOM (CompactFlash card) shipped with the V2416A-LX computer. In this chapter, we describe how to recover the Linux operating system if the operating system files and/or the disk file system have been damaged, which occurs only rarely.

The following topics are covered in this chapter:

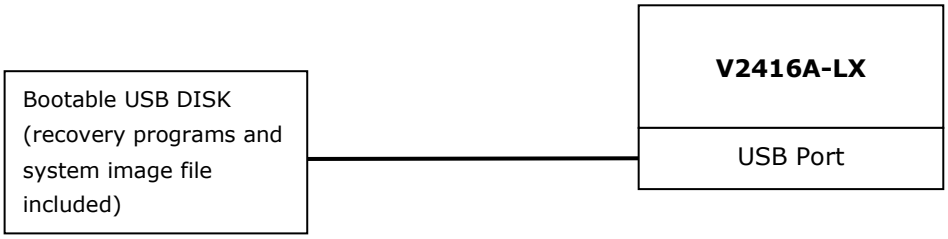
- **Recovery Environment**
- **Recovery Procedure**
- **Saving the System to the USB Drive**

Recovery Environment

The recovery environment includes the V2416A embedded computer and a bootable USB disk with the recovery programs and system image file.

Hardware

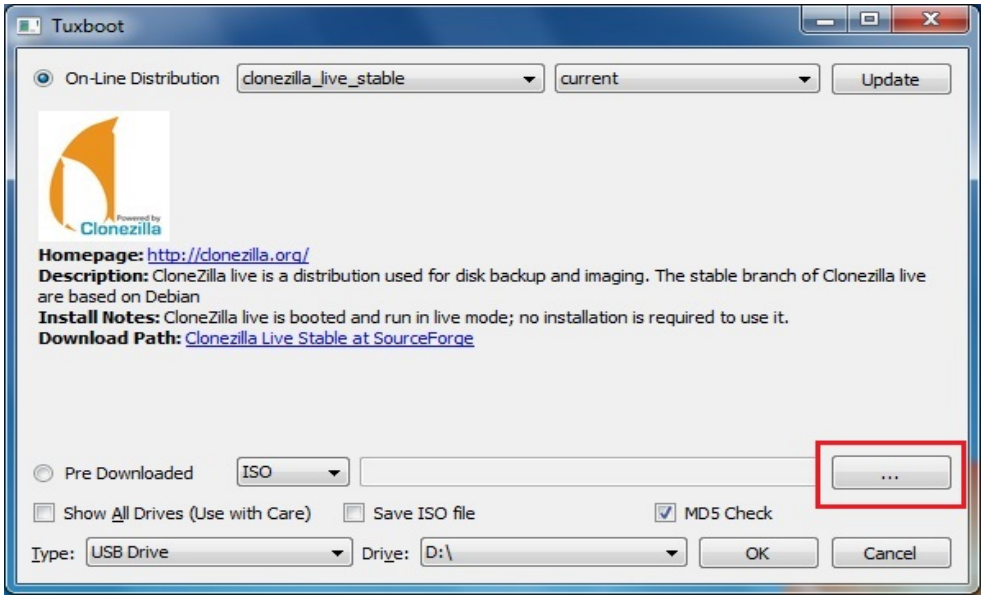
The hardware used includes a PC, a V2416A computer and a USB disk with the recovery programs. **(Note: The USB disk should be at least 2 GB.).**



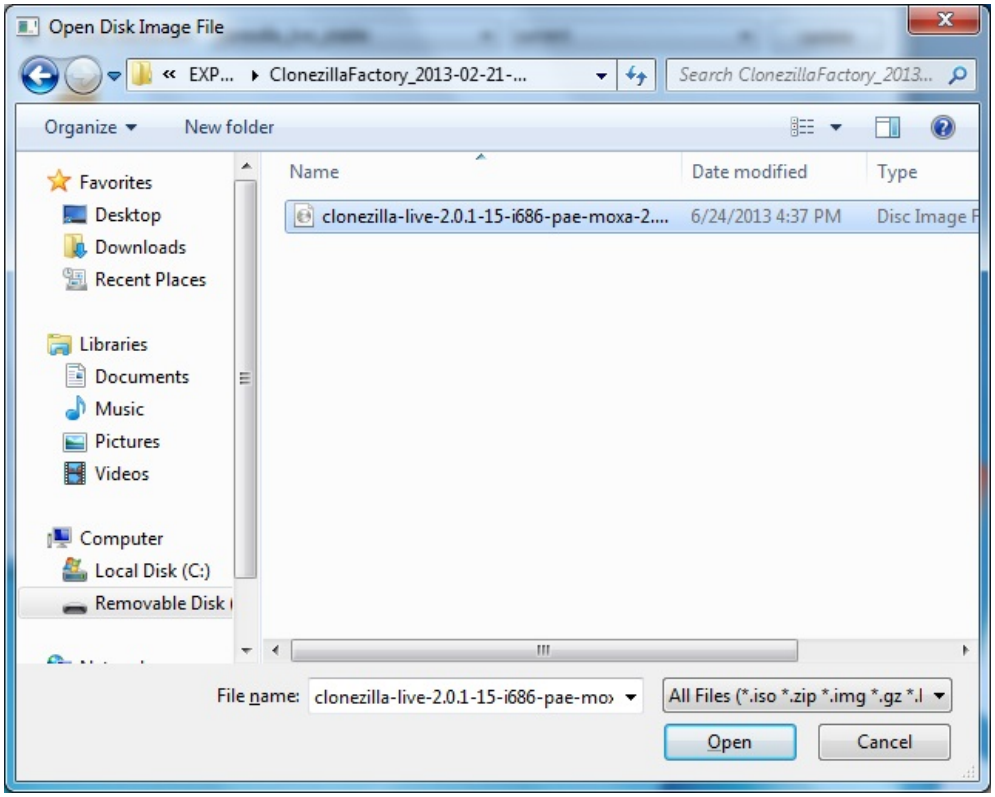
Recovery Procedure

Step 1: Prepare your USB drive

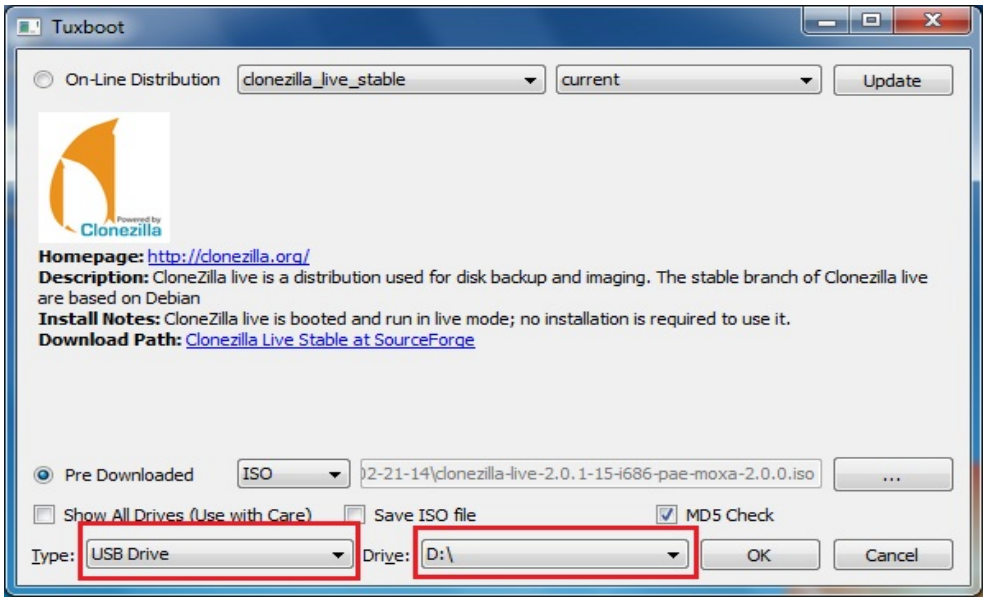
1. Execute **tuxboot-windows-23.exe** from the **utility_tools/CloneZilla** folder on the Software CD, select **Pre Download**, and then click "**...**"



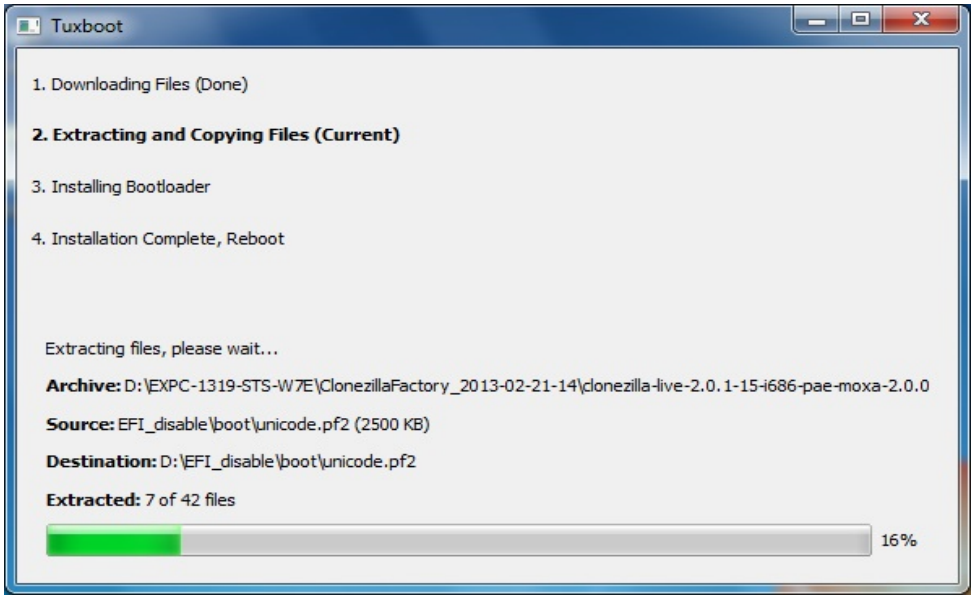
2. Select the ISO file in the directory **<Software DVD> \Recovery\V2416A-LX_Recovery\CloneZilla**



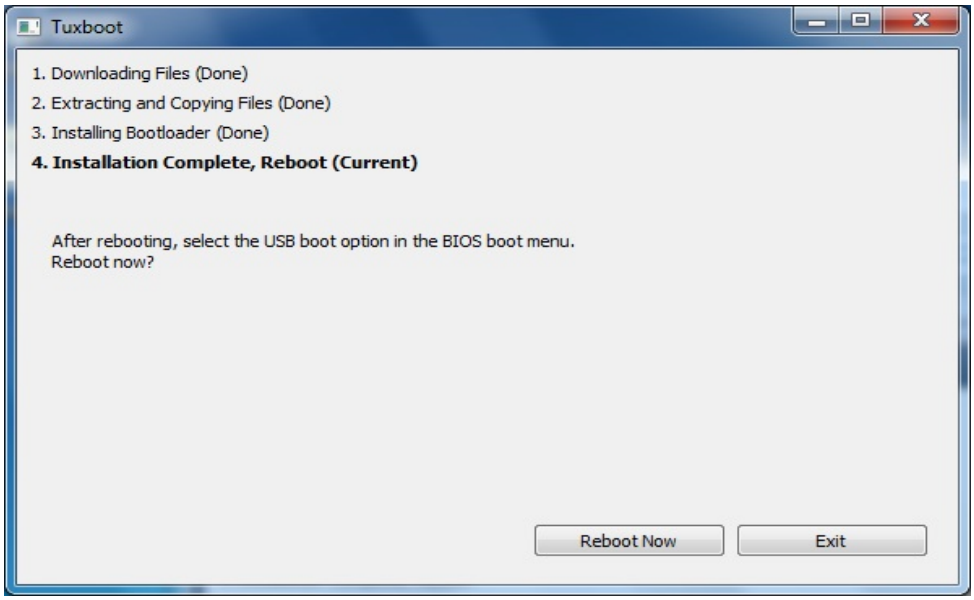
3. Select **USB Drive** type, select a **Drive**, and then click **OK** to continue.



4. The boot files will be copied to your USB drive.



5. When finished, click **Exit** to stop the program.



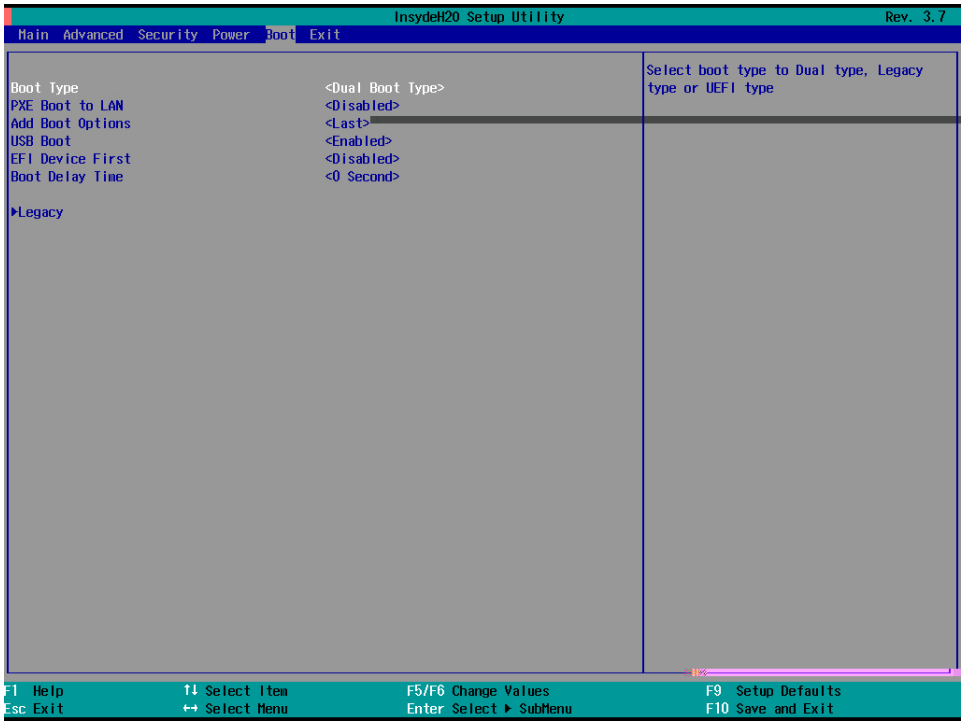
6. Manually copy the **os_image** directory from the <Software DVD> \Recovery\V2416A-LX_Recovery\CloneZilla\ folder on the Software DVD to **\home\partimag** on the USB drive.

Step 2: Change the BIOS Settings

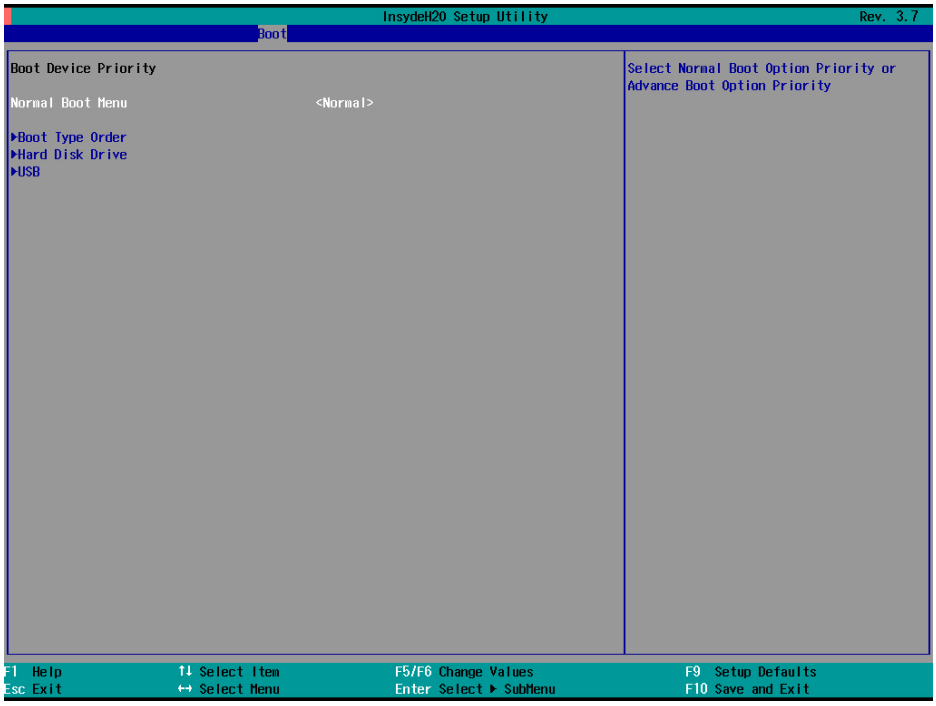
You will need to change the BIOS settings to boot from the USB disk.

1. Turn on the computer and press **F2.** Select **SCU** in the following screen.

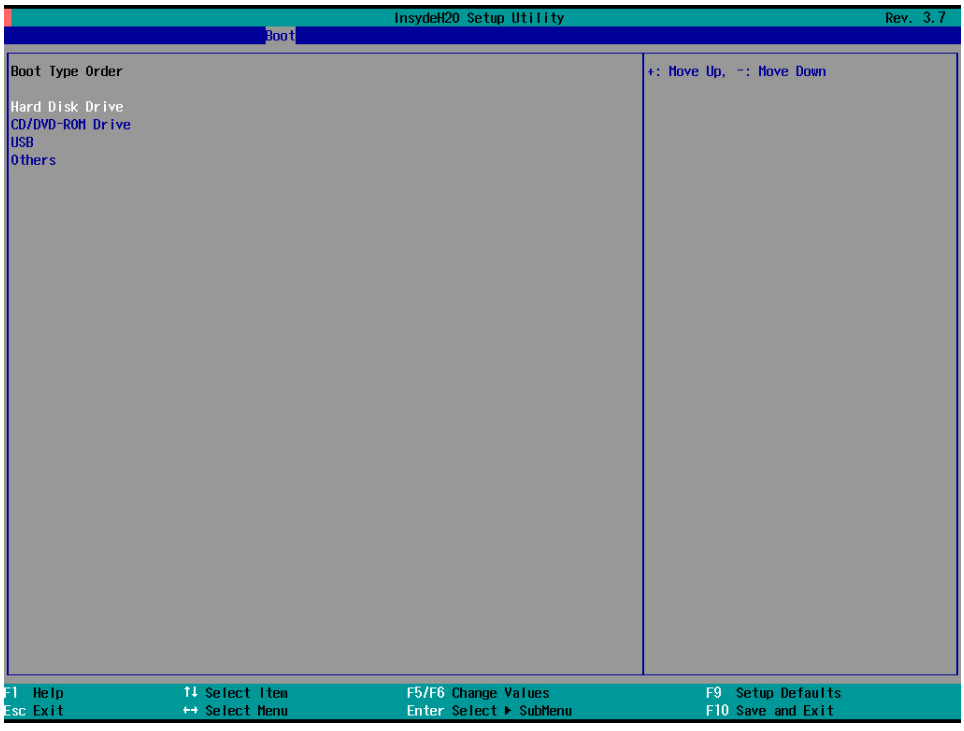
2. Select **Boot** and then select **Legacy**. Press **Enter** to continue.



3. Select **Boot Type Order**.



4. Select USB disk and then press "+" to move it to the first boot device position.
   Warning: An incorrect boot priority will lead to recovery failure.



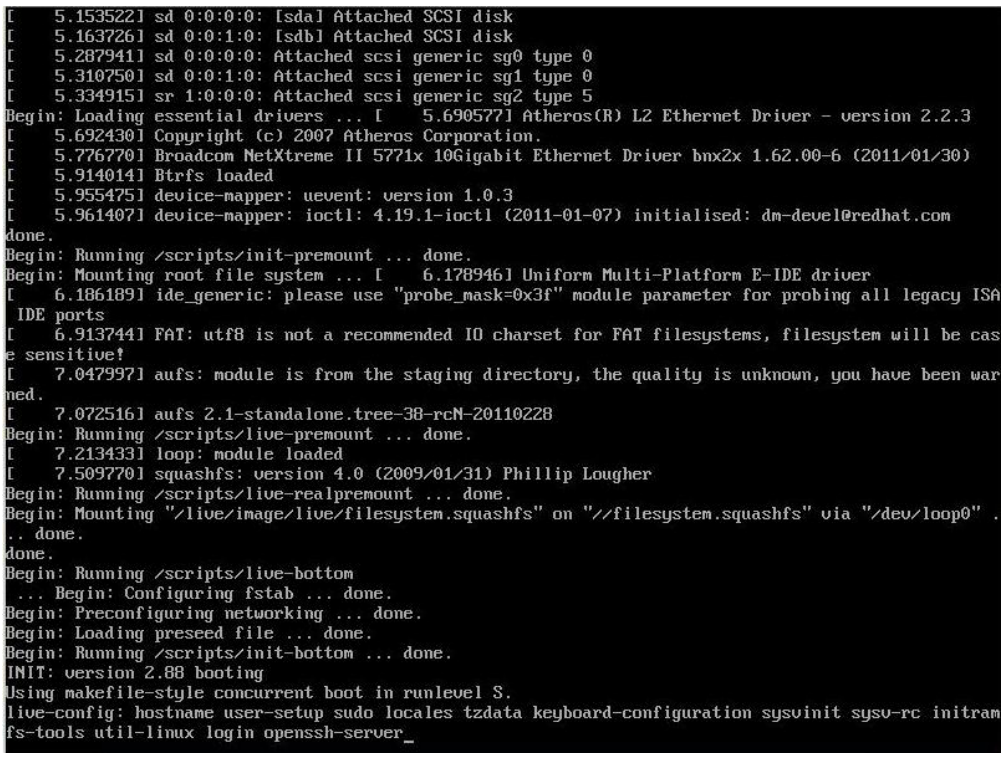
5. Press **F10** and then press **Enter** to save and exit the BIOS setup.

Step 3: Restore the system from the USB drive

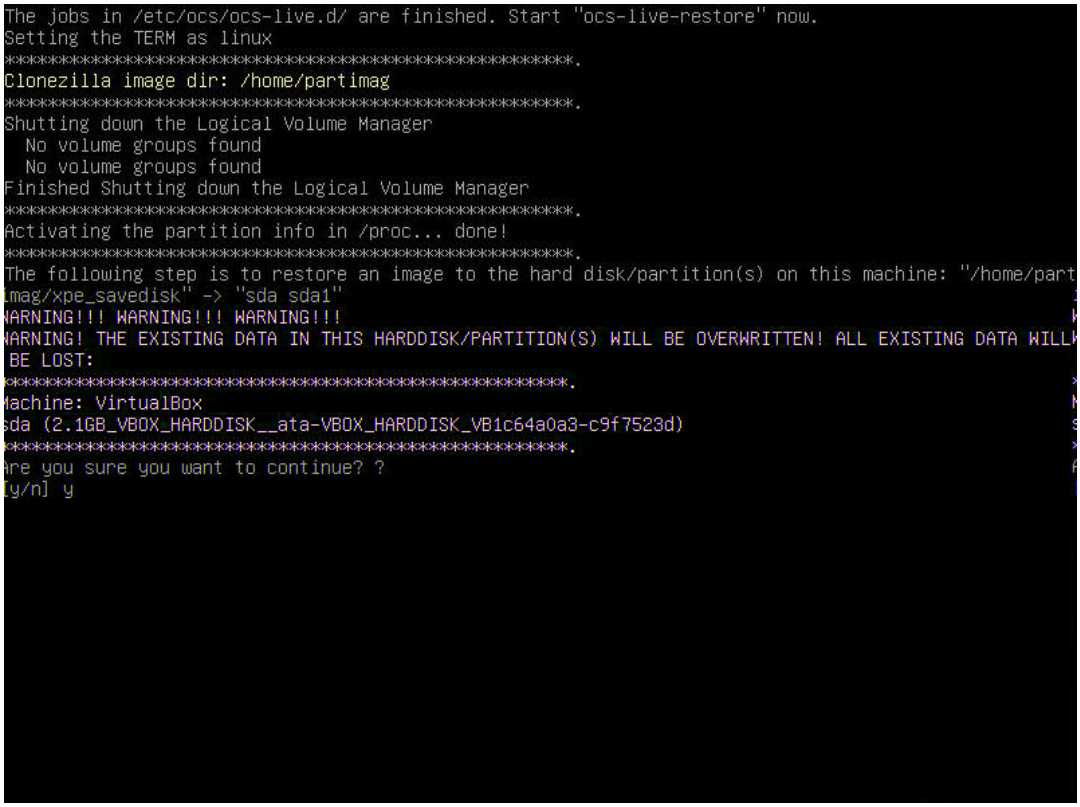
Connect the USB disk to any of the V2416A's USB ports and then reboot the computer. The system will boot from the USB disk and the Pre-installation Environment and the recovery utility will appear.

1. Select **clonezilla live restore disk**.

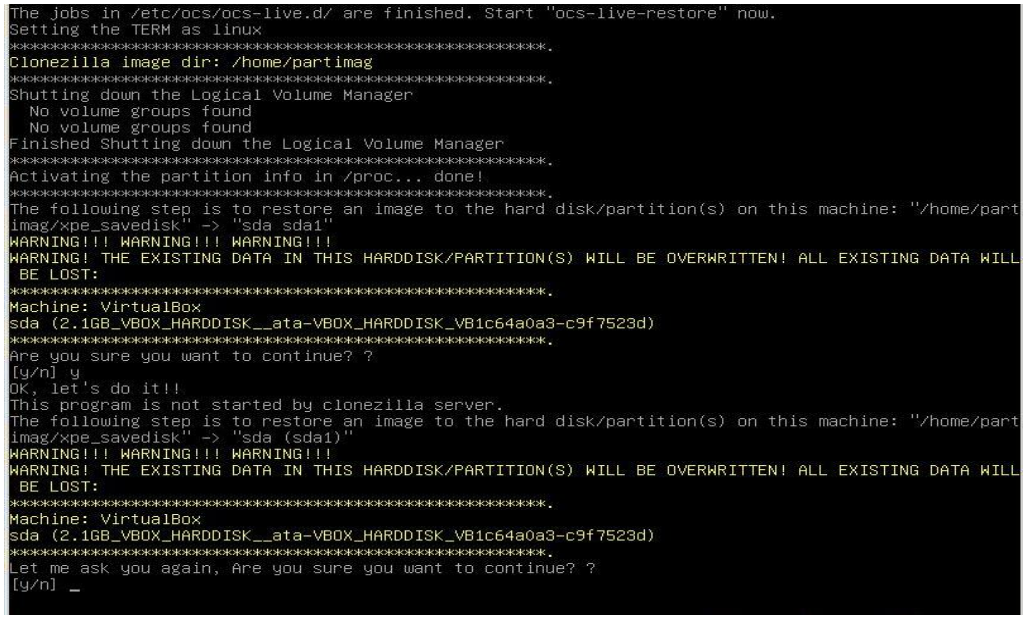
2. Wait for the USB drive boot process to finish.



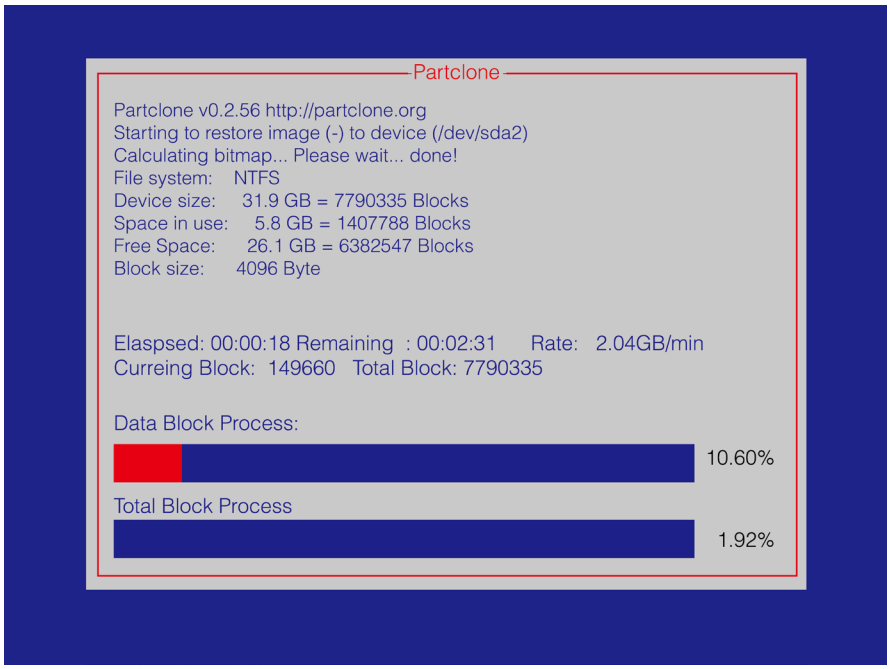
3. Enter **y** to continue the restore process.



4. Enter **y** to confirm again.



5. Wait for the process to finish.



6. Select **(0) Poweroff** to power off the computer.

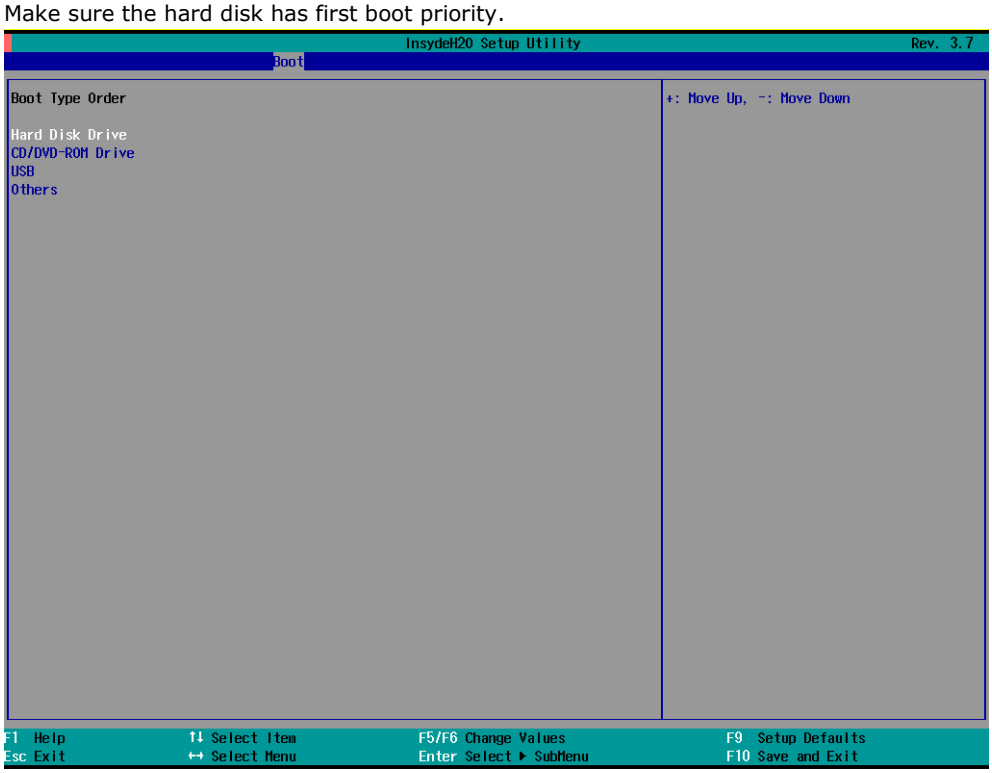
```
Restoring the first 446 bytes of MBR data, i.e. executable code area, for sda... done!
************************************************************.
Now resize the partition for sda1
ntfsresize -f /dev/sda1
ntfsresize v2.0.0 (libntfs 10:0:0)
Device name        : /dev/sda1
NTFS volume version: 3.1
Cluster size       : 2048 bytes
Current volume size: 2064511488 bytes (2065 MB)
Current device size: 2064513024 bytes (2065 MB)
New volume size    : 2064511488 bytes (2065 MB)
Nothing to do: NTFS volume size is already OK.
************************************************************.
The grub directory is NOT found. Maybe it does not exist (so other boot manager exists) or the file
system is not supported in the kernel. Skip running grub-install.
************************************************************.
Found NTFS boot partition among the restored partition(s): /dev/sda1
Head and sector no. of /dev/sda from EDD: 64, 63.
The start sector of NTFS partition /dev/sda1: 63
Adjust filesystem geometry for the NTFS partition: /dev/sda1
Running: partclone.ntfsfixboot -w -h 64 -t 63 -s 63 /dev/sda1
ntfsfixboot version 0.9
done!
************************************************************.
************************************************************
************************************************************
This program is not started by Clonezilla server, so skip notifying it the job is done.
Finished!
Now syncing - flush filesystem buffers...

"ocs-live-restore" is finished.
Now you can choose to:
(0) Poweroff
(1) Reboot
(2) Enter command line prompt
(3) Start over
[2]
```

7. Remove the USB drive after the computer has been powered off.

Step 4: Change the BIOS Settings to Boot from the Original Disk

Now you will need to change the boot priority so that it can boot from the original disk. As the system reboots, press **F2** to enter the BIOS setup menu.

1. Select **Hard Disk Drive** and then press + to move to the first boot device position, and then press **Enter**. Make sure the hard disk has first boot priority.



2. Press **F10** and then press **Enter** to save and exit BIOS settings.

Step 5: Reboot the Computer

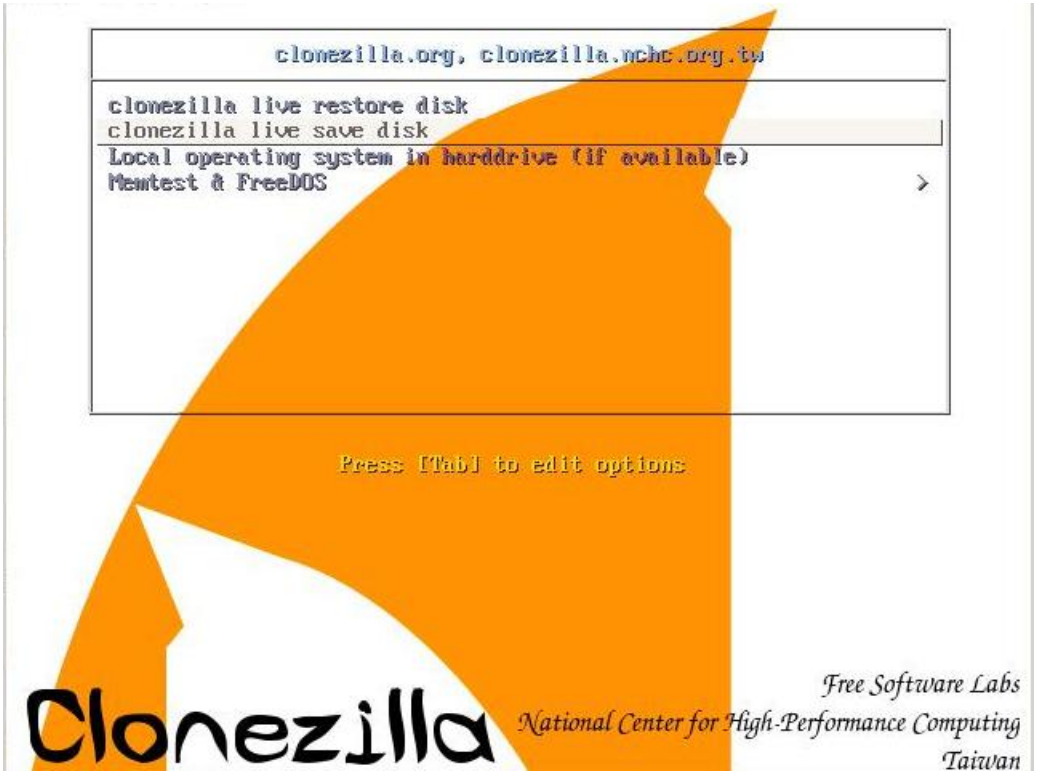
You need to wait about 10 to 15 minutes for the system to restart since the system configuration files will be initiated while booting up for the first time. **Do not turn off the computer or shut down the computer** while the system is restarting; otherwise, the IIS service will be terminated. When the operating system has successfully launched, you will need to restart your computer so that the new settings can be activated.

Saving the System to the USB Drive

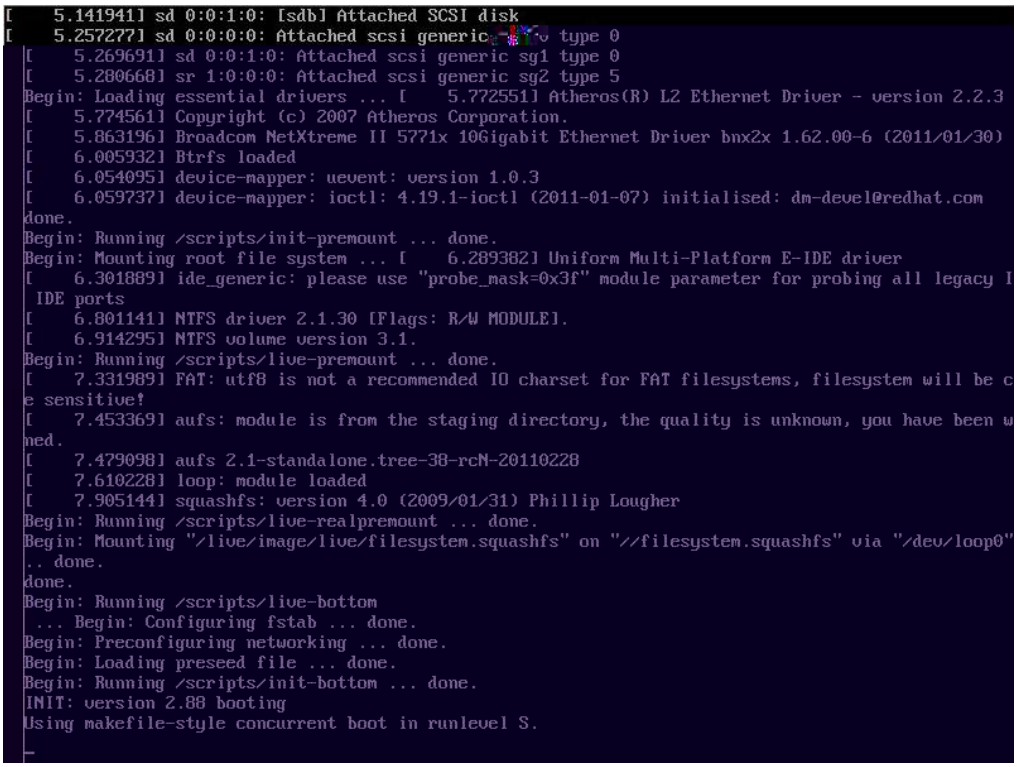
You may also save the current system to the USB drive for system recovery in case the system crashes. Before saving the system to the USB drive, we suggest removing all files under **\home\partimag** on the USB drive. In addition, change the BIOS settings to make the USB drive the first boot priority.

When the system has been launched, take the following steps.

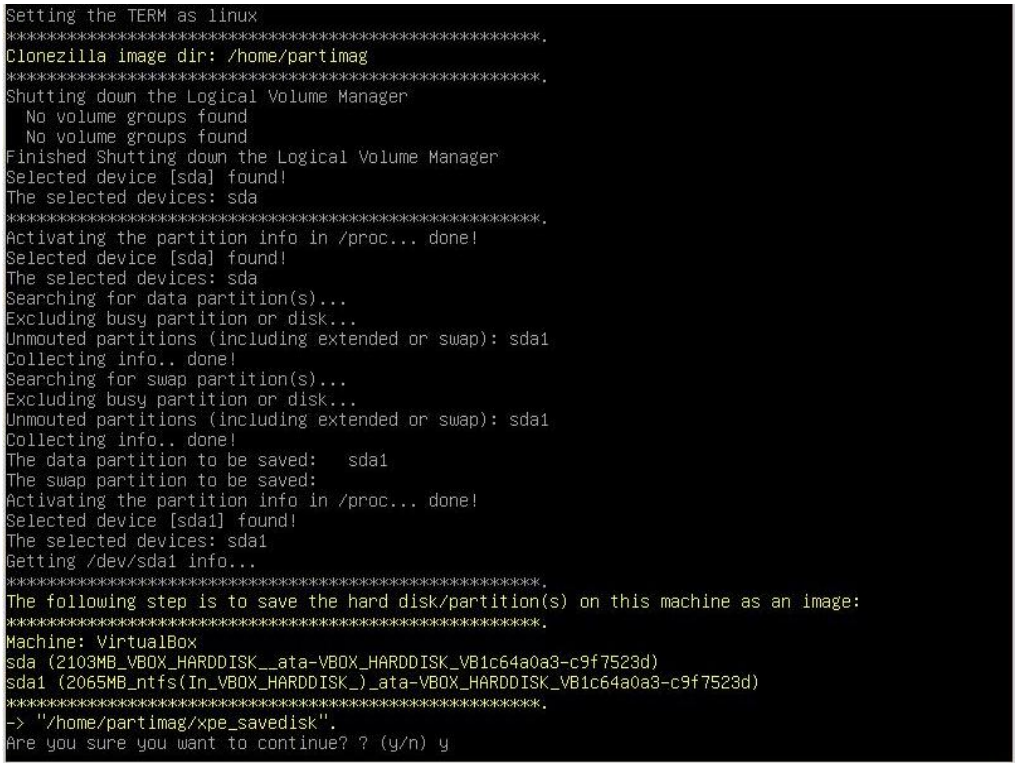
1. Select **clonezilla live save disk.**



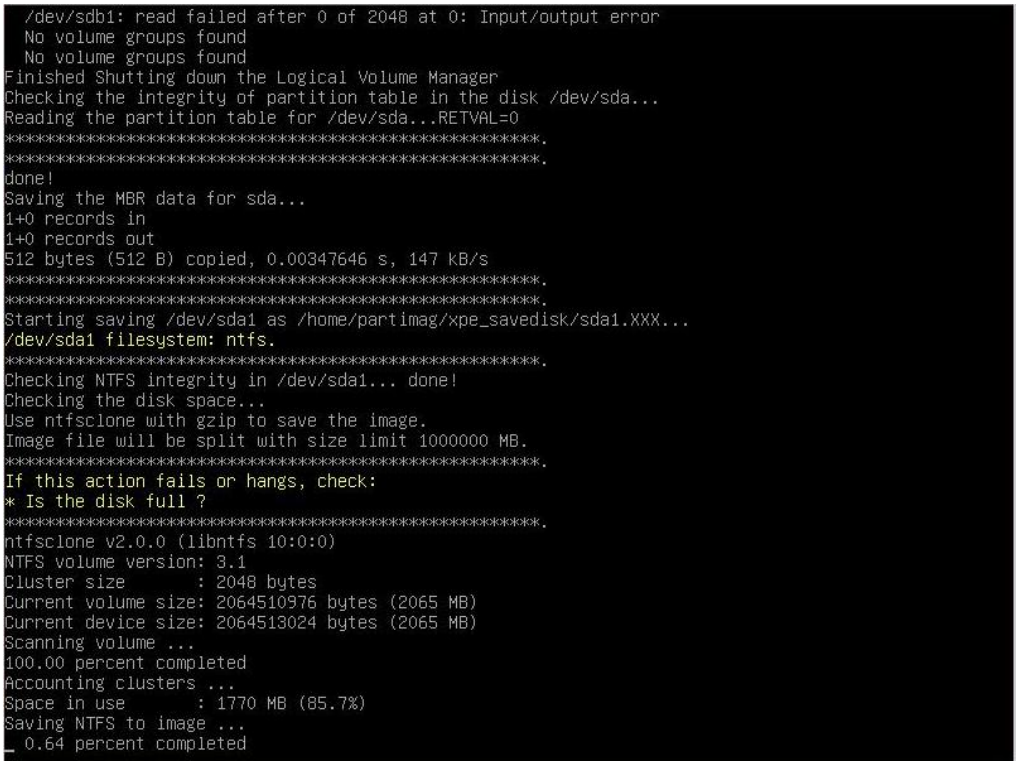
2. Wait for the USB drive boot process to finish.



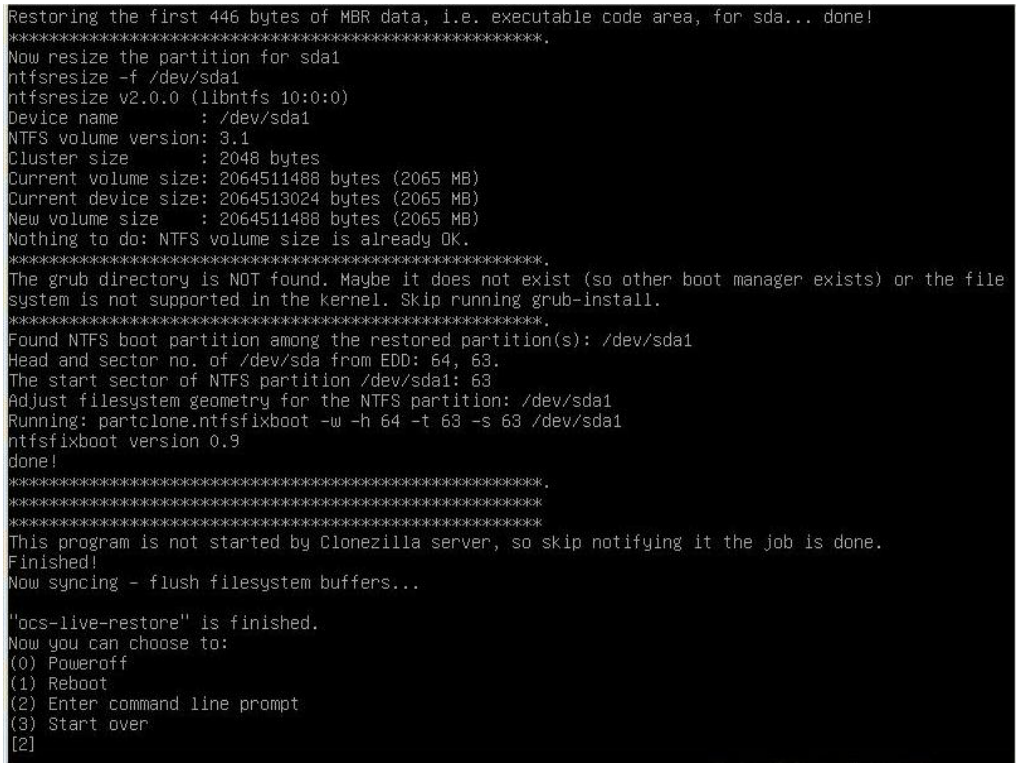
3. Enter **y** to continue.



4. Wait for the process to finish.



5. Select **(0) Poweroff** so that the computer will power off when the process is finished.



A

Software Components

acpi	1.6-1	displays information on ACPI devices
acpi-support-base	0.140-5+deb7u3	scripts for handling base ACPI events such as the power button
acpid	1:2.0.16-1+deb7u1	Advanced Configuration and Power Interface event daemon
adduser	3.113+nmu3	add and remove users and groups
alsa-base	1.0.25+3~deb7u1	ALSA driver configuration files
alsa-utils	1.0.25-4	Utilities for configuring and using ALSA
apache2	2.2.22-13+deb7u4	Apache HTTP Server metapackage
apache2-mpm-prefork	2.2.22-13+deb7u4	Apache HTTP Server - traditional non-threaded model
apache2-utils	2.2.22-13+deb7u4	utility programs for webservers
apache2.2-bin	2.2.22-13+deb7u4	Apache HTTP Server common binary files
apache2.2-common	2.2.22-13+deb7u4	Apache HTTP Server common files
apt	0.9.7.9+deb7u7	commandline package manager
apt-listchanges	2.85.11	package change history notification tool
apt-utils	0.9.7.9+deb7u7	package managment related utility programs
aptitude	0.6.8.2-1	terminal-based package manager
aptitude-common	0.6.8.2-1	architecture indepedent files for the aptitude package manager
at	3.1.13-2+deb7u1	Delayed job execution and batch processing
base-files	7.1wheezy8	Debian base system miscellaneous files
base-passwd	3.5.26	Debian base system master password and group files
bash	4.2+dfsg-0.1+deb7u3	GNU Bourne Again SHell
bash-completion	1:2.0-1	programmable completion for the bash shell
bc	1.06.95-2+b1	The GNU bc arbitrary precision calculator language
bind9-host	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	Version of 'host' bundled with BIND 9.X
binutils	2.22-8+deb7u2	GNU assembler, linker and binary utilities
bridge-utils	1.5-6	Utilities for configuring the Linux Ethernet bridge
bsdmainutils	9.0.3	collection of more utilities from FreeBSD
bsdutils	1:2.20.1-5.3	Basic utilities from 4.4BSD-Lite
build-essential	11.5	Informational list of build-essential packages
busybox	1:1.20.0-7	Tiny utilities for small and embedded systems
bzip2	1.0.6-4	high-quality block-sorting file compressor - utilities
ca-certificates	20130119+deb7u1	Common CA certificates
console-setup	1.88	console font and keymap setup program
console-setup-linux	1.88	Linux specific part of console-setup
coreutils	8.13-3.5	GNU core utilities
cpio	2.11+dfsg-0.1+deb7u1	GNU cpio -- a program to manage archives of files
cpp	4:4.7.2-1	GNU C preprocessor (cpp)
cpp-4.7	4.7.2-5	GNU C preprocessor
cron	3.0pl1-124	process scheduling daemon
dash	0.5.7-3	POSIX-compliant shell

db5.1-util	5.1.29-5	Berkeley v5.1 Database Utilities
dbus	1.6.8-1+deb7u6	simple interprocess messaging system (daemon and utilities)
dc	1.06.95-2+b1	The GNU dc arbitrary precision reverse-polish calculator
debconf	1.5.49	Debian configuration management system
debconf-i18n	1.5.49	full internationalization support for debconf
debian-archive-keyring	2014.3~deb7u1	GnuPG archive keys of the Debian archive
debian-faq	5.0.1	Debian FAQ
debianutils	4.3.2	Miscellaneous utilities specific to Debian
diffutils	1:3.2-6	File comparison utilities
discover	2.1.2-5.2	hardware identification system
discover-data	2.2010.10.18	Data lists for Discover hardware detection system
dmidecode	2.11-9	SMBIOS/DMI table decoder
dmsetup	2:1.02.74-8	Linux Kernel Device Mapper userspace library
dnsutils	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	Clients provided with BIND
dpkg	1.16.15	Debian package management system
dpkg-dev	1.16.15	Debian package development tools
e2fslibs:amd64	1.42.5-1.1	ext2/ext3/ext4 file system libraries
e2fsprogs	1.42.5-1.1	ext2/ext3/ext4 file system utilities
ethtool	1:3.4.2-1	display or change Ethernet device settings
fakeroot	1.18.4-2	tool for simulating superuser privileges
file	5.11-2+deb7u6	Determines file type using "magic" numbers
findutils	4.4.2-4	utilities for finding files--find, xargs
firmware-linux-free	3.2	Binary firmware for various drivers in the Linux kernel
firmware-ralink	0.36+wheezy.1	Binary firmware for Ralink wireless cards
firmware-realtek	0.43~bpo70+1	Binary firmware for Realtek wired and wireless network adapters
g++	4:4.7.2-1	GNU C++ compiler
g++-4.7	4.7.2-5	GNU C++ compiler
gcc	4:4.7.2-1	GNU C compiler
gcc-4.7	4.7.2-5	GNU C compiler
gcc-4.7-base:amd64	4.7.2-5	GCC, the GNU Compiler Collection (base package)
geoip-database	20130213-1	IP lookup command line tools that use the GeoIP library (country database)
gettext-base	0.18.1.1-9	GNU Internationalization utilities for the base system
gnupg	1.4.12-7+deb7u6	GNU privacy guard - a free PGP replacement
gpgv	1.4.12-7+deb7u6	GNU privacy guard - signature verification tool
grep	2.12-2	GNU grep, egrep and fgrep
groff-base	1.21-9	GNU troff text-formatting system (base system components)
grub-common	1.99-27+deb7u2	GRand Unified Bootloader (common files)
grub-pc	1.99-27+deb7u2	GRand Unified Bootloader, version 2 (PC/BIOS version)
grub-pc-bin	1.99-27+deb7u2	GRand Unified Bootloader, version 2 (PC/BIOS binaries)
grub2-common	1.99-27+deb7u2	GRand Unified Bootloader (common files for version 2)
gzip	1.5-1.1	GNU compression utilities
hdparm	9.39-1+b1	tune hard disk parameters for high performance
host	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	Transitional package
hostname	3.11	utility to set/show the host name or domain name
ifupdown	0.7.8	high level tools to configure network interfaces
initramfs-tools	0.115~bpo70+1	generic modular initramfs generator
initscripts	2.88dsf-41+deb7u1	scripts for initializing and shutting down the system

insserv	1.14.0-5	boot sequence organizer using LSB init.d script dependency information
install-info	4.13a.dfsg.1-10	Manage installed documentation in info format
iproute	20120521-3+b3	networking and traffic control tools
iptables	1.4.14-3.1	administration tools for packet filtering and NAT
iputils-ping	3:20101006-1+b1	Tools to test the reachability of network hosts
irqbalance	1.0.6-2~bpo70+1	Daemon to balance interrupts for SMP systems
isc-dhcp-client	4.2.2.dfsg.1-5+deb70u6	ISC DHCP client
isc-dhcp-common	4.2.2.dfsg.1-5+deb70u6	common files used by all the isc-dhcp* packages
iso-codes	3.41-1	ISO language, territory, currency, script codes and their translations
kbd	1.15.3-9	Linux console font and keytable utilities
keyboard-configuration	1.88	system-wide keyboard preferences
klibc-utils	2.0.1-3.1	small utilities built with klibc for early boot
kmod	9-3	tools for managing Linux kernel modules
krb5-locales	1.10.1+dfsg-5+deb7u2	Internationalization support for MIT Kerberos
laptop-detect	0.13.7	attempt to detect a laptop
less	444-4	pager program similar to more
libacl1:amd64	2.2.51-8	Access control list shared library
libalgorithm-diff-perl	1.19.02-2	module to find differences between files
libalgorithm-diff-xs-perl	0.04-2+b1	module to find differences between files (XS accelerated)
libalgorithm-merge-perl	0.08-2	Perl module for three-way merge of textual data
libapache2-mod-php5	5.4.38-0+deb7u1	server-side, HTML-embedded scripting language (Apache 2 module)
libapr1	1.4.6-3+deb7u1	Apache Portable Runtime Library
libaprutil1	1.4.1-3	Apache Portable Runtime Utility Library
libaprutil1-dbd-sqlite3	1.4.1-3	Apache Portable Runtime Utility Library - SQLite3 Driver
libaprutil1-ldap	1.4.1-3	Apache Portable Runtime Utility Library - LDAP Driver
libapt-inst1.5:amd64	0.9.7.9+deb7u7	deb package format runtime library
libapt-pkg4.12:amd64	0.9.7.9+deb7u7	package managment runtime library
libasound2:amd64	1.0.25-4	shared library for ALSA applications
libasprintf0c2:amd64	0.18.1.1-9	GNU library to use fprintf and friends in C++
libattr1:amd64	1:2.4.46-8	Extended attribute shared library
libbind9-80	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	BIND9 Shared Library used by BIND
libblkid1:amd64	2.20.1-5.3	block device id library
libboost-iostreams1.49.0	1.49.0-3.2	Boost.Iostreams Library
libbsd0:amd64	0.4.2-1	utility functions from BSD systems - shared library
libbz2-1.0:amd64	1.0.6-4	high-quality block-sorting file compressor library - runtime
libc-bin	2.13-38+deb7u8	Embedded GNU C Library: Binaries
libc-dev-bin	2.13-38+deb7u8	Embedded GNU C Library: Development binaries
libc6:amd64	2.13-38+deb7u8	Embedded GNU C Library: Shared libraries
libc6-dev:amd64	2.13-38+deb7u8	Embedded GNU C Library: Development Libraries and Header Files
libcap-ng0	0.6.6-2	An alternate POSIX capabilities library
libcap2:amd64	1:2.22-1.2	support for getting/setting POSIX.1e capabilities
libclass-isa-perl	0.36-3	report the search path for a class's ISA tree
libcomerr2:amd64	1.42.5-1.1	common error description library
libcwidget3	0.5.16-3.4	high-level terminal interface library for C++ (runtime files)
libdb5.1:amd64	5.1.29-5	Berkeley v5.1 Database Libraries [runtime]

libdbus-1-3:amd64	1.6.8-1+deb7u6	simple interprocess messaging system (library)
libdevmapper1.02.1:amd64	2:1.02.74-8	Linux Kernel Device Mapper userspace library
libdiscover2	2.1.2-5.2	hardware identification library
libdns88	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	DNS Shared Library used by BIND
libdpkg-perl	1.16.15	Dpkg perl modules
libedit2:amd64	2.11-20080614-5	BSD editline and history libraries
libept1.4.12	1.0.9	High-level library for managing Debian package information
libevent-2.0-5:amd64	2.0.19-stable-3	Asynchronous event notification library
libexpat1:amd64	2.1.0-1+deb7u1	XML parsing C library - runtime library
libffi5:amd64	3.0.10-3	Foreign Function Interface library runtime
libfile-fcntllock-perl	0.14-2	Perl module for file locking with fcntl(2)
libfreetype6:amd64	2.4.9-1.1	FreeType 2 font engine, shared library files
libfuse2:amd64	2.9.0-2+deb7u1	Filesystem in Userspace (library)
libgc1c2	1:7.1-9.1	conservative garbage collector for C and C++
libgcc1:amd64	1:4.7.2-5	GCC support library
libgcrypt11:amd64	1.5.0-5+deb7u2	LGPL Crypto library - runtime library
libgdbm3:amd64	1.8.3-11	GNU dbm database routines (runtime version)
libgeoip1	1.4.8+dfsg-3	non-DNS IP-to-country resolver library
libglib2.0-0:amd64	2.33.12+really2.32.4-5	GLib library of C routines
libglib2.0-data	2.33.12+really2.32.4-5	Common files for GLib library
libgmp10:amd64	2:5.0.5+dfsg-2	Multiprecision arithmetic library
libgnutls26:amd64	2.12.20-8+deb7u2	GNU TLS library - runtime library
libgomp1:amd64	4.7.2-5	GCC OpenMP (GOMP) support library
libgpg-error0:amd64	1.10-3.1	library for common error values and messages in GnuPG components
libgpgme11	1.2.0-1.4+deb7u1	GPGME - GnuPG Made Easy
libgpm2:amd64	1.20.4-6	General Purpose Mouse - shared library
libgssapi-krb5-2:amd64	1.10.1+dfsg-5+deb7u2	MIT Kerberos runtime libraries - krb5 GSS-API Mechanism
libgssglue1:amd64	0.4-2	mechanism-switch gssapi library
libidn11:amd64	1.25-2	GNU Libidn library, implementation of IETF IDN specifications
libisc84	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	ISC Shared Library used by BIND
libisccc80	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	Command Channel Library used by BIND
libisccfg82	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	Config File Handling Library used by BIND
libitm1:amd64	4.7.2-5	GNU Transactional Memory Library
libiw30:amd64	30~pre9-8	Wireless tools - library
libk5crypto3:amd64	1.10.1+dfsg-5+deb7u2	MIT Kerberos runtime libraries - Crypto Library
libkeyutils1:amd64	1.5.5-3+deb7u1	Linux Key Management Utilities (library)
libklibc	2.0.1-3.1	minimal libc subset for use with initramfs
libkmod2:amd64	9-3	libkmod shared library
libkrb5-3:amd64	1.10.1+dfsg-5+deb7u2	MIT Kerberos runtime libraries
libkrb5support0:amd64	1.10.1+dfsg-5+deb7u2	MIT Kerberos runtime libraries - Support library
libldap-2.4-2:amd64	2.4.31-1+nmu2	OpenLDAP libraries
liblocale-gettext-perl	1.05-7+b1	module using libc functions for internationalization in Perl
liblockfile-bin	1.09-5	support binaries for and cli utilities based on liblockfile

liblockfile1:amd64	1.09-5	NFS-safe locking library
liblwres80	1:9.8.4.dfsg.P1-6+nmu2+deb7u3	Lightweight Resolver Library used by BIND
liblzma5:amd64	5.1.1alpha+20120614-2	XZ-format compression library
liblzo2-2:amd64	2.06-1+deb7u1	data compression library
libmagic1:amd64	5.11-2+deb7u6	File type determination library using "magic" numbers
libmount1	2.20.1-5.3	block device id library
libmpc2:amd64	0.9-4	multiple precision complex floating-point library
libmpfr4:amd64	3.1.0-5	multiple precision floating-point computation
libncurses5:amd64	5.9-10	shared libraries for terminal handling
libncurses5-dev	5.9-10	developer's libraries for ncurses
libncursesw5:amd64	5.9-10	shared libraries for terminal handling (wide character support)
libnet-telnet-perl	3.03-3	Script telnetable connections
libnewt0.52	0.52.14-11.1	Not Erik's Windowing Toolkit - text mode windowing with slang
libnfnetlink0	1.0.0-1.1	Netfilter netlink library
libnfsidmap2:amd64	0.25-4	NFS idmapping library
libnl-3-200:amd64	3.2.7-4	library for dealing with netlink sockets
libnl-genl-3-200:amd64	3.2.7-4	library for dealing with netlink sockets - generic netlink
libnuma1	2.0.8~rc4-1	Libraries for controlling NUMA policy
libonig2	5.9.1-1	Oniguruma regular expressions library
libp11-kit0:amd64	0.12-3	Library for loading and coordinating access to PKCS#11 modules - runtime
libpam-modules:amd64	1.1.3-7.1	Pluggable Authentication Modules for PAM
libpam-modules-bin	1.1.3-7.1	Pluggable Authentication Modules for PAM - helper binaries
libpam-runtime	1.1.3-7.1	Runtime support for the PAM library
libpam0g:amd64	1.1.3-7.1	Pluggable Authentication Modules library
libpci3:amd64	1:3.1.9-6	Linux PCI Utilities (shared library)
libpcre3:amd64	1:8.30-5	Perl 5 Compatible Regular Expression Library - runtime files
libpcsclite1:amd64	1.8.4-1+deb7u1	Middleware to access a smart card using PC/SC (library)
libperl-dev	5.14.2-21+deb7u2	Perl library: development files
libperl5.14	5.14.2-21+deb7u2	shared Perl library
libpipeline1:amd64	1.2.1-1	pipeline manipulation library
libpkcs11-helper1:amd64	1.09-1	library that simplifies the interaction with PKCS#11
libpopt0:amd64	1.16-7	lib for parsing cmdline parameters
libprocps0:amd64	1:3.3.3-3	library for accessing process information from /proc
libpth20	2.0.7-16	The GNU Portable Threads
libqdbm14	1.8.78-2	QDBM Database Libraries without GDBM wrapper[runtime]
libquadmath0:amd64	4.7.2-5	GCC Quad-Precision Math Library
libreadline5:amd64	5.2+dfsg-2~deb7u1	GNU readline and history libraries, run-time libraries
libreadline6:amd64	6.2+dfsg-0.1	GNU readline and history libraries, run-time libraries
libsamplerate0:amd64	0.1.8-5	Audio sample rate conversion library
libsasl2-2:amd64	2.1.25.dfsg1-6+deb7u1	Cyrus SASL - authentication abstraction library
libsasl2-modules:amd64	2.1.25.dfsg1-6+deb7u1	Cyrus SASL - pluggable authentication modules
libselinux1:amd64	2.1.9-5	SELinux runtime shared libraries
libsemanage-common	2.1.6-6	Common files for SELinux policy management libraries
libsemanage1:amd64	2.1.6-6	SELinux policy management library
libsensors4:amd64	1:3.3.2-2+deb7u1	library to read temperature/voltage/fan sensors

libsepol1:amd64	2.1.4-3	SELinux library for manipulating binary security policies
libsigc++-2.0-0c2a:amd64	2.2.10-0.2	type-safe Signal Framework for C++ - runtime
libslang2:amd64	2.2.4-15	S-Lang programming library - runtime version
libsnmp-base	5.4.3~dfsg-2.8+deb7u1	SNMP (Simple Network Management Protocol) MIBs and documentation
libsnmp15	5.4.3~dfsg-2.8+deb7u1	SNMP (Simple Network Management Protocol) library
libsqlite3-0:amd64	3.7.13-1+deb7u1	SQLite 3 shared library
libsqlite3-dev	3.7.13-1+deb7u1	SQLite 3 development files
libss2:amd64	1.42.5-1.1	command-line interface parsing library
libssl1.0.0:amd64	1.0.1e-2+deb7u13	SSL shared libraries
libstdc++6:amd64	4.7.2-5	GNU Standard C++ Library v3
libstdc++6-4.7-dev	4.7.2-5	GNU Standard C++ Library v3 (development files)
libswitch-perl	2.16-2	switch statement for Perl
libsystemd-login0:amd64	44-11+deb7u4	systemd login utility library
libtasn1-3:amd64	2.13-2+deb7u1	Manage ASN.1 structures (runtime)
libtext-charwidth-perl	0.04-7+b1	get display widths of characters on the terminal
libtext-iconv-perl	1.7-5	converts between character sets in Perl
libtext-wrapi18n-perl	0.06-7	internationalized substitute of Text::Wrap
libtimedate-perl	1.2000-1	collection of modules to manipulate date/time information
libtinfo-dev:amd64	5.9-10	developer's library for the low-level terminfo library
libtinfo5:amd64	5.9-10	shared low-level terminfo library for terminal handling
libtirpc1:amd64	0.2.2-5	transport-independent RPC library
libtokyocabinet9:amd64	1.4.47-2	Tokyo Cabinet Database Libraries [runtime]
libudev0:amd64	175-7.2	libudev shared library
libusb-0.1-4:amd64	2:0.1.12-20+nmu1	userspace USB programming library
libusb-1.0-0:amd64	2:1.0.11-1	userspace USB programming library
libustr-1.0-1:amd64	1.0.4-3	Micro string library: shared library
libuuid-perl	0.02-5	Perl extension for using UUID interfaces as defined in e2fsprogs
libuuid1:amd64	2.20.1-5.3	Universally Unique ID library
libwrap0:amd64	7.6.q-24	Wietse Venema's TCP wrappers library
libx11-6:amd64	2:1.5.0-1+deb7u1	X11 client-side library
libx11-data	2:1.5.0-1+deb7u1	X11 client-side library
libx86-1:amd64	1.1+ds1-10	x86 real-mode library
libxapian22	1.2.12-2	Search engine library
libxau6:amd64	1:1.0.7-1	X11 authorisation library
libxcb1:amd64	1.8.1-2+deb7u1	X C Binding
libxdmcp6:amd64	1:1.1.1-1	X11 Display Manager Control Protocol library
libxext6:amd64	2:1.3.1-2+deb7u1	X11 miscellaneous extension library
libxml2:amd64	2.8.0+dfsg1-7+wheezy2	GNOME XML library
libxmuu1:amd64	2:1.1.1-1	X11 miscellaneous micro-utility library
linux-base	3.5	Linux image base package
linux-image-3.16.0-0.bpo.4-amd64	3.16.7-ckt4-3~bpo70+1	Linux 3.16 for 64-bit PCs
linux-image-amd64	3.16+63~bpo70+1	Linux for 64-bit PCs (meta-package)
linux-libc-dev:amd64	3.2.65-1+deb7u2	Linux support headers for userspace development
locales	2.13-38+deb7u6	Embedded GNU C Library: National Language (locale) data [support]
lockfile-progs	0.1.17	Programs for locking and unlocking files and mailboxes
login	1:4.1.5.1-1	system login tools

logrotate	3.8.1-4	Log rotation utility
lsb-base	4.1+Debian8+deb7u1	Linux Standard Base 4.1 init script functionality
lsb-release	4.1+Debian8+deb7u1	Linux Standard Base version reporting utility
lsof	4.86+dfsg-1	Utility to list open files
m4	1.4.16-3	a macro processing language
make	3.81-8.2	An utility for Directing compilation.
man-db	2.6.2-1	on-line manual pager
manpages	3.44-1	Manual pages about using a GNU/Linux system
manpages-dev	3.44-1	Manual pages about using GNU/Linux for development
mawk	1.3.3-17	a pattern scanning and text processing language
mime-support	3.52-1+deb7u1	MIME files 'mime.types' & 'mailcap', and support programs
mlocate	0.23.1-1	quickly find files on the filesystem based on their name
module-init-tools	9-3	transitional dummy package (module-init-tools to kmod)
mount	2.20.1-5.3	Tools for mounting and manipulating filesystems
multiarch-support	2.13-38+deb7u6	Transitional package to ensure multiarch compatibility
mutt	1.5.21-6.2+deb7u3	text-based mailreader supporting MIME, GPG, PGP and threading
ncurses-base	5.9-10	basic terminal type definitions
ncurses-bin	5.9-10	terminal-related programs and man pages
ncurses-term	5.9-10	additional terminal type definitions
net-tools	1.60-24.2	The NET-3 networking toolkit
netbase	5.0	Basic TCP/IP networking system
netcat-traditional	1.10-40	TCP/IP swiss army knife
nfs-common	1:1.2.6-4	NFS support files common to client and server
ntpdate	1:4.2.6.p5+dfsg-2+deb7u4	client for setting system time from NTP servers
openssh-blacklist	0.4.1+nmu1	list of default blacklisted OpenSSH RSA and DSA keys
openssh-blacklist-extra	0.4.1+nmu1	list of non-default blacklisted OpenSSH RSA and DSA keys
openssh-client	1:6.0p1-4+deb7u2	secure shell (SSH) client, for secure access to remote machines
openssh-server	1:6.0p1-4+deb7u2	secure shell (SSH) server, for secure access from remote machines
openssl	1.0.1e-2+deb7u14	Secure Socket Layer (SSL) binary and related cryptographic tools
openvpn	2.2.1-8+deb7u3	virtual private network daemon
os-prober	1.58	utility to detect other OSes on a set of drives
passwd	1:4.1.5.1-1	change and administer password and group data
patch	2.6.1-3	Apply a diff file to an original
pciutils	1:3.1.9-6	Linux PCI Utilities
perl	5.14.2-21+deb7u2	Larry Wall's Practical Extraction and Report Language
perl-base	5.14.2-21+deb7u2	minimal Perl system
perl-modules	5.14.2-21+deb7u2	Core Perl modules
php5	5.4.38-0+deb7u1	server-side, HTML-embedded scripting language (metapackage)
php5-cli	5.4.38-0+deb7u1	command-line interpreter for the php5 scripting language
php5-common	5.4.38-0+deb7u1	Common files for packages built from the php5 source
pm-utils	1.4.1-9	utilities and scripts for power management
pmount	0.9.23-2	mount removable devices as normal user
powermgmt-base	1.31	Common utils and configs for power management

procps	1:3.3.3-3	/proc file system utilities
psmisc	22.19-1+deb7u1	utilities that use the proc file system
python	2.7.3-4+deb7u1	interactive high-level object-oriented language (default version)
python-apt	0.8.8.2	Python interface to libapt-pkg
python-apt-common	0.8.8.2	Python interface to libapt-pkg (locales)
python-chardet	2.0.1-2	universal character encoding detector
python-debian	0.1.21	Python modules to work with Debian-related data formats
python-debianbts	1.11	Python interface to Debian's Bug Tracking System
python-fpconst	0.7.2-5	Utilities for handling IEEE 754 floating point special values
python-minimal	2.7.3-4+deb7u1	minimal subset of the Python language (default version)
python-reportbug	6.4.4+deb7u1	Python modules for interacting with bug tracking systems
python-soappy	0.12.0-4	SOAP Support for Python
python-support	1.0.15	automated rebuilding support for Python modules
python2.6	2.6.8-1.1	Interactive high-level object-oriented language (version 2.6)
python2.6-minimal	2.6.8-1.1	Minimal subset of the Python language (version 2.6)
python2.7	2.7.3-6+deb7u2	Interactive high-level object-oriented language (version 2.7)
python2.7-minimal	2.7.3-6+deb7u2	Minimal subset of the Python language (version 2.7)
readline-common	6.2+dfsg-0.1	GNU readline and history libraries, common files
rpcbind	0.2.0-8	converts RPC program numbers into universal addresses
rsyslog	5.8.11-3+deb7u2	reliable system and kernel logging daemon
sed	4.2.1-10	The GNU sed stream editor
sensible-utils	0.0.7	Utilities for sensible alternative selection
sgml-base	1.26+nmu4	SGML infrastructure and SGML catalog file support
shared-mime-info	1.3-1~bpo70+1	FreeDesktop.org shared MIME database and spec
snmpd	5.4.3~dfsg-2.8+deb7u1	SNMP (Simple Network Management Protocol) agents
sqlite3	3.7.13-1+deb7u1	Command line interface for SQLite 3
ssh	1:6.0p1-4+deb7u2	secure shell client and server (metapackage)
ssl-cert	1.0.32	simple debconf wrapper for OpenSSL
sudo	1.8.5p2-1+nmu1	Provide limited super user privileges to specific users
sysstat	10.0.5-1	system performance tools for Linux
sysv-rc	2.88dsf-41+deb7u1	System-V-like runlevel change mechanism
sysvinit	2.88dsf-41+deb7u1	System-V-like init utilities
sysvinit-utils	2.88dsf-41+deb7u1	System-V-like utilities
tar	1.26+dfsg-0.1	GNU version of the tar archiving utility
tasksel	3.14.1	Tool for selecting tasks for installation on Debian systems
tasksel-data	3.14.1	Official tasks used for installation of Debian systems
tcpd	7.6.q-24	Wietse Venema's TCP wrapper utilities
time	1.7-24	GNU time program for measuring CPU resource usage
traceroute	1:2.0.18-3	Traces the route taken by packets over an IPv4/IPv6 network
tzdata	2014j-0wheezy1	time zone and daylight-saving time data
ucf	3.0025+nmu3	Update Configuration File: preserve user changes to config files.
udev	175-7.2	/dev/ and hotplug management daemon
usbmount	0.0.22	automatically mount and unmount USB mass storage devices

usbutils	1:005-3	Linux USB utilities
util-linux	2.20.1-5.3	Miscellaneous system utilities
v2400a-cpu	1.0.0	cpu
v2400a-dio	1.0.0	dio driver for V2400A
v2400a-muestty	1.0.0	muestty is a utility for mxser
v2400a-mxupcie	1.0.0	mxupcie driver for V2400A
v2400a-superio	1.0.0	superio driver for V2400A
v2400a-watchdog	1.0.0	watchdog driver for V2400A
v2400a-ifenslave	1.0.0	ifenslave for V2400A
vbetool	1.1-2	run real-mode video BIOS code to alter hardware state
vim	2:7.3.547-7	Vi IMproved - enhanced vi editor
vim-common	2:7.3.547-7	Vi IMproved - Common files
vim-runtime	2:7.3.547-7	Vi IMproved - Runtime files
vim-tiny	2:7.3.547-7	Vi IMproved - enhanced vi editor - compact version
w3m	0.5.3-8	WWW browsable pager with excellent tables/frames support
wamerican	7.1-1	American English dictionary words for /usr/share/dict
watchdog	5.12-1	system health checker and software/hardware watchdog handler
wget	1.13.4-3+deb7u2	retrieves files from the web
whiptail	0.52.14-11.1	Displays user-friendly dialog boxes from shell scripts
whois	5.1.1~deb7u1	intelligent WHOIS client
wireless-tools	30~pre9-8	Tools for manipulating Linux Wireless Extensions
wpasupplicant	1.0-3+deb7u1	client support for WPA and WPA2 (IEEE 802.11i)
xauth	1:1.0.7-1	X authentication utility
xkb-data	2.5.1-3	X Keyboard Extension (XKB) configuration data
xml-core	0.13+nmu2	XML infrastructure and XML catalog file support
xz-utils	5.1.1alpha+20120614-2	XZ-format compression utilities
zlib1g:amd64	1:1.2.7.dfsg-13	compression library - runtime