MC-1100 Series Linux Software User's Manual

Edition 1.0, May 2017

www.moxa.com/product

MC-1100 Series Linux Software User’s Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

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 Europe

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

Moxa India

Tel:	+91-80-4172-9088
Fax:	+91-80-4132-1045

Moxa China (Shanghai office)

Toll-free:	800-820-5036
Tel:	+86-21-5258-9955
Fax:	+86-21-5258-5505

Moxa Asia-Pacific

Tel:	+886-2-8919-1230
Fax:	+886-2-8919-1231

Table of Contents

1

Introduction

Thank you for purchasing a Moxa MC-1100 series x86, ready-to-run embedded computer. This manual describes the software configuration and management process for the MC-1100 Linux models. For details regarding hardware installation, connector interfaces, setup, and BIOS upgrade, refer to the *MC-1100 Series 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 an embedded computer with a GNU cross compiler and minimum source-code modifications. A typical Linux-based device is designed for a specific use. It is often not connected to other computers. 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:

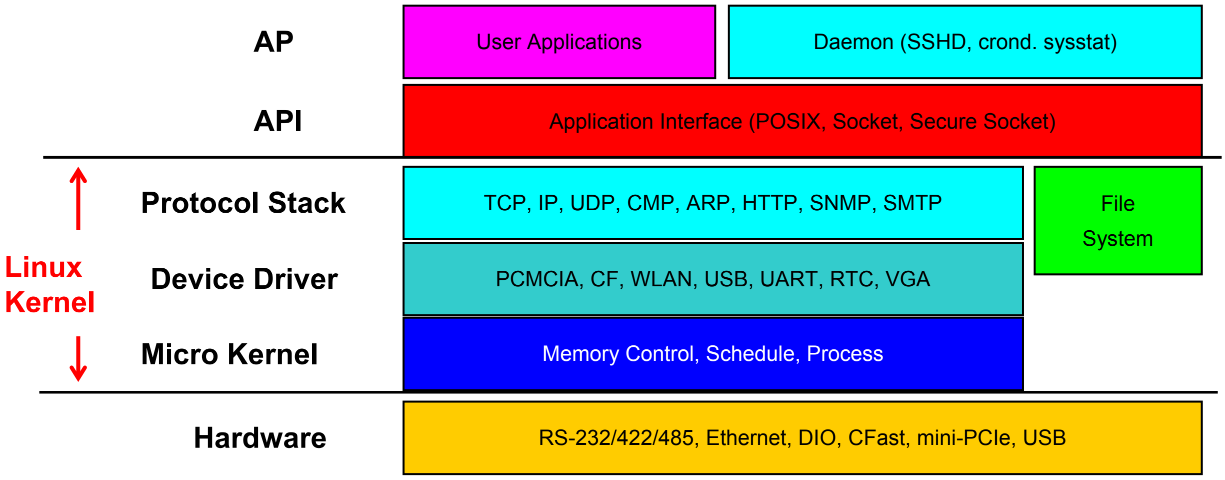
Overview

The MC-1100 series DIN-rail mountable computers are x86 platforms built on Intel® Atom™ series processor. They feature the most reliable I/O design with the highest number of connectivity options, including support for wireless modules (Wi-Fi/3G/LTE), making them suitable for a diverse range of communication applications.

With a wide operating temperature range (-40 to 70°C) and compliance with safety, EMI, and EMS standards, the MC-1100 series is ideal for intelligent computing and communication solutions operating in critical environments, including marine communication, oil & gas field site monitoring, and transportation.

Software Specifications

The Linux operating system preinstalled on the MC-1100 embedded computers is a **Debian Jessie** distribution. The Debian project consists of a group of volunteers located worldwide 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. You can use the GNU Tool Chain provided by Moxa to port programs on to the Debian Linux platform. In addition to standard POSIX APIs, device drivers for Moxa UART and other peripherals such as programmable LEDs and relay are also included along with the Linux operating system. An example that represents the software architecture is shown below:


AP
User Applications
Daemon (SSHD, crond. sysstat)
API
Application Interface (POSIX, Socket, Secure Socket)
Linux Kernel
Protocol Stack
TCP, IP, UDP, CMP, ARP, HTTP, SNMP, SMTP
File System
Device Driver
PCMCIA, CF, WLAN, USB, UART, RTC, VGA
Micro Kernel
Memory Control, Schedule, Process
Hardware
RS-232/422/485, Ethernet, DIO, CFast, mini-PCIe, USB

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 this illustration.

ATTENTION

For information and documentation regarding Debian GNU/Linux and the free software concept, refer to the following links:

- http://www.debian.org/
- http://www.gnu.org/

2

Software Configuration

This chapter covers some basic Linux operating system configuration on the MC-1100 Linux model. For advanced network configuration and management instructions refer to *Chapter 3, Managing Communications*.

The following topics are covered in this chapter:

- **Account Management**
- **Setting up Desktop Environments**
- **Starting from a VGA Console**
- **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**
- **Executing Scheduled Commands Using the Cron Daemon**
- **Inserting a USB Storage Device into the Computer**
- **Checking the Linux Version**
- **Installing and Removing Packages Using APT**
- **Power Management—Device Suspend**
- **Wake On LAN**

Account Management

Connect the MC-1100 to a display and turn on the computer. Enter the following information to log in the computer.

Login: moxa
Password: moxa

WARNING

Change the default password after the first login to ensure a higher level of security.
The `root` account on this computer has been disabled by default for security reasons**.**

```
login as: moxa
moxa@192.168.27.42's password:

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

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 the 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.

```
moxa@Moxa:~# sudo ifconfig eth0 192.168.100.100
[sudo] password for moxa:
moxa@Moxa:~$ sudo ifconfig eth1
[sudo] password for moxa:
eth1      Link encap:Ethernet  HWaddr 00:90:e8:00:df:fe
          inet addr:192.168.100.100  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0  (0.0 B)  TX bytes:0  (0.0 B)
          Interrupt:41 Base address:0xe000
moxa@Moxa:~$
```

In addition, use **sudo –i** to login as root to have more privileges.

```
moxa@Moxa:~# sudo –i
[sudo] password for moxa:
root@Moxa:~$
```

Setting up Desktop Environments

In this section we discuss how to setup various desktop environments on the MC-1100 computer. By default, the MC-1100 Linux operating system doesn't install a desktop environment. However, Debian supports various full-fledged graphical environments such as Gnome, KDE as well as lighter environments like Xfce and LXDE. Users can install one of these desktop systems on the MC-1100.
To install Gnome use the following command:

```
moxa@Moxa:~# sudo apt-get install task-gnome-desktop
```

To install KDE use the following commands:

```
moxa@Moxa:~# sudo apt-get install aptitude tasksel
moxa@Moxa:~# sudo aptitude install ~t^desktop$ ~t^kde-desktop$
```

To install Xfce use the following command:

```
moxa@Moxa:~# sudo apt-get install xfce4 xfce4-goodies task-xfce-desktop
```

To install the minimum LXDE environment use the following command:

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

Starting from a VGA Console

Connect the display monitor to the MC-1100 and power it up connecting 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

Login: **moxa**; **Password:** **moxa**

```
Moxa login: moxa
Password:

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

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

Connecting from an SSH Console

You can use an SSH console on your workstation/PC to connect to the MC-1100 computers. These connections provide users better network security as compared to Telnet connections. The default IP addresses and netmasks of the network interfaces are as follows:

	Default IP Address	Netmask
LAN1	192.168.3.127	255.255.255.0
LAN2	192.168.4.127	255.255.255.0
LAN3	192.168.5.127	255.255.255.0
LAN4	192.168.6.127	255.255.255.0
WLAN	192.168.7.127	255.255.255.0

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

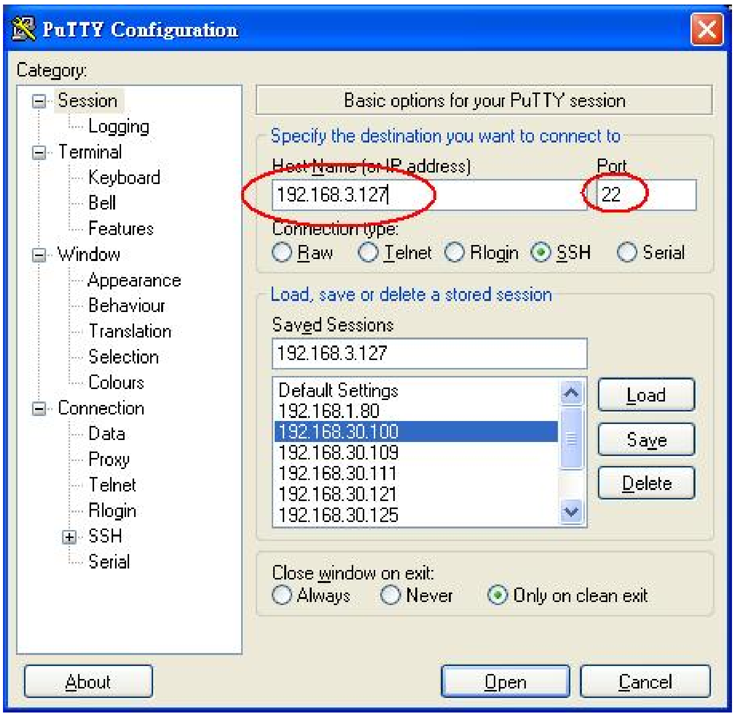
Use a cross-over Ethernet cable to connect your development workstation directly to the MC-1100, or use a straight-through Ethernet cable to connect the MC-1100 to a LAN hub or switch. Use an SSH client installed on your development workstation to connect to the MC-1100. After a connection has been established, type the login and password as requested to log on to the MC-1100. The default values are both **moxa**.

Login: **moxa**
Password: **moxa**

Windows Users

To set up an SSH console for the MC-1100 in a Windows environment, use the **PuTTY** application available for free at: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

The following screenshot shows an example of the configuration that is required in **PuTTY**:



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.3.127
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: 192.168.3.127

Default Settings
192.168.1.80
192.168.30.100
192.168.30.109
192.168.30.111
192.168.30.121
192.168.30.125

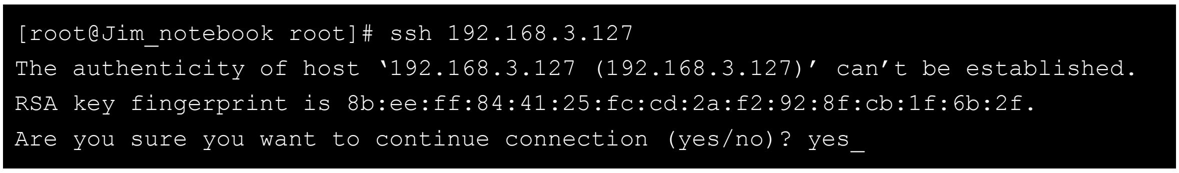
Close window on exit: ○ Always ○ Never ● Only on clean exit

Linux Users

To access the MC-1100's console utility via SSH from a Linux machine, use the **ssh** command as follows:

```
#ssh moxa@192.168.3.127
```

Type **yes** to open the connection.



```
[root@Jim_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 MC-1100 has two time settings: The system time, and the time of the RTC (Real Time Clock) built into the MC-1100 hardware.

Setting the Time Manually

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

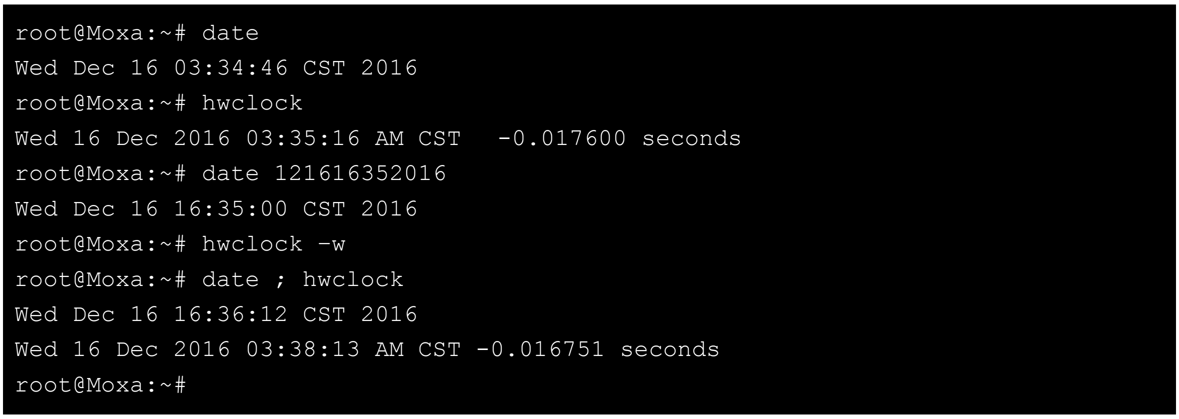
```
# date MMDDhhmmYYYY
```

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

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

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 2016
root@Moxa:~# hwclock
Wed 16 Dec 2016 03:35:16 AM CST   -0.017600 seconds
root@Moxa:~# date 121616352016
Wed Dec 16 16:35:00 CST 2016
root@Moxa:~# hwclock -w
root@Moxa:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2016
Wed 16 Dec 2016 03:38:13 AM CST -0.016751 seconds
root@Moxa:~#
```

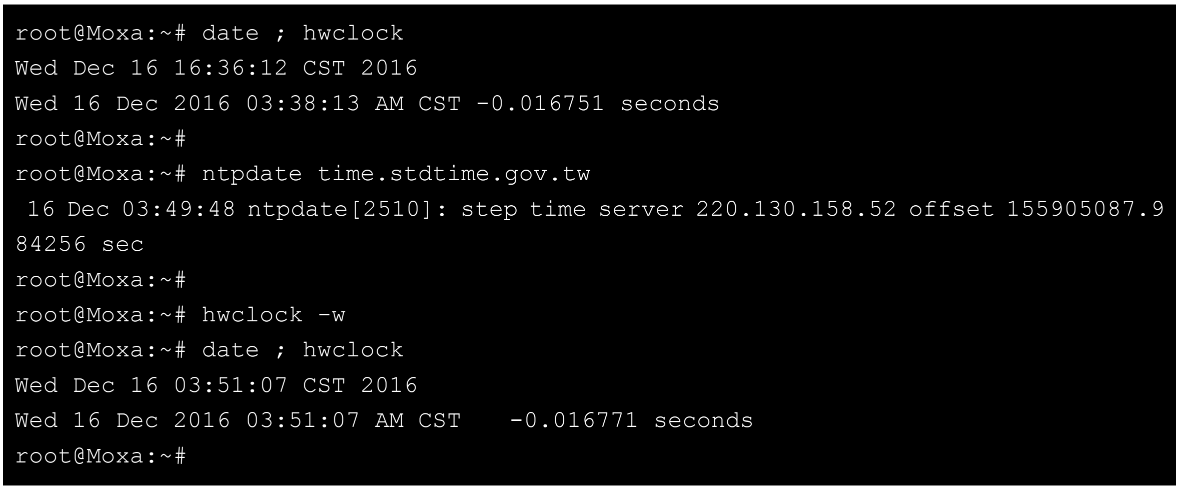
NTP Client

The MC-1100 can use a NTP (Network Time Protocol) client to initialize a time request to a remote NTP server, which you can install using the following command:

```
# apt-get install ntpdate
```

Use the **ntpdate** command to update the system time

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



```
root@Moxa:~# date ; hwclock
Wed Dec 16 16:36:12 CST 2016
Wed 16 Dec 2016 03:38:13 AM 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 2016
Wed 16 Dec 2016 03:51:07 AM CST   -0.016771 seconds
root@Moxa:~#
```

For more information on NTP and NTP server addresses, visit http://www.ntp.org.

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 scripts to automatically update the system time.

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.
```

You can save this shell script using any file name. Let's call it **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 the following command to change the shell script mode:

```
# chmod 755 fixtime
```

Use the **vi** editor to edit the **/etc/inittab** file as follows:

```
# vi /etc/inittab
```

At the end of the script add the following line:

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

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

```
# init q
```

Enabling and Disabling Daemons

By default, the following daemons are enabled in the MC-1100:

sftpd	SFTP Server / Client daemon
sshd	Secure Shell Server daemon

You can manage the services that you want to run in the background using the **insserv** command. The following example shows how to add the apache daemon to the current run level:

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

The apache daemon will not be activated in the current boot session, but will run in the background starting with 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 the **/etc/init.d** file.

To start the apache daemon in the current boot session, use:

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

To stop the apache daemon, use:

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

You can use the management utility called **systemctl** to list all services running at any time as follows:

```
moxa@Moxa:~$ sudo systemctl
```

The **inetd** daemon is another service that you can activate in the current boot session.

To enable the **inetd** service, use the following command.

```
moxa@Moxa:~$ sudo systemctl enable inetd
```

To disable the **inetd** service, use the following command:

```
moxa@Moxa:~$ sudo systemctl disable inetd
```

Linux daemons can also be started or stopped using the **systemctl** command.

To start the **inetd** daemon, use:

```
moxa@Moxa:~$ sudo systemctl start inetd
```

To stop the **inetd** service, use:

```
moxa@Moxa:~$ sudo systemctl stop inetd
```

To restart the **inetd** service, use:

```
moxa@Moxa:~$ sudo systemctl restart inetd
```

To show the status of **inetd** service, use:

```
moxa@Moxa:~$ sudo systemctl status inetd
```

To run a private daemon, you can edit the file **rc.local**, as shown below:

1. Type cd /etc/ to change the directory.

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

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

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

3. Add the application daemon that you want to run to the **rc.local** file.
   Here we use the example of the **tcps2-release** program, which you can find on the CD, to illustrate how to configure and run a service as a background process.

```
# !/bin/sh
# Add the daemon that you want to run
/root/tcps2-release &~
```

4. After rebooting the system, you can find the daemon that you enabled by running the **ps** command.

```
root@Moxa:~# ps -ef
    PID  Uid        VmSize Stat  Command
      1 root          1296 S     init
      2 root               S     [keventd]
      3 root               S     [ksoftirqd_CPU0]
      4 root               S     [kswapd]
      5 root               S     [bdflush]
      6 root               S     [kupdated]
      7 root               S     [mtdblockd]
      8 root               S     [khubd]
     10 root               S     [jffs2_gcd_mtd3]
     32 root               D     [ixp425_csr]
     38 root          1256 S     stdef
     47 root          1368 S     /usr/sbin/inetd
     53 root          4464 S     /usr/sbin/httpd
     63 nobody        4480 S     /usr/sbin/httpd
     64 nobody        4480 S     /usr/sbin/httpd
     65 nobody        4480 S     /usr/sbin/httpd
     66 nobody        4480 S     /usr/sbin/httpd
     67 nobody        4480 S     /usr/sbin/httpd
     92 bin           1460 S     /sbin/portmap
     97 root          1264 S     /root/tcps2-release
    105 root          1556 S     /usr/sbin/rpc.statd
    109 root          4044 S     /usr/sbin/snmpd -s -l   /dev/null
    111 root          2832 S     /usr/sbin/snmptrapd -s
    140 root          1364 S     /sbin/cardmgr
    144 root          1756 S     /usr/sbin/rpc.nfsd
    146 root          1780 S     /usr/sbin/rpc.mountd
    153 root          2960 S     /usr/sbin/sshd
    161 root          1272 S     /bin/reportip
    162 root          3464 S     /bin/massupfirm
    163 root          1532 S     /sbin/getty 115200 ttyS0
    164 root          1532 S     /sbin/getty 115200 ttyS1
    166 root          3464 S     /bin/massupfirm
    168 root          3464 S     /bin/massupfirm
    171 root          3652 S     /usr/sbin/sshd
    172 root          2200 S     -bash
    174 root          1592 S     ps -ef
root@Moxa:~#
```

Executing Scheduled Commands Using the Cron Daemon

The **cron** deamon is used to run scheduled tasks in a Linux environment. When the **cron** daemon is running in the background, it searches the **/etc/crontab** file, **every minute,** to check each command to see if it should be run at that time. The output of the commands that are run is sent to the owner of the **crontab** or to the user named in the MAILTO environment variable in the **crontab**, if such a user exists.

To set up your scheduled applications, you must modify the **/etc/crontab** file**.** The **crontab** entries 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, add an entry to the **crontab** file in the format specified below 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 the **cron** service to update the system time and RTC time at 8:00 AM every day:

1. Write the following shell script and save it as fixtime.sh in the /home/.folder:

   ```
   #!/bin/sh
   ntpdate time.stdtime.gov.tw
   hwclock –w
   exit 0
   ```

2. Change the mode of **fixtime.sh**

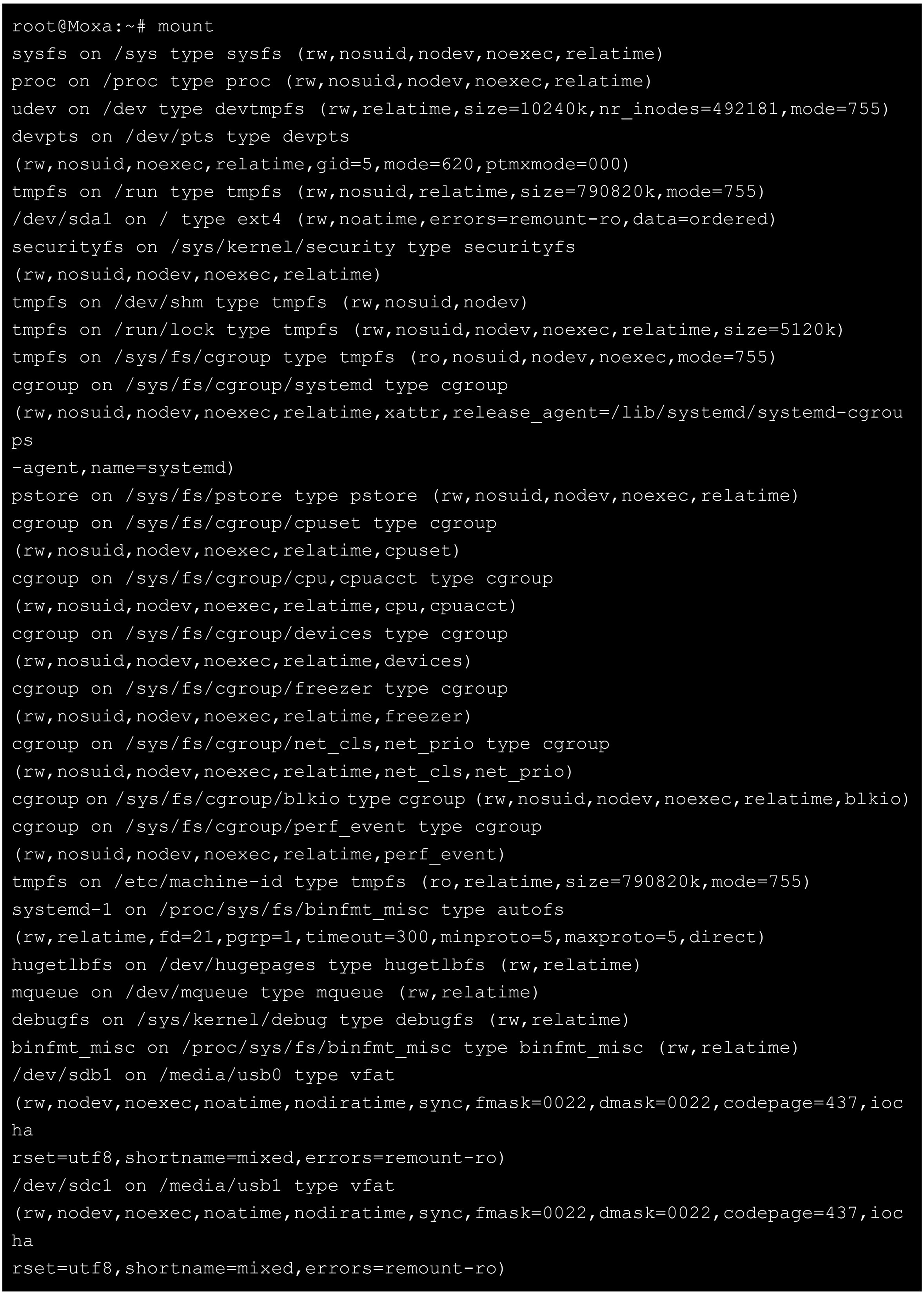
   ```
   # chmod 755 fixtime.sh
   ```

3. Modify the /etc/crontab file to run **fixtime.sh** at 8:00 every day by adding the following line to the end of **crontab** content:

   ```
   * 8 * * *  root    /home/fixtime.sh
   ```

Inserting a USB Storage Device into the Computer

By default, the USB storage devices on the MC-1100 will not be mounted automatically. You have to mount the storage devices manually. Since mounting USB storage devices manually can be difficult, a Debian package called **usbmount** is available to mount the USB drivers automatically. The **usbmount** package relies on the **udev** package to mount the USB storage devices automatically on certain mounting points such as /media/usb0 and /media/usb1.



```
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=492181,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,relatime,size=790820k,mode=755)
/dev/sda1 on / type ext4 (rw,noatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs
(rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup
(rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgrou
ps
-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/cpuset type cgroup
(rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup
(rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/devices type cgroup
(rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup
(rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup
(rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/perf_event type cgroup
(rw,nosuid,nodev,noexec,relatime,perf_event)
tmpfs on /etc/machine-id type tmpfs (ro,relatime,size=790820k,mode=755)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=21,pgrp=1,timeout=300,minproto=5,maxproto=5,direct)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,relatime)
/dev/sdb1 on /media/usb0 type vfat
(rw,nodev,noexec,noatime,nodiratime,sync,fmask=0022,dmask=0022,codepage=437,ioc
ha
rset=utf8,shortname=mixed,errors=remount-ro)
/dev/sdc1 on /media/usb1 type vfat
(rw,nodev,noexec,noatime,nodiratime,sync,fmask=0022,dmask=0022,codepage=437,ioc
ha
rset=utf8,shortname=mixed,errors=remount-ro)
```

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.

Checking the Linux Version

The **uname** command, short for UNIX name, prints the name, version, and other details of the operating system running on the current computer. Use the **-a** option to see the information in the following format:

```
moxa@Moxa:~# uname -a
Linux Moxa 4.6.0-0.bpo.1-amd64 #1 SMP Debian 4.6.4-1~bpo8+1 (2016-08-11) x86_64
GNU/Linux
```

Installing and Removing Packages Using APT

The advance package tool (APT) is a Debian tool used to install and remove packages. Before you use this tool to install a package, you need to configure the apt source file**.**

1. Use the vi editor to configure the apt source file **/etc/apt/sources.list** as follows:

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

deb http://ftp.us.debian.org/debian/ jessie main
deb-src http://ftp.us.debian.org/debian/ jessie main

deb http://security.debian.org/ jessie/updates main
deb-src http://security.debian.org/ jessie/updates main

# jessie-updates, previously known as 'volatile'
deb http://ftp.us.debian.org/debian/ jessie-updates main
deb-src http://ftp.us.debian.org/debian/ jessie-updates main

deb http://ftp.debian.org/debian jessie-backports main
deb-src http://ftp.debian.org/debian jessie-backports main

# Moxa's update
deb http://debian.moxa.com/debian jessie main
#deb http://220.135.161.42/debian jessie main
```

2. Add Moxa's apt repository to /etc/apt/sources.list

To enable you to add or update the drivers, libraries, and utility packages provided by Moxa, we have already included the **deb http://debian.moxa.com/debian jessie main** command in the **source. list** by default. If it is deleted or not available for some reason, you should add these to the source list before you add Moxa's **apt** repository.

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

ATTENTION

Moxa has encrypted its packages with a GPG key, which allows you to check if a package is verified by us. Hence, you must first upload the GPG key to the Moxa embedded computer and add it to the GPG key list before installing the packages. You can find the GPG key in the CD/DVD or on the Moxa website.

You can check if the GPG key is already uploaded to your computer by checking the key list as follows:

```
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]
```

The GPG key will be displayed using **`MOXA SYS`**. If you do not find the GPG key, you can add it to the GPG key list as follows:

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

You must then update the package list as follows:

```
root@Moxa:~# apt-get update
Ign http://debian.moxa.com jessie Release.gpg
Ign http://debian.moxa.com/debian/ jessie/main Translation-en
Ign http://debian.moxa.com/debian/ jessie/main Translation-en_HK
Get:1 http://debian.moxa.com jessie Release [1,633 B]
Ign http://debian.moxa.com jessie/main i386 Packages
Get:2 http://debian.moxa.com jessie/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, you can use the **apt-get** command to install or upgrade the packages from Moxa's **`apt`** repository as follows:

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 (for example: **ipsec-tools**), type:

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

Use one of the following commands to remove a package:

For a simple package removal:

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

For a complete package removal:

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

ATTENTION

You can free up the cache space using the **`# apt-get clean`** command.

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

Power Management—Device Suspend

The MC-1100 supports the power management function ACPI S3 (suspend to RAM). To enable this support, you must first enable the S3 option in the BIOS and then use the following command:

```
Moxa:~# pm-suspend --quirk-s3-bios
```

Once the **suspend** command is in effect, you can use the power button to wake up the computer.

In X Windows, an administrator (root) user can use the **System** → **Shutdown** → **Suspend** function to suspend a device. This function is not available to non-root users.

When a Moxa embedded computer comes out of a suspended state, some application components may need to be reset. You can write a simple script in the **/usr/lib/pm-utils/sleep.d/** folder to perform this procedure as demonstrated in the following example, **script 99serial:**

```
#!/bin/sh

case “$1” in
      hibernate|suspend)
             echo “close AP and tty ports which are opened”
             echo “operations before serial ports suspend”
             ;;
      thaw|resume)
             echo “restart AP”
             echo “operations after serial ports resume”
             ;;
      *) exit $NA
             ;;
esac
```

NOTE To run this script, start **rsyslogd** using the command **/etc/init.d/rsyslogd start** and then view the file **/var/log/pm-suspend.log**.

Wake On LAN

The MC-1100 supports wake on LAN (WOL), a feature used to wake up a device for suspend (S3) and shutdown (S5) states.

To check the WOL support on Ethernet port x, type **ethtool** eth*x*, where “x” is the port number.

```
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: g
        Current message level: 0x00000033 (51)
        Link detected: yes
Moxa:/#
```

The default WOL support parameter is "**g**" (wake on Magic packet).

If the WOL setting is not **g**, we suggest that you modify the setting using the **ethtool -s eth*x* wol g** command.

The following example illustrates how to suspend your computer and how to wake it up using another computer (wake up on suspend, S3):

1. Moxa embedded computer
   i. Enable the **S3** options in the BIOS
   ii. Get the MAC address using the **ifconfig eth*x*** (x is the port number) command
   iii. Suspend to RAM using the **pm-suspend --quirk-s3-bios** command
2. On a remote computer

   Run the **etherwake -b** *mac_of_the_Moxa_computer* command to wake the Moxa embedded computer. For example:

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

The following example illustrates how to shut down your computer and how to wake it up using another computer (wake up on shutdown, S5):

1. Moxa embedded computer

   Shut down your computer using the **shutdown -h now** command
2. On a remote computer

   Run the **etherwake -b *mac_of_the_Moxa_computer*** command to wake the Moxa embedded computer. For example:

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

3

Managing Communications

The MC-1100 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:

- ❒ **Renaming the Network Interfaces**
- ❒ **Changing the Network Settings**
  - ➢ Editing the Interfaces Configuration File
  - ➢ Adjusting IP Addresses with the ifconfig Command
- ❒ **DNS Client**
  - ➢ /etc/hostname
  - ➢ /etc/resolv.conf
  - ➢ /etc/nsswitch.conf
- ❒ **Configuring Ethernet Bonding**
- ❒ **Apache Web Server**
  - ➢ Installing the 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 Hardwired Link
  - ➢ Checking the Connection
  - ➢ Setting Up a Machine for Incoming PPP Connections
- ❒ **PPPoE**
- ❒ **NFS (Network File System) Client**
- ❒ **SNMP (Simple Network Management Protocol)**
- ❒ **OpenVPN**
  - ➢ Installing OpenVPN
  - ➢ Ethernet Bridging for Private Networks on Different Subnets
  - ➢ Ethernet Bridging for Private Networks on the Same Subnet
  - ➢ Routed IP
- ❒ **Wireless Management**
  - ➢ iw
  - ➢ wpa_supplicant
- ➢ **Hostapd**

Renaming the Network Interfaces

You can use the **udevd** command on Linux-based systems to detect new network interfaces, including Ethernet and wireless interfaces. The default order of the network interfaces is not the same as the order of the interfaces as labelled on the product. The udev rule, **/etc/udev/rules.d/01-rename_net_interface.rules**, can be used to reorder the network interface. The content of the file is similar to the following example:

```
KERNEL=="eth*", KERNELS=="0000:01:00.0", NAME="eth0"
KERNEL=="eth*", KERNELS=="0000:02:00.0", NAME="eth1"
KERNEL=="eth*", KERNELS=="0000:06:00.0", NAME="eth2"
KERNEL=="eth*", KERNELS=="0000:07:00.0", NAME="eth3"
```

Changing the Network Settings

The MC-1100 computer has four 10/100/1000 Ethernet ports. 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
LAN3	192.168.5.127	255.255.255.0
LAN4	192.168.6.127	255.255.255.0
WLAN	192.168.7.127	255.255.255.0

These network settings can be modified by editing the parameters in the **interfaces** configuration file. You can also temporarily adjust these settings using the **ifconfig** command.

Editing the Interfaces Configuration File

1. Type cd /etc/network to change the path to network folder.

```
Moxa:~# cd /etc/network
```

2. Type **vi interfaces** to edit the network configuration file using the **vi** editor.

   You can then modify the network parameter in the **interfaces** file.

```
Moxa:/etc/network# vi interfaces
```

For example, you can configure the MC-1100's Ethernet ports for static or dynamic (DHCP) IP addresses.

Configuring a Static IP Address

The default static IP addresses can be modified as shown in the following example:

```
# 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

auto eth2
iface eth2 inet static
       address 192.168.5.127
       netmask 255.255.255.0
       broadcast 192.168.5.255

auto eth3
iface eth3 inet static
       address 192.168.6.127
       netmask 255.255.255.0
       broadcast 192.168.6.255

auto wlan0
iface wlan0 inet static
       address 192.168.7.127
       netmask 255.255.255.0
       broadcast 192.168.7.255
```

Configuring 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

```
Moxa:~# /etc/init.d/networking restart
```

Adjusting IP Addresses with the ifconfig Command

IP settings can be adjusted during run-time, but the new settings must be saved to the disk by modifying the **/etc/network/interfaces** file and saving the changes. For example, type the command **# ifconfig eth0 192.168.1.1** to change the IP address of LAN1 to 192.168.1.1.

```
Moxa:~# ifconfig eth0 192.168.1.1
Moxa:~#
```

DNS Client

The MC-1100 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.

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

3. Check the new hostname.

```
moxa@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** command 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 the /etc/resolv.conf (assuming the DNS server’s IP address is 8.8.8.8): **nameserver 8.8.8.8**

```
Moxa:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
#nameserver 192.168.1.16
nameserver 8.8.8.8
nameserver 8.8.4.4
nameserver 8.8.8.8
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, you have to load the bonding driver with mode setting. Then use the **ifenslave** command to add the Ethernet interface into the bond0 interface. Here is the script that bonds the eth1 and eth2 interfaces together, which you can add to the **/etc/init.d/bonding.sh** file

```
#! /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 eth2                 # putting down eth2
    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 eth2        # set eth2 in slave for bond0
    ifenslave bond0 eth1        # set eth1 in slave for bond0
    ;;

  stop)
    echo "Stopping bonding service: $NAME"
    ifenslave -d bond0 eth2          # release eth2 from bond0
    ifenslave -d bond0 eth1          # release eth1 from bond0

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

    ifup eth2
    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 add this to run level by insserv.

```
moxa@Moxa:~# sudo insserv -v -d bonding.sh
```

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

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

Apache Web Server

Installing the Apache Web Server

The Apache web server is installed on the MC-1100 computer but not enabled. You can enable the server using the **systemctl** command as follows:

```
Moxa:~# sudo systemctl enable apache2
```

Manually start the apache2 server using the following command:

```
Moxa:~# sudo systemctl start apache2
```

Default Homepage

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

Save your own homepage to the directory **/var/www**

Save your CGI page to the directory **/var/www**

Before you modify the homepage, use a browser (such as Microsoft Internet Explorer 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/**

To test the default CGI page, type:**http://192.168.3.127/cgi-bin/w3mmail.cgi**

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 MC-1100 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.

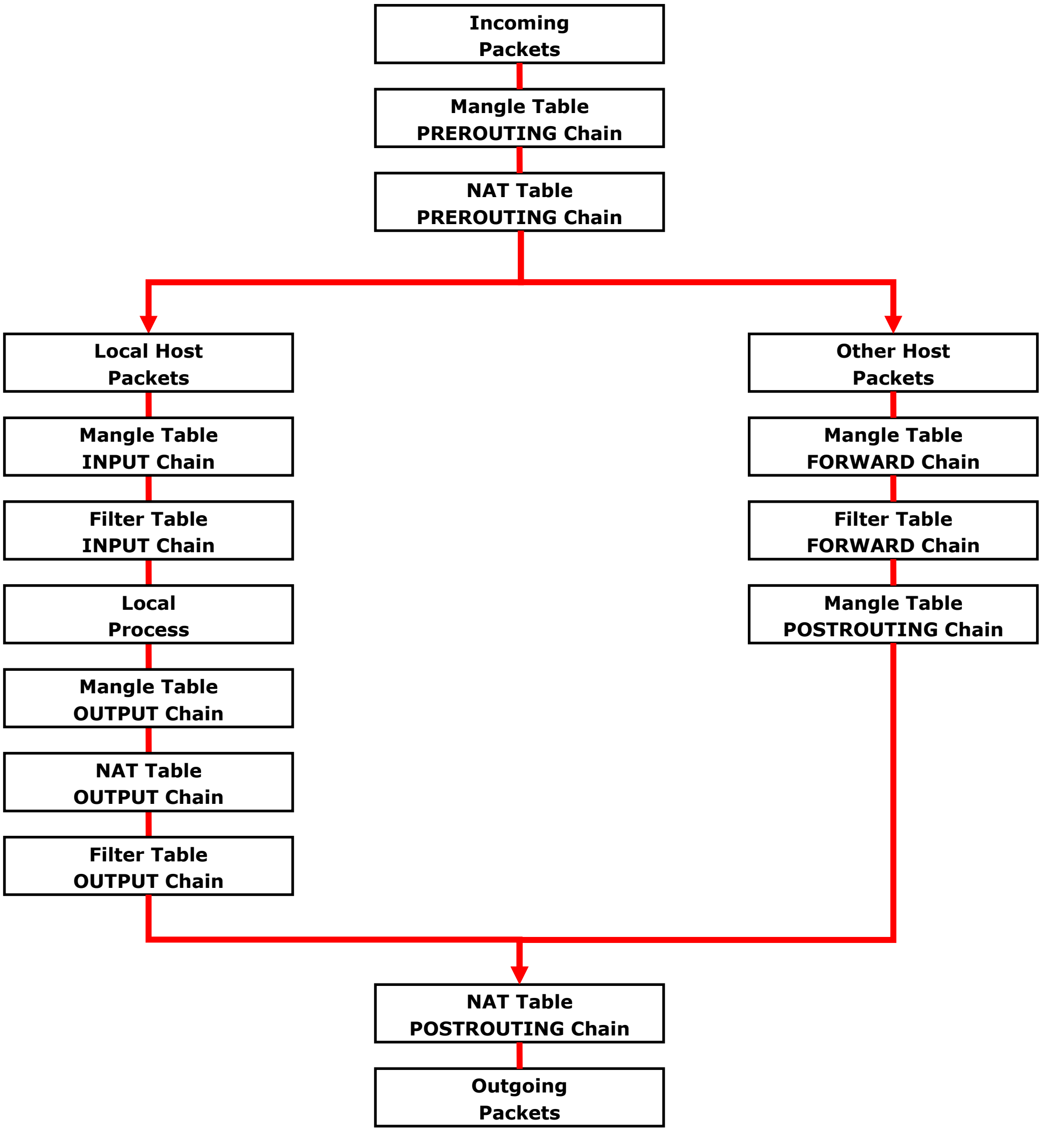
Mangle Table—Includes Two Chains

- **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.


Incoming Packets
Mangle Table PREROUTING Chain
NAT Table PREROUTING Chain
Local Host Packets
Mangle Table INPUT Chain
Filter Table INPUT Chain
Local Process
Mangle Table OUTPUT Chain
NAT Table OUTPUT Chain
Filter Table OUTPUT Chain
Other Host Packets
Mangle Table FORWARD Chain
Filter Table FORWARD Chain
Mangle Table POSTROUTING Chain
NAT Table POSTROUTING Chain
Outgoing Packets

IPTables Modules

IPTables supports the following sub-modules. Be sure to 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 the **lsmod** command to check if the **ip_tables** module has already been loaded in the MC-1100. Use **modprobe** to insert and enable the module.

Use the **iptables**, **iptables-restore**, and **iptables-save** commands 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 MC-1100.
OUTPUT: For locally-generated packets.
FORWARD: For packets routed out through the MC-1100.
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 MC-1100’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 MC-1100’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)

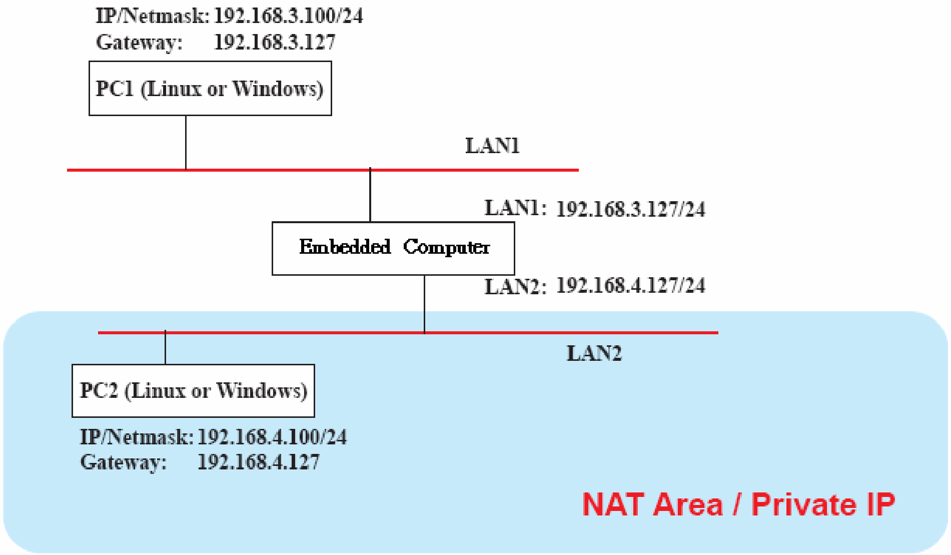
The NAT (Network Address Translation) protocol translates IP addresses used on one network into IP addresses used on a connecting network. One network is designated the inside network and the other is the outside network. Typically, the MC-1100 connects several devices on a network and maps local inside 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 the following link for more information on 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 MC-1100 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 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 embedded computer Ethernet port. Since PPP is a peer-to-peer system, the Linux operating system can 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 “”login: username password: password’
/dev/ ttyS0 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/
ttyS0 115200 crtscts modem
```

The pppd options are described below:

Option	Description	
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 ...

Option	Description	
	CONNECT	Wait for an answer.
	“ “	Send a return (null text followed by the usual return)
	login: **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 Hardwired Link

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

```
#pppd connect ‘chat –v” “ “ “ ‘ noipdefault /dev/ttyS0 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/ttyS0 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 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:

```
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/ttyS0 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/ttyS0 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-ttyS0.chat”

# Set up routing to go through this PPP link
defaultroute

# Default modem (you better replace this with /dev/ttySx!)
/dev/ttyS0

# 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-ttyS0.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.

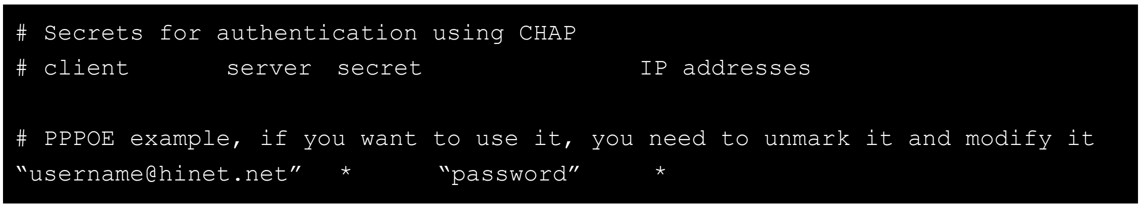
```
Moxa:~# mount –o remount,rw /dev/sda1 /
Moxa:~# echo “p0:2345:respawn:pppd call dialin” >> /etc/inittab
Moxa:~# umount /
```

PPPoE

Use the following procedure to configure PPPoE:

1. Connect the MC-1100’s LAN port to an ADSL modem with a cross-over cable, HUB, or switch.
2. Log in to the MC-1100 as the root user.
3. Edit the file /etc/ppp/chap-secrets and add the following:

“username@hinet.net”   *   “password” *

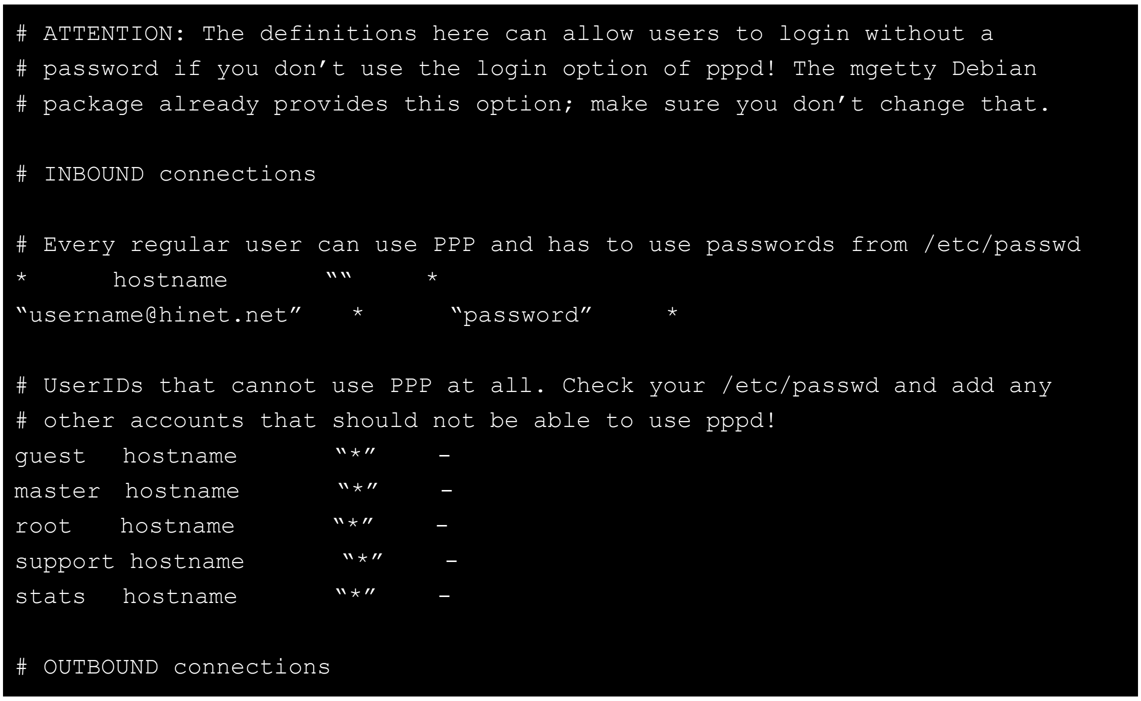


```
# Secrets for authentication using CHAP
# client       server  secret               IP addresses

# 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” *

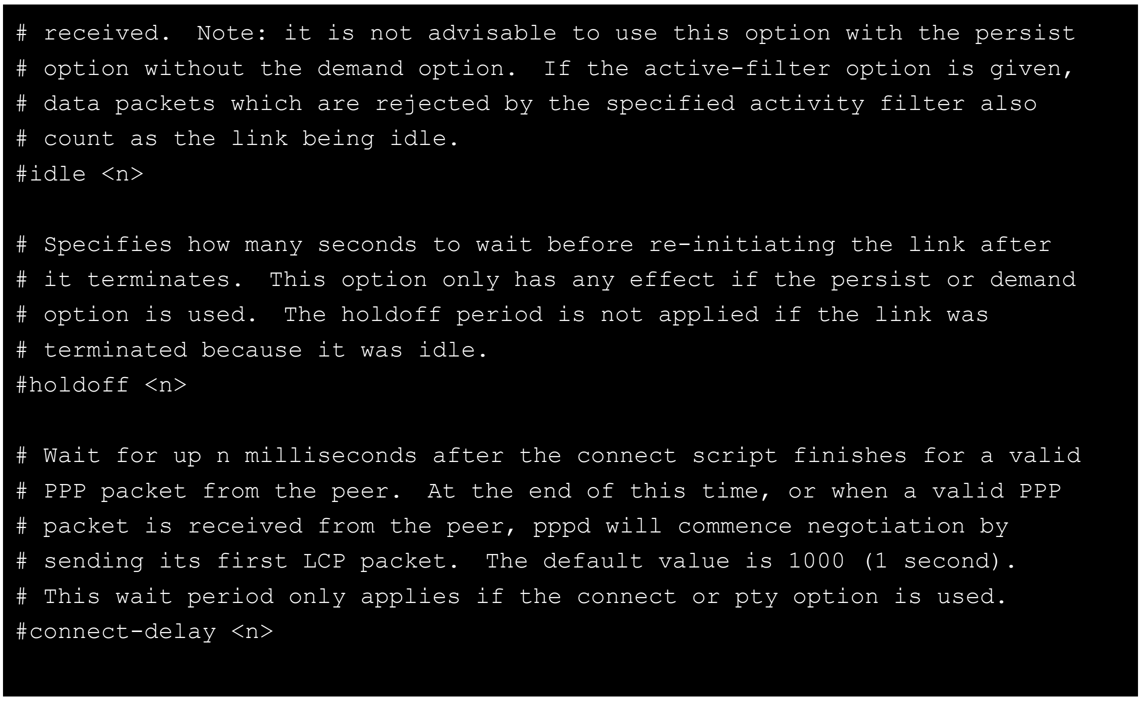


```
# ATTENTION: The definitions here can allow users to login without a
# password if you don’t use the login option of pppd! The mgetty Debian
# package already provides this option; make sure you don’t change that.

# 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 8.8.8.8
   nameserver 8.8.4.4

```
Moxa:/etc# cat resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
nameserver 8.8.8.8
nameserver 8.8.4.4
Moxa:/etc#
```

Use the following command to create a **pppoe** connection:

#pppd eth0

1. The ADSL modem is connected to the LAN1 port, which is named eth0. If the ADSL modem is connected to LAN2, use eth1, etc.
2. 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
```

   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 and seamless sharing of files across a network. NFS allows users to develop applications for the MC-1100 without worrying about the amount of disk space that will be available. The MC-1100 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
   ```

   (192.168.3.100 is the IP address of the NFS server in this example.)

SNMP (Simple Network Management Protocol)

The MC-1100 comes with the SNMP v2c (Simple Network Management Protocol) software package. The snmpd service default is disabled. You can enable it by this command.

```
moxa@Moxa:~# sudo systemctl enable snmpd
```

The snmpd configuration file is located at **/etc/snmp/snmpd.conf**. All network interfaces (LAN ports) can be configured with a snmpd listening address and port number.

```
# /etc/snmp/snmpd.conf
# ...
#  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)
# For security concern, we comment out this line. If you want to support SNMP on all Ethernet Interfaces, please remove the '#' and restart the snmpd service.
# agentAddress udp:161,udp6:[::1]:161
```

Then restart the snmpd service.

```
moxa@Moxa:~# sudo systemctl restart snmpd
```

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

```
iso.3.6.1.2.1.1.1.0 = STRING: "Linux Moxa 4.6.0-0.bpo.1-amd64 #1 SMP Debian
4.6.4-1~bpo8+1 (2016-08-11) x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8691.12.720
iso.3.6.1.2.1.1.3.0 = Timeticks: (163) 0:00:01.63
```

```
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: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
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.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for the
SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
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.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification,
plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (4) 0:00:00.04
iso.3.6.1.2.1.25.1.1.0 = Timeticks: (55237) 0:09:12.37
iso.3.6.1.2.1.25.1.2.0 = Hex-STRING: 07 E0 09 13 03 0B 2F 00 2D 04 00
iso.3.6.1.2.1.25.1.3.0 = INTEGER: 393216
iso.3.6.1.2.1.25.1.4.0 = STRING: "BOOT_IMAGE=/boot/vmlinuz-4.6.0-0.bpo.1-amd64
root=LABEL=root ro initrd=/install/initrd.gz quiet 8250.nr_uarts=2
"
iso.3.6.1.2.1.25.1.5.0 = Gauge32: 2
iso.3.6.1.2.1.25.1.6.0 = Gauge32: 24
iso.3.6.1.2.1.25.1.7.0 = INTEGER: 0
iso.3.6.1.2.1.25.1.7.0 = No more variables left in this MIB View (It is past the
end of the MIB tree)
…
```

OpenVPN

Installing OpenVPN

OpenVPN is an open-source VPN software. You can install it by this command.

```
moxa@Moxa:~$ sudo apt-get install openvpn
```

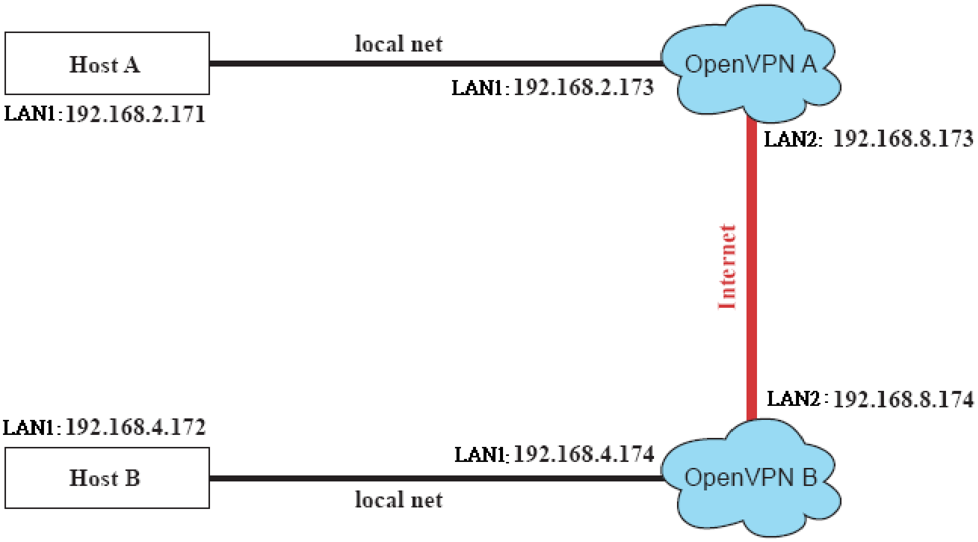
OpenVPN provides two types of tunnels for users to implement VPNS: **Routed IP Tunnels** and **Bridged Ethernet Tunnels**.

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 pre-shared 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. 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---------------------------
```

6. 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”
...
```

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

8. 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
```

9. 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 -----------------------------
```

10. 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”
...
```

11. 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
```

12. 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
```

13. 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.

14. 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
```

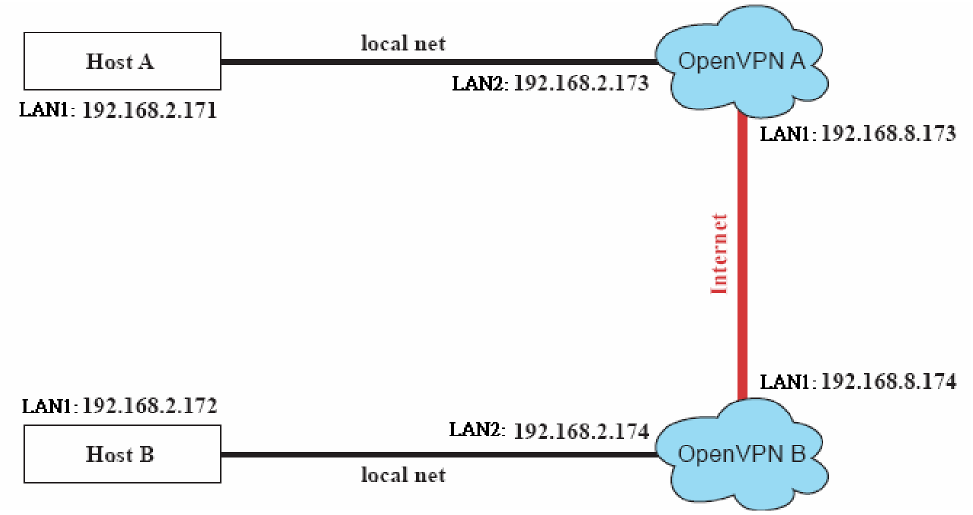
A successful ping indicates that you have created a VPN system that only allows authorized users from one internal network to access users at the remote site. For this system, all data is transmitted by UDP packets on port 5000 between OpenVPN peers.

15. 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:

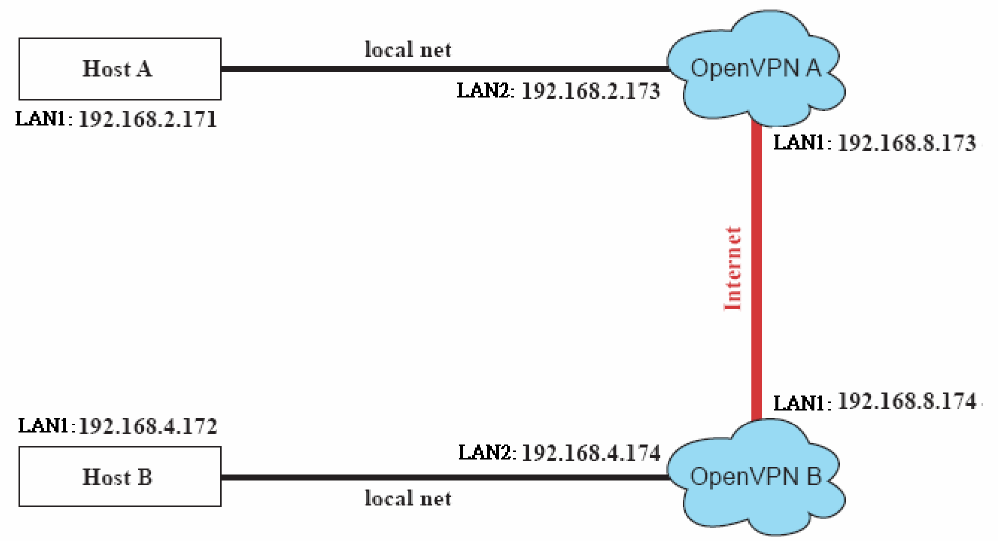


2. The configuration procedure is almost the same as for the previous example. The only difference is that you will need to comment out the parameter **up** in **/etc/openvpn/tap0-br.conf** of OpenVPN A and **/etc/openvpn/tap0-br.conf** of OpenVPN B.

```
# 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. 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 the 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
```

5. 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.

6. After you run the OpenVPN programs, check the routing table by typing the **# route** command.

```
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
```

Wireless Management

MC-1121 has a mini-PCIe slot for wirelesss module installation. Currently we support the WPEA252NI module in the Linux system. The wireless module is manageable by **`iw`**, **`wpa_supplicant`**, or **`hostapd`** software package in the system.

iw

iw is a new nl80211 based CLI configuration utility for wireless devices. It replaces the old tool iwconfig, which uses Wireless Extensions interface, is deprecated and recommended to switch to iw and nl80211.

Get device capabilities:

```
moxa@Moxa:~$ sudo iw list
```

Scanning:

```
moxa@Moxa:~$ sudo iw dev wlan0 scan
```

Listening to events:

```
moxa@Moxa:~$ sudo iw event
```

Establishing a basic connection to an AP that has encryption disabled:

```
moxa@Moxa:~$ sudo iw dev wlan0 connect TARGET_AP_SSID
```

To connect to an AP that uses WEP, you can use:

```
moxa@Moxa:~$ sudo iw dev wlan0 connect TARGET_AP_SSID key 0:abcde d:1:6061626364
```

Disconnect from the current network:

```
moxa@Moxa:~$ sudo iw dev wlan0 disconnect
```

Get the usage information of the iw utility:

```
moxa@Moxa:~$ sudo iw –help
```

wpa_supplicant

Create /etc/wpa_supplicant/wpa_supplicant.conf for connecting to an AP that has encryption disabled.

```
        moxa@Moxa:~$ sudo vi /etc/wpa_supplicant/wpa_supplicant.conf

        network={

        ssid="MYAPSSID"
        key_mgmt=NONE

        }
```

To connect to an AP that uses WEP, use the following:

```
moxa@Moxa:~$ sudo vi /etc/wpa_supplicant/wpa_supplicant.conf

network={

        ssid="MYAPSSID"
        key_mgmt=NONE
        wep_key0=1234567890
        wep_key1="abcde"
        wep_key2="1234567890123"
        wep_tx_keyidx=0
        priority=5

        }
```

To connect to an AP that use WPA, use the following:

```
moxa@Moxa:~$ sudo vi /etc/wpa_supplicant/wpa_supplicant.conf

network={

        ssid="MYAPSSID"
        psk=”1234567890”
        }
```

Manually run the wpa_supplicant in daemon mode with the above configures.

```
moxa@Moxa:~$ sudo wpa_supplicant –Dnl80211 -iwlan0 -c
/etc/wpa_supplicant/wpa_supplicant.conf –B
```

You can then check the connection using the **`iw`** command.

```
moxa@Moxa:~$ sudo iw dev wlan0 link
```

After you have configured the settings in the /etc/wpa_supplicant/wpa_supplicant.conf file, edit /etc/network/interfaces to use wpa_supplicant.

```
moxa@Moxa:~$ sudo vi /etc/network/interfaces

...

auto wlan0
iface wlan0 inet dhcp
```

```
       wpa-driver nl80211
       wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Then you can use the **ifdown wlan0** and **ifup wlano** commands to check the wpa_supplicant configuration file.

Hostapd

The **hostapd** is a user space daemon for access point and authentication server. It is not installed by default on the computer. If you want to support the AP mode in your computer, you must configure the **hostapd** daemon as follows:

1. Install the hostapd package using the following command:

```
moxa@Moxa:~$ sudo apt-get install hostapd
```

2. Setup the wlan0 interface to support the AP mode.
3. Edit the **/etc/default/hostapd** file to configure hostapd.
4. Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration file.

```
moxa@Moxa:~$ sudo vi /etc/default/hostapd

...

DAEMON_CONF=”/etc/hostapd/hostapd.conf”
```

5. Create the **/etc/hostapd/hostapd.conf** file as follows:

```
moxa@Moxa:~$ sudo vi /etc/hostapd/hostapd.conf

# The Wireless interface name
interface=wlan0

# Set the bridge name
bridge=br0

# Set the driver name
driver=nl80211

# Country name code in ISO/IEC 3166-1 format. This is used to set regulator domain.
# This can limit available channels and transmit power
# (IN: INDIA, UK:United Kingdom, US: United States and so on
country_code=US

# Set the SSID
ssid=TestAP

# Set the operation mode
hw_mode=g

# Set the channel number
channel=6

# Set wpa mode to 2
wpa=2

# Set the passphrase
```

```
wpa_passphrase=0123456789

# Set key and authentication for WPA2
wpa_key_mgmt=WPA-PSK WPA-EAP

# Set of accepted cipher suites (encryption algorithms) for pairwise keys
# CCMP = AES in Counter mode with CBC-MAC [RFC 3610, IEEE 802.11i/D7.0]
# TKIP = Temporal Key Integrity Protocol [IEEE 802.11i/D7.0]
wpa_pairwise=TKIP CCMP
# Pairwise cipher for RSN/WPA2 (default: use wpa_pairwise value)
rsn_pairwise=CCMP

# IEEE 802.11 specifies two authentication algorithms.
# bit 0 = Open System Authentication
# bit 1 = Shared Key Authentication (requires WEP)
auth_algs=3

# Station MAC address - accept unless in deny list
macaddr_acl=0
```

6. Test the AP configuration by manually starting the **hostapd** service

```
moxa@Moxa:~$ sudo hostapd –d /etc/hostapd/hostapd.conf
```

or restart using **`/etc/init.d/hostpad`**

```
moxa@Moxa:~$ sudo /etc/init.d/hostapd restart
```

7. Configure /etc/network/interfaces to set br0, wlan0 and eth0, in bridge mode.

```
moxa@Moxa:~$ sudo vi /etc/network/interfaces
...
auto br0
iface br0 inet static
      bridge_ports wlan0 eth0
      address 192.168.8.127
      netmask 255.255.255.0
      network 192.168.8.0
      gateway 192.168.8.254
      dns-nameservers 192.168.8.254
```

8. Restart networking and hostapd services to test the configuration.

```
moxa@Moxa:~$ sudo /etc/init.d/networking restart
moxa@Moxa:~$ sudo /etc/init.d/hostapd restart
```

NOTE Since the AP is configured in bridge mode, DHCPD is not required on the Wi-Fi AP. It's used DHCPD server located on the LAN. If you don't' use the bridge mode, you should install and setup a DHCP server in this system.

For example install the ISC DHCP server.

```
moxa@Moxa:~$ sudo apt-get install isc-dhcp-server
```

Set the isc-dhcp-server default interface in /etc/default/isc-dhcp-server

```
moxa@Moxa:~$ sudo vi /etc/default/isc-dhcp-server
...
INTERFACES=”wlan0”
```

Configure the subnet in /etc/dhcp/dhcpd.conf

```
moxa@Moxa:~$ sudo vi /etc/dhcp/dhcpd.conf
...
## Set a domain name for your LAN ##

option domain-name "example.org";
## Set DNS server IP address, you use Google DNS server##
       option domain-name-servers 192.168.3.2, 192.168.3.3;
...
## Set the lease time
default-lease-time 86400;
...
## Uncommented the authoritative directive for misconfigured clients
authoritative;

## Configure your subnet
subnet 192.168.7.0 netmask 255.255.255.0 {
       ## dhcp start  and end IP range ##
       range 192.168.7.100 192.168.7.200;
       option subnet-mask 255.255.255.0;
       option broadcast-address 192.168.7.255;
       option routers 192.168.7.254;
}
```

IMPORTANT!

The MC-1100 series has two models. Only the MC-1121 supports mini-PCIe slots. The iw, wpa_supplicant and iw are only for MC-1121.

4

Programming Guide

The following topics are covered in this chapter:

- ❒ **Device API**
- ❒ **RTC (Real Time Clock)**
- ❒ **UART**
- ❒ **Digital I/O and Mini-PCIe Power Control**
- ❒ **WDT (Watch Dog Timer)**
  - ➢ Introduction
  - ➢ Watchdog Usage
  - ➢ How the WDT Works
  - ➢ The Watchdog Device IOCTL Commands
  - ➢ Examples
- ❒ **Cellular Management—cell_mgmt**

Device API

The MC-1100 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

RTC (Real Time Clock)

The device node is located at **/dev/rtc**. The MC-1100 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.
```

UART

The normal tty device nodes are **/dev/ttyS0** and **/det/ttyS1**. The MC-1100 supports standard Linux termios control with the serial ports. The **setinterface** utility supports to configure the RS-232/422/485 mode. By default, the serial interface is set to RS-232.

setinterface device-node [interface-no]

device-node: /dev/ttySn; n = 0,1,2,...
interface-no: [see following table]:

interface-no	**Operation Mode**
None	Display current setting
0	RS-232
1	2-wire RS-485
2	RS-422
3	4-wire RS-485

For example, use the following commands to set /dev/ttyM0 to RS-422:

```
Moxa:~# setinterface /dev/ttyS0 2
Moxa:~# setinterface /dev/ttyS0
Now setting is RS422 interface.
Moxa:~#
```

The MC-1100 series has two models, MC-1111 and MC-1121 with different numbers of serial port.

Model	Serial Port
MC-1111	N/A
MC-1121	/dev/ttyS0 /dev/ttyS1

NOTE Although the Linux system will create **`/dev/ttyS2`** and **`/dev/ttyS3`** device nodes, it does not associate it with the serial port device. Use the **`setinterface`** command to associate these device nodes to the serial ports.

Digital I/O and Mini-PCIe Power Control

Digital Output channels can be set to high or low. The channels are controlled by the **`/sys`** file. The mapping table for DI, DO, and mini-PCIe power control is given below:

Function	GPIO number	Sysfs file
DI1	329	/sys/class/gpio/di1/value
DI2	330	/sys/class/gpio/di2/value
DI3	331	/sys/class/gpio/di3/value
DI4	332	/sys/class/gpio/di4/value
DO1	333	/sys/class/gpio/do1/value
DO2	334	/sys/class/gpio/do2/value
DO3	335	/sys/class/gpio/do3/value
DO4	336	/sys/class/gpio/do4/value
mini-PCIe	337	/sys/class/gpio/minipcie/value

The DI/DO can be accessed via the **/sys/class/gpio/gpioN/**value. All the GPIO sysfs files are exported by **/etc/systemd/system/multi-user.target.wants/mx_gpio_export.service** at boot sequence. You don't need to export the GPIO entry. You can use echo to control the GPIO directly in a shell program.

To set DO1 status to low, use the following:

```
moxa@Moxa:~# sudo echo 0 > /sys/class/gpio/do1/value
```

To set DO1 to status to high:

```
moxa@Moxa:~# sudo echo 1 > /sys/class/gpio/do1/value
```

To read the DI1 status:

```
moxa@Moxa:~# sudo cat /sys/class/gpio/di1/value
```

To power off/power on the LTE module in the mini-PCIe slot:

```
moxa@Moxa:~# sudo echo "0" > /sys/class/gpio/minipcie/value

<...Wait about 20 seconds for the LTE module ready ... >

moxa@Moxa:~# sudo echo "1" > /sys/class/gpio/minipcie/value
```

To set the value of DO1 in C:

```
sprint(buf, "/sys/class/gpio/do%d /value", gpio);

fd = open(buf, O_WRONLY);

// Set GPIO high status
write(fd, "1", 1);
// Set GPIO low status
write(fd, "0", 1);

close(fd);
```

To set the value of DI1 in C:

```
Sprint(buf, "/sys/class/gpio/di%d /value", gpio);

fd = open(buf, O_RDONLY);

read(fd, &value, 1);

if(value == '0')
{
    // Current GPIO status low
}
else
{
    // Current GPIO status high
}
close(fd);
```

NOTE The MC-1100 series has two models. Only the MC-1121 supports DIO and mini-PCIe slots. Hence, the DIO control function is only for the MC-1121.

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

User can set the acknowledge time from a minimum of 1 sec to a maximum of 1day.The default timer is 60 seconds. So, if the watchdog daemon crashes, the system will reboot after the timeout has passed.

How the WDT Works

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

```
moxa@Moxa:~# sudo insserv -v -d wd_keepalive
insserv: enable service ../init.d/wd_keepalive ->
/etc/init.d/../rc2.d/S01wd_keepalive
insserv: enable service ../init.d/wd_keepalive ->
/etc/init.d/../rc3.d/S01wd_keepalive
insserv: enable service ../init.d/wd_keepalive ->
/etc/init.d/../rc4.d/S01wd_keepalive
insserv: enable service ../init.d/wd_keepalive ->
/etc/init.d/../rc5.d/S01wd_keepalive
insserv: creating .depend.boot
insserv: creating .depend.start
insserv: creating .depend.stop
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:~$
```

The watchdog configuration file is located in **/etc/watchdog.conf**. Currently we configure the watchdog daemon to acknowledge the watchdog device in 60 seconds. Set the **`realtime`** parameter in the configuration file to **`yes`** to lock the watchdog timer into memory so that it is never swapped out to prevent watchdog acknowledge delay. You can configure the **watchdog.conf** file to enable the watchdog as per your system requirement.

```
…
watchdog-device = /dev/watchdog
…
interval             = 60
realtime             = yes
priority             = -10
…
```

If you want to remove the watchdog timer from run-level, use this command:

```
moxa@Moxa:~# sudo insserv -v -r watchdog
insserv: remove service /etc/init.d/../rc0.d/K01watchdog
insserv: remove service /etc/init.d/../rc1.d/K01watchdog
insserv: remove service /etc/init.d/../rc2.d/S05watchdog
insserv: remove service /etc/init.d/../rc3.d/S05watchdog
insserv: remove service /etc/init.d/../rc4.d/S05watchdog
insserv: remove service /etc/init.d/../rc5.d/S05watchdog
insserv: remove service /etc/init.d/../rc6.d/K01watchdog
insserv: creating .depend.boot
insserv: creating .depend.start
insserv: creating .depend.stop
moxa@Moxa:~# sudo insserv -v -r wd_keepalive
insserv: remove service /etc/init.d/../rc2.d/S01wd_keepalive
insserv: remove service /etc/init.d/../rc3.d/S01wd_keepalive
insserv: remove service /etc/init.d/../rc4.d/S01wd_keepalive
insserv: remove service /etc/init.d/../rc5.d/S01wd_keepalive
insserv: creating .depend.boot
insserv: creating .depend.start
insserv: creating .depend.stop
```

The Watchdog Device IOCTL Commands

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

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

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

IOCTL	**WDIOC_SETOPTIONS**
Description	Used to enable or disable the WDT I/O card
Input	None
Output	(int *)arg)
Return	On success, return 0; Otherwise, return 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, return 0. Otherwise, return < 0 value.

IOCTL	**WDIOC_SETTIMEOUT**
Description	Sets the watchdog timeout value
Input	arg: 1 to 60 seconds
Output	None
Return	On success, return 0. Otherwise, return < 0 value.

IOCTL	**WDIOC_GETTIMEOUT**
Description	Gets the current watchdog timeout value
Input	None
Output	arg: 1 to 60 seconds
Return	On success, return 0. Otherwise, return < 0 value.

Examples

The following example **watchdog-simple.c** can be used to send **`acks`** to 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;
}
```

Cellular Management—cell_mgmt

The MC-1100 has a mini-PCIe slot. You can install the LTE mini-PCIe module on this slot. Most of the popular mini-PCIe modules support AT command or QMI interface. We provide a cellular management utility, **`cell_mgmt`**, for cellular module management. Here is the usage menu of this utility.

```
Usage:
/sbin/cell_mgmt [OPTIONS]
OPTIONS
start [APN=[APN],Username=[user],Password=[pass],PIN=[pin_code]]
Start network.
example:
cell_mgmt start
cell_mgmt start APN=internet
cell_mgmt start APN=internet PIN=0000
cell_mgmt start APN=internet Username=moxa Password=pass PIN=0000
stop
Stop network.
restart
Restart network.
reset
Reset cellular.
power_on
Power ON.
power_off
Power OFF.
power_cycle
Power cycle
gps_on
GPS ON.
gps_off
GPS OFF.
status
Query network connection status.
signal
Get signal strength.
set_default
RESET module to factory default.
at ['AT_COMMAND']
Input AT Command.
Must use SINGLE QUOTATION to enclose AT Command.
sim_status
Query sim card status.
set_pin [PIN]
Set PIN code to configuration file.
pin_protection [PIN1|PIN2] [enable|disable] [current_PIN]
Set PIN code to configuration file.
pin_protection [PIN1|PIN2] [enable|disable] [current_PIN]
Set PIN protection in the UIM.
set_apn [APN]
Set APN to configuration file.
check_carrier
Check current carrier.
switch_carrier [Verizon|ATT|Sprint|Generic]
```

```
Switching between US carrier frequency bands.
interface [#slot]
Switching and checking module slot.
m_info
Module information.
operator
Telecommunication operator.
vzwauto
Verizon Private Network auto dialup.
version
Cellular management version.
```

Following are some examples that show how you can use the cellular management utility to manage the cellular module, TE910-EUG.

Getting the signal strength:

```
moxa@Moxa:~# sudo cell_mgmt signal
lte -68 dbm
```

Getting the SIM card status:

```
moxa@Moxa:~# sudo cell_mgmt sim_status
+CPIN: READY
```

Checking module information:

```
moxa@Moxa:~# sudo cell_mgmt m_info
Module=Telit
WWAN_node=wwan0
AT_port=/dev/ttyUSB2
GPS_port=/dev/ttyUSB1
LAC=
CellID=41223949
ICC-ID=89886920041303297151
IMEI=359852051060953
QMI_port=/dev/cdc-wdm0
LCID=
```

Checking cellular operator information:

```
moxa@Moxa:~# sudo cell_mgmt operator
Chunghwa Telecom
```

To power on/off the cellular module:

```
moxa@Moxa:~# sudo cell_mgmt power_off
Network already stopped
Clearing state...
root@Moxa:~# ls -al /dev/cdc-wdm0
ls: cannot access /dev/cdc-wdm0: No such file or directory
root@Moxa:~# sudo cell_mgmt power_on
root@Moxa:~# ls -al /dev/cdc-wdm0
crw------- 1 root root 180, 0 Sep 20 01:56 /dev/cdc-wdm0
```

Checking the **`at`** command:

```
moxa@Moxa:~# sudo cell_mgmt at "at+csq"
+CSQ: 23,99 OK
```

Resetting the cellular module and checking the cellular module information:

```
moxa@Moxa:~# sudo cell_mgmt reset
moxa@Moxa:~# sudo dmesg
...
[ 1740.205317] usb 1-1.3.1: USB disconnect, device number 8
[ 1740.206225] option1 ttyUSB0: GSM modem (1-port) converter now disconnected
from ttyUSB0
[ 1740.206303] option 1-1.3.1:1.0: device disconnected
[ 1740.206930] qmi_wwan 1-1.3.1:1.2 wwan0: unregister 'qmi_wwan'
usb-0000:00:1d.0-1.3.1, WWAN/QMI device
[ 1740.233955] option1 ttyUSB1: GSM modem (1-port) converter now disconnected
from ttyUSB1
[ 1740.233993] option 1-1.3.1:1.3: device disconnected
[ 1740.235131] option1 ttyUSB2: GSM modem (1-port) converter now disconnected
from ttyUSB2
[ 1740.235167] option 1-1.3.1:1.4: device disconnected
[ 1740.237083] option1 ttyUSB3: GSM modem (1-port) converter now disconnected
from ttyUSB3
[ 1740.237122] option 1-1.3.1:1.5: device disconnected
[ 1740.238483] option1 ttyUSB4: GSM modem (1-port) converter now disconnected
from ttyUSB4
[ 1740.238520] option 1-1.3.1:1.6: device disconnected
[ 1752.436526] usb 1-1.3.1: new high-speed USB device number 9 using ehci-pci
[ 1752.536781] usb 1-1.3.1: New USB device found, idVendor=1bc7, idProduct=1201
[ 1752.536803] usb 1-1.3.1: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 1752.536817] usb 1-1.3.1: Product: Android
[ 1752.536829] usb 1-1.3.1: Manufacturer: Android
[ 1752.536841] usb 1-1.3.1: SerialNumber: 0123456789ABCDEF
[ 1752.556214] option 1-1.3.1:1.0: GSM modem (1-port) converter detected
[ 1752.557043] usb 1-1.3.1: GSM modem (1-port) converter now attached to ttyUSB0
[ 1752.561008] qmi_wwan 1-1.3.1:1.2: cdc-wdm0: USB WDM device
[ 1752.561956] qmi_wwan 1-1.3.1:1.2 wwan0: register 'qmi_wwan' at
usb-0000:00:1d.0-1.3.1, WWAN/QMI device, 02:29:d8:91:80:f5
[ 1752.562375] option 1-1.3.1:1.3: GSM modem (1-port) converter detected
[ 1752.562909] usb 1-1.3.1: GSM modem (1-port) converter now attached to ttyUSB1
[ 1752.563628] option 1-1.3.1:1.4: GSM modem (1-port) converter detected
[ 1752.564122] usb 1-1.3.1: GSM modem (1-port) converter now attached to ttyUSB2
[ 1752.569265] option 1-1.3.1:1.5: GSM modem (1-port) converter detected
[ 1752.569558] usb 1-1.3.1: GSM modem (1-port) converter now attached to ttyUSB3
[ 1752.569960] option 1-1.3.1:1.6: GSM modem (1-port) converter detected
[ 1752.570227] usb 1-1.3.1: GSM modem (1-port) converter now attached to ttyUSB4
[ 1752.570857] usb-storage 1-1.3.1:1.7: USB Mass Storage device detected
[ 1752.571702] scsi host2: usb-storage 1-1.3.1:1.7
[ 1753.571280] scsi 2:0:0:0: Direct-Access Linux File-CD Gadget 0000 PQ:
0 ANSI: 2
```

Turning GPS on/off:

```
moxa@Moxa:~# sudo cell_mgmt gps_on
GPS function is *ENABLE*
moxa@Moxa:~# cat /dev/ttyUSB1
moxa@Moxa:~# sudo cell_mgmt gps_on
GPS function is *ENABLE*
```

Connecting/disconnecting to/from the cellular operator:

```
root@Moxa:~# cell_mgmt start APN="Internet"
PIN code: Disabled or verified
[/dev/cdc-wdm0] expected data format set to: 802-3
Starting network with '_qmicli --wds-start-network=Internet
--client-no-release-cid --device-open-net=net-802-3|net-no-qos-header'...
Saving state... (CID: 10)
Saving state... (PDH: 1138741424)
Network started successfully
root@Moxa:~# ifconfig wwan0
wwan0 Link encap:Ethernet HWaddr ca:fc:f5:43:cf:c8
inet addr:10.166.110.17 Bcast:10.166.110.19 Mask:255.255.255.252
inet6 addr: fe80::c8fc:f5ff:fe43:cfc8/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2 errors:0 dropped:0 overruns:0 frame:0
TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:632 (632.0 B) TX bytes:1332 (1.3 KiB)
root@Moxa:~# ping -I wwan0 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 10.166.110.17 wwan0: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=46 time=284 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=46 time=94.8 ms
root@Moxa:~# cell_mgmt stop
Killed old client process
Stopping network with '_qmicli --wds-stop-network=1138741424
--client-cid=10'...
Network stopped successfully
Clearing state...
root@Moxa:~# ifconfig wwan0
wwan0 Link encap:Ethernet HWaddr ca:fc:f5:43:cf:c8
inet6 addr: fe80::c8fc:f5ff:fe43:cfc8/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:6 errors:0 dropped:0 overruns:0 frame:0
TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:856 (856.0 B) TX bytes:1954 (1.9 KiB)
root@Moxa:~# ping -I wwan0 8.8.8.8
ping: Warning: source address might be selected on device other than wwan0.
PING 8.8.8.8 (8.8.8.8) from 192.168.4.127 wwan0: 56(84) bytes of data.
```

NOTE The `cell_mgmt` command is only applicable to the MC-1121 model in the MC-1100 series because only this model supports an LTE mini-PCIe slot.

5

System Restore

The MC-1100 is installed with the Embedded Linux operating system, which is located in the CFast card shipped with the MC-1100-LX computer. Although it rarely happens, you may find on occasion that operating system files and/or the disk file system have been damaged. In this chapter we describe how to restore the Linux operating system.

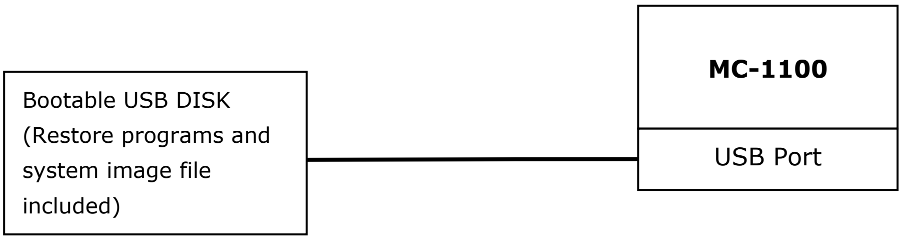
The following topics are covered in this chapter:

Restore Environment

The restore environment includes the MC-1100 embedded computer and a bootable USB disk with the restore programs and system image file.

Hardware

The hardware used includes a PC, a MC-1100 computer and a USB disk with the restore programs. **(Note: The USB disk should be at least 2GB.).**

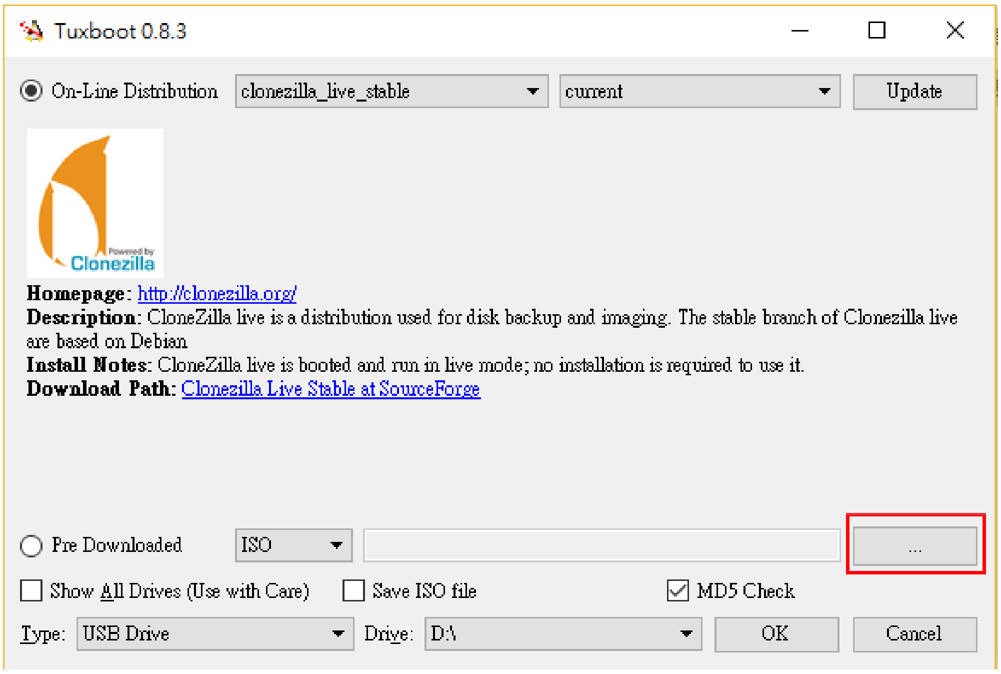

Bootable USB DISK (Restore programs and system image file included)
MC-1100
USB Port

Restore Procedure

Step 1: Prepare Your USB Drive

Windows Users

Execute **tuxboot** from the **utility_tools/tuxboot/Windows** folder on the Software CD, select **Pre Download**, and then click "**...**".



Tuxboot 0.8.3

● On-Line Distribution: clonezilla_live_stable, current

Homepage: http://clonezilla.org/
Description: CloneZilla live is a distribution used for disk backup and imaging. The stable branch of Clonezilla live are based on Debian
Install Notes: CloneZilla live is booted and run in live mode; no installation is required to use it.
Download Path: Clonezilla Live Stable at SourceForge

○ Pre Downloaded: ISO

□ Show All Drives (Use with Care) □ Save ISO file ☑ MD5 Check

Type: USB Drive Drive: D:\

Debian Linux Users

1. Install the libqt4-dev package in Debian.

```
$ apt-get install libqt4-dev
```

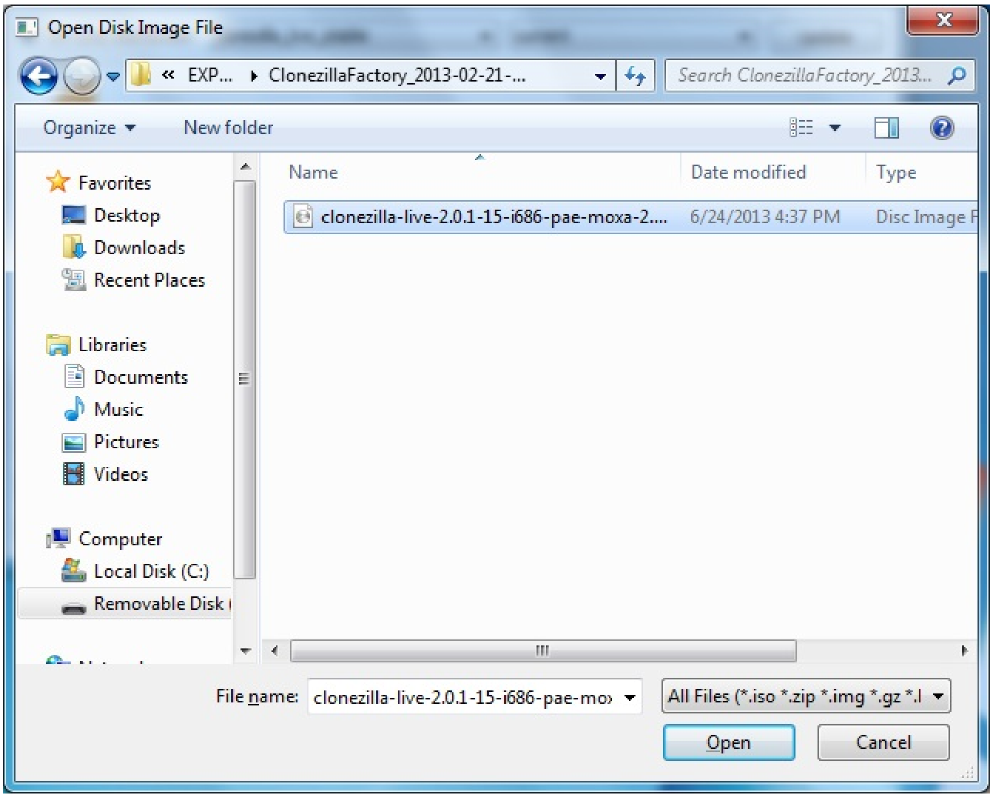
2. Untar the source file:

```
$ tar -xzvf tuxboot-0.8.src.tar.gz
$ cd tuxboot-0.8
```

3. Run the following commands to build **tuxboot**:

```
$ sed -i '/^RESOURCES/d' tuxboot.pro
$ lupdate-qt4 tuxboot.pro
$ lrelease-qt4 tuxboot.pro
$ qmake-qt4 "DEFINES += NOSTATIC" "RESOURCES -= tuxboot.qrc"
$ make
```

4. Launch the tuxboot in Debian Desktop environment.
5. Select the ISO file in the directory of **<Software DVD> \Restore\Clonezilla_Image\**



Open Disk Image File

« EXP... ▸ ClonezillaFactory_2013-02-21-...

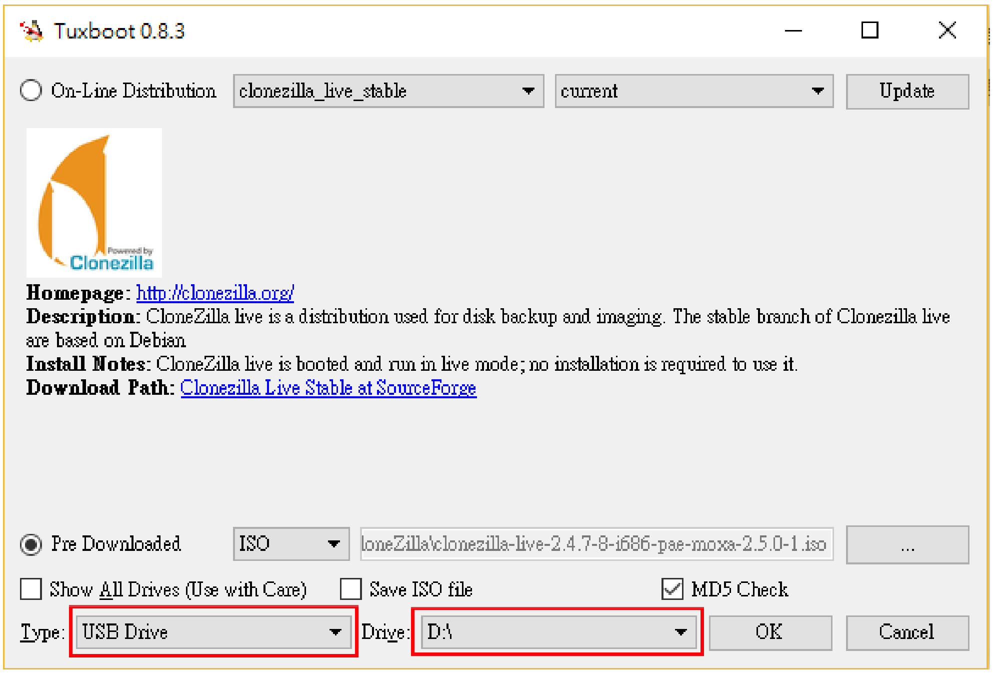
Search ClonezillaFactory_2013...

Name	Date modified	Type
clonezilla-live-2.0.1-15-i686-pae-moxa-2....	6/24/2013 4:37 PM	Disc Image F

File name: clonezilla-live-2.0.1-15-i686-pae-mo›

All Files (*.iso *.zip *.img *.gz *.l

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



Tuxboot 0.8.3

○ On-Line Distribution clonezilla_live_stable current Update

Homepage: http://clonezilla.org/

Description: CloneZilla live is a distribution used for disk backup and imaging. The stable branch of Clonezilla live are based on Debian

Install Notes: CloneZilla live is booted and run in live mode; no installation is required to use it.

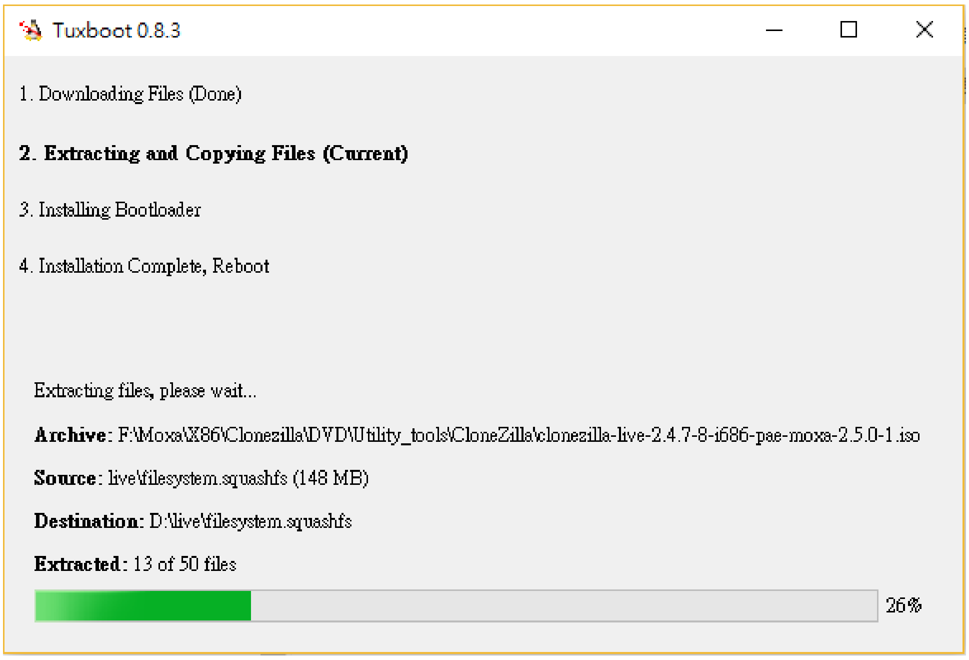
Download Path: Clonezilla Live Stable at SourceForge

● Pre Downloaded ISO loneZilla\clonezilla-live-2.4.7-8-i686-pae-moxa-2.5.0-1.iso ...

□ Show All Drives (Use with Care) □ Save ISO file ☑ MD5 Check

Type: USB Drive Drive: D:\ OK Cancel

The boot files will be copied to your USB drive.



Tuxboot 0.8.3

1. Downloading Files (Done)

2. Extracting and Copying Files (Current)

3. Installing Bootloader

4. Installation Complete, Reboot

Extracting files, please wait...

Archive: F:\Moxa\X86\Clonezilla\DVD\Utility_tools\CloneZilla\clonezilla-live-2.4.7-8-i686-pae-moxa-2.5.0-1.iso

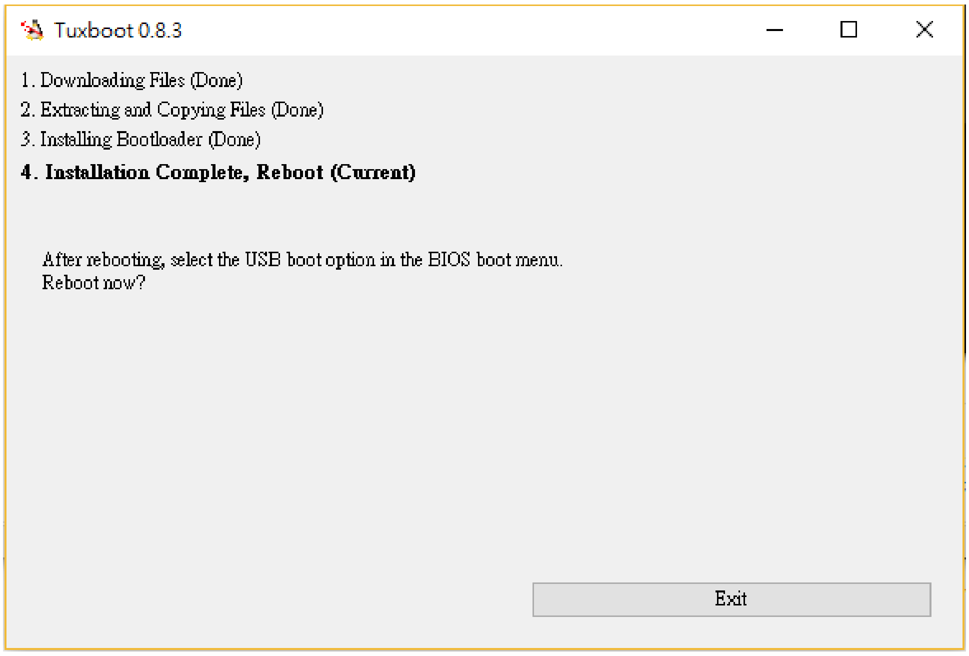
Source: live\filesystem.squashfs (148 MB)

Destination: D:\live\filesystem.squashfs

Extracted: 13 of 50 files

26%

7. After all the boot files are copied to the USB drive, click **Exit** to stop the program.



1. Downloading Files (Done)
2. Extracting and Copying Files (Done)
3. Installing Bootloader (Done)
4. Installation Complete, Reboot (Current)

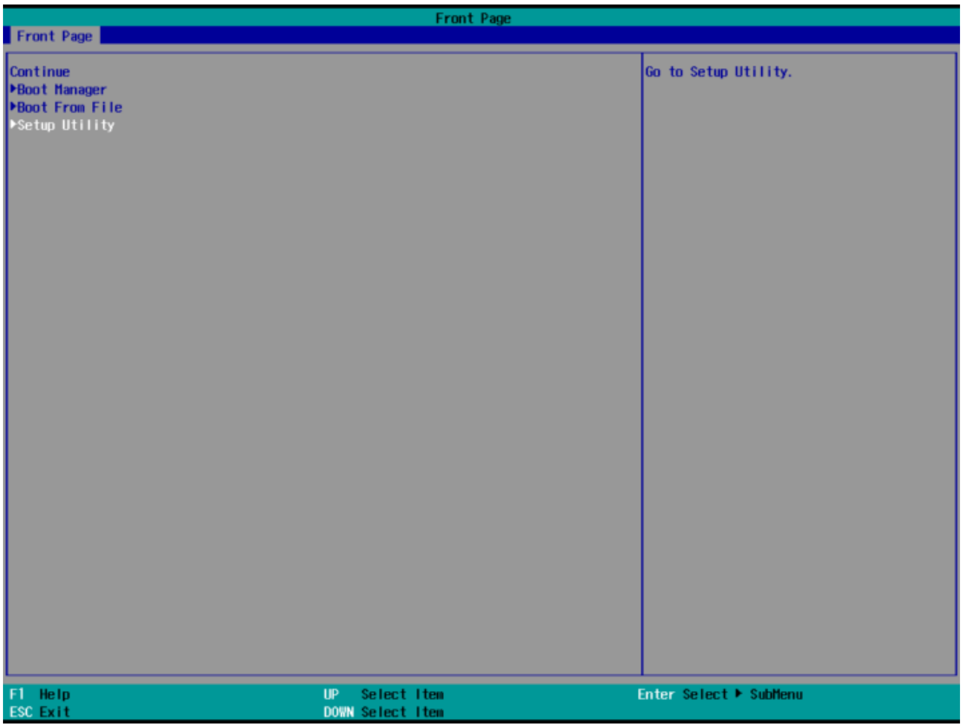
After rebooting, select the USB boot option in the BIOS boot menu.
Reboot now?

8. Manually copy the **os_image** directory from the <Software DVD> \Restore\CloneZilla_Image\ folder on the Software DVD to **\home\partimag\** on the USB drive.

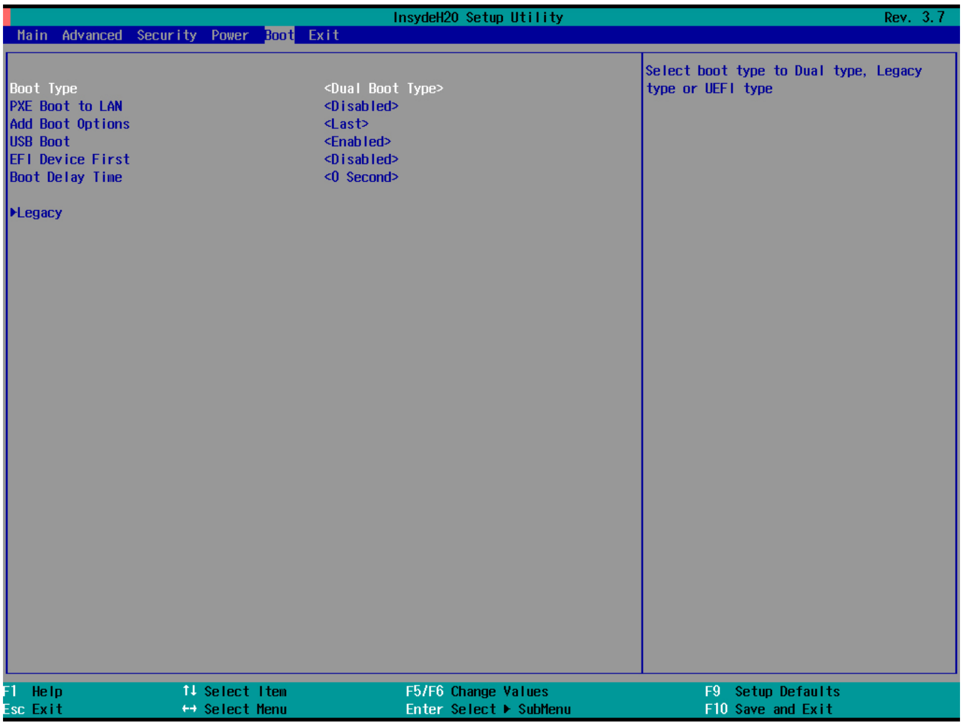
Step 2: Change the BIOS Settings

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

1. Turn on the computer and press F2. Select Setup Utility in the following screen.

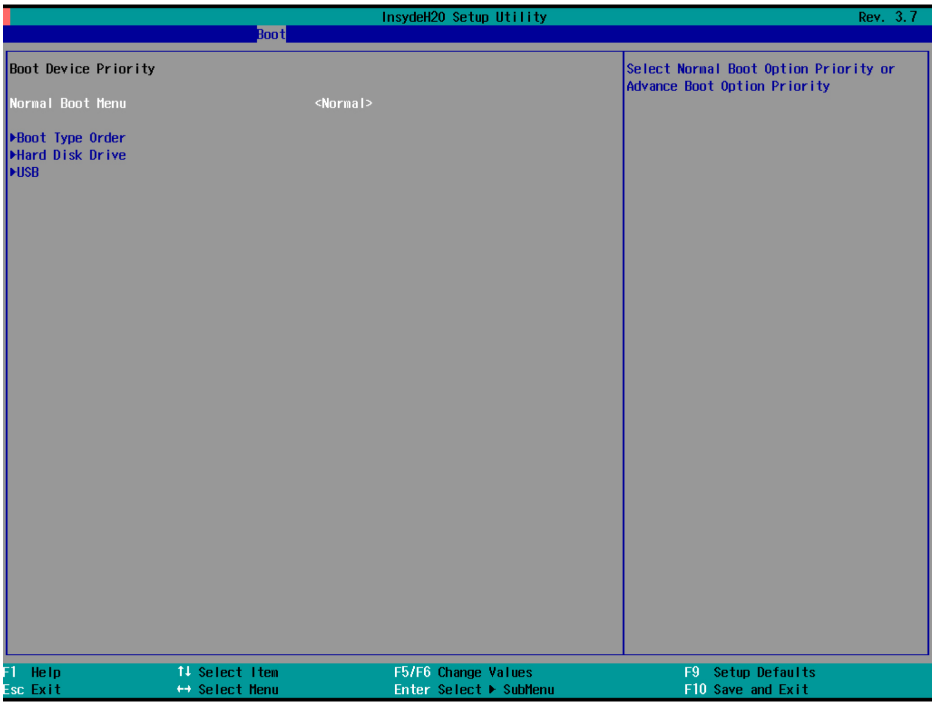


2. Select Boot and then select Legacy. Press **Enter** to continue.



```
                              InsydeH20 Setup Utility                          Rev. 3.7
 Main  Advanced  Security  Power  Boot  Exit

                                                          Select boot type to Dual type, Legacy
Boot Type                          <Dual Boot Type>       type or UEFI type
PXE Boot to LAN                    <Disabled>
Add Boot Options                   <Last>
USB Boot                           <Enabled>
EFI Device First                   <Disabled>
Boot Delay Time                    <0 Second>

▶Legacy

F1  Help            ↑↓ Select Item       F5/F6 Change Values         F9  Setup Defaults
Esc Exit            ↔ Select Menu        Enter Select ▶ SubMenu      F10 Save and Exit
```

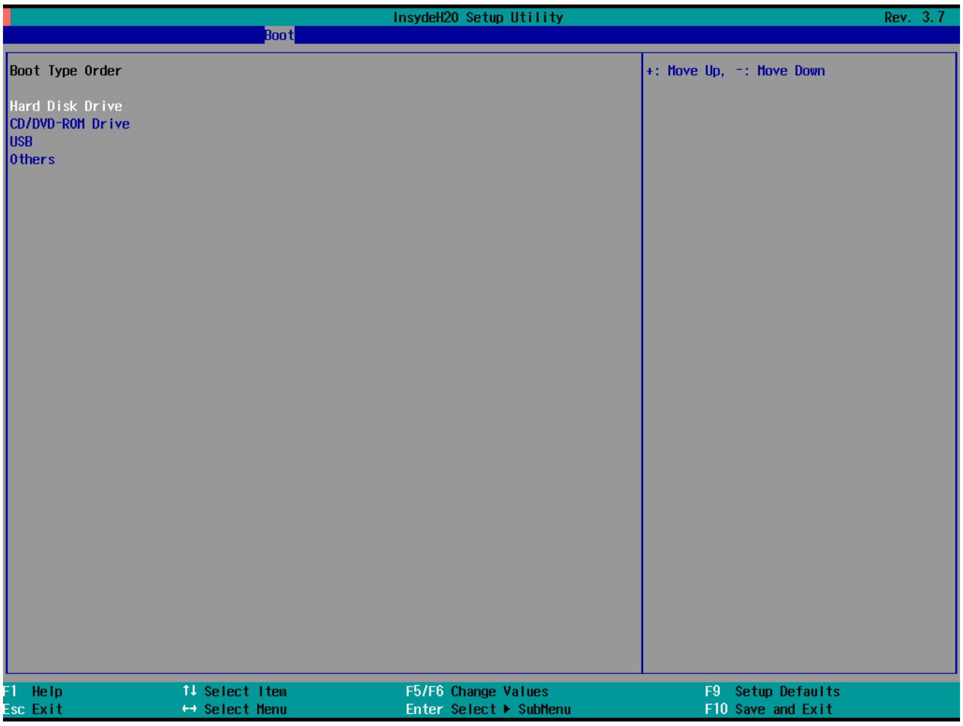
3. Select Boot order.



```
                              InsydeH20 Setup Utility                          Rev. 3.7
                                    Boot
Boot Device Priority                                      Select Normal Boot Option Priority or
                                                          Advance Boot Option Priority
Normal Boot Menu                   <Normal>

▶Boot Type Order
▶Hard Disk Drive
▶USB

F1  Help            ↑↓ Select Item       F5/F6 Change Values         F9  Setup Defaults
Esc Exit            ↔ Select Menu        Enter Select ▶ SubMenu      F10 Save and Exit
```

4. Select USB disk and then press "+" to move it to the first boot device position.

   Warning: An incorrect boot priority will lead to restore or boot failure.



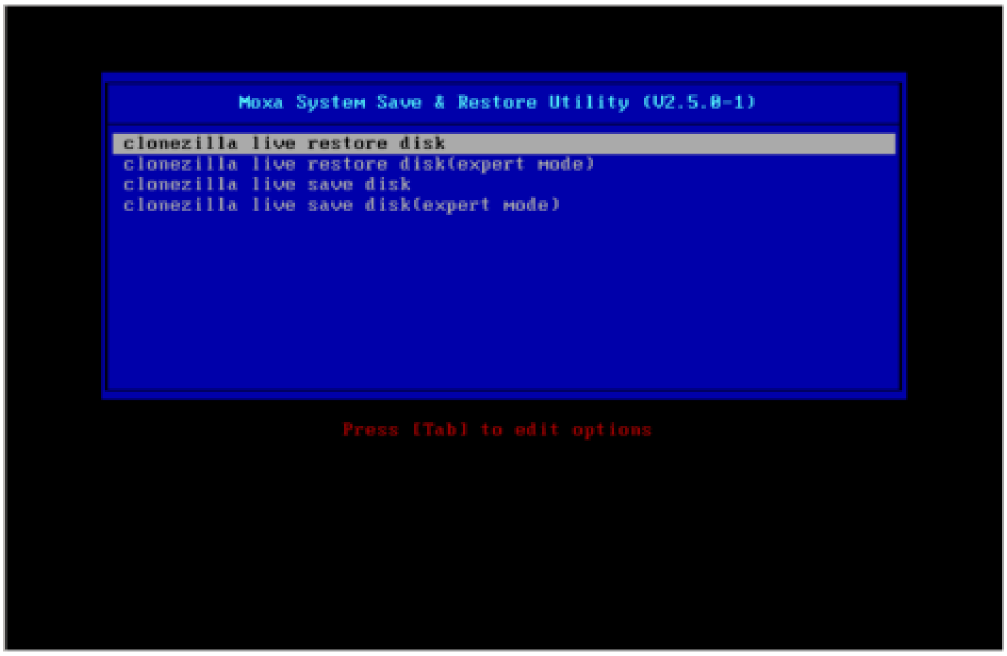
```
                         InsydeH2O Setup Utility                         Rev. 3.7
                    Boot
Boot Type Order                                          +: Move Up, -: Move Down

Hard Disk Drive
CD/DVD-ROM Drive
USB
Others

F1  Help        ↑↓ Select Item     F5/F6 Change Values       F9  Setup Defaults
Esc Exit        ↔  Select Menu     Enter Select ▸ SubMenu    F10 Save and Exit
```

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

Step 3: Restore the system from USB drive

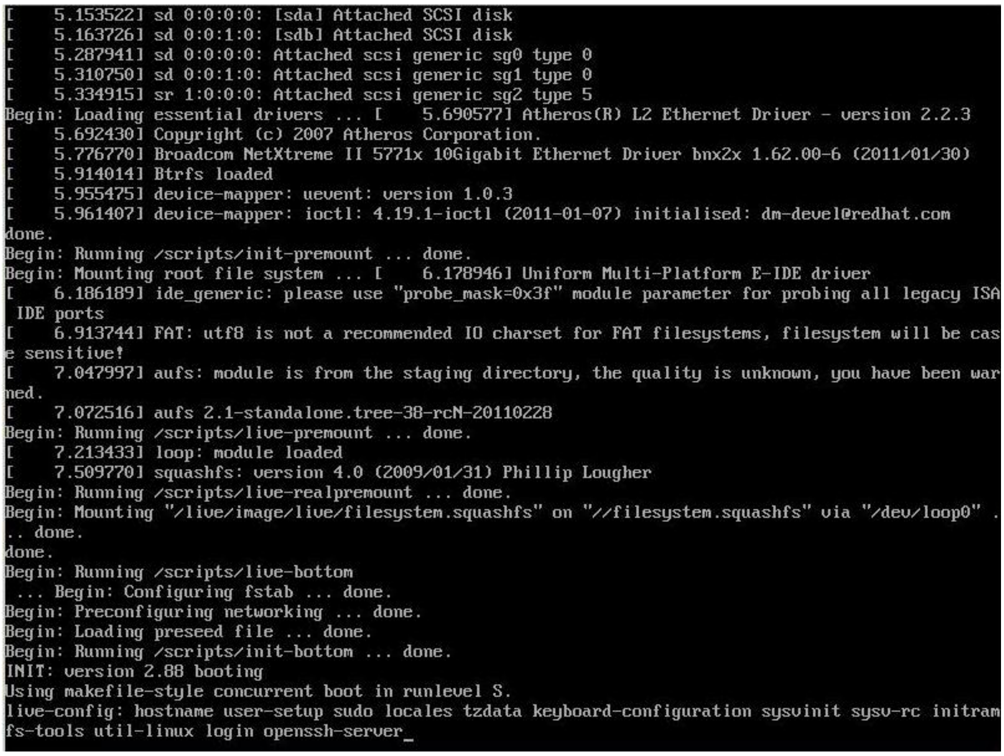
Connect the USB disk to any of the MC-1100's USB ports and then reboot the computer. The system will boot from the USB disk and the Pre-installation Environment and the restore utility will appear.

1. Select clonezilla live restore disk.



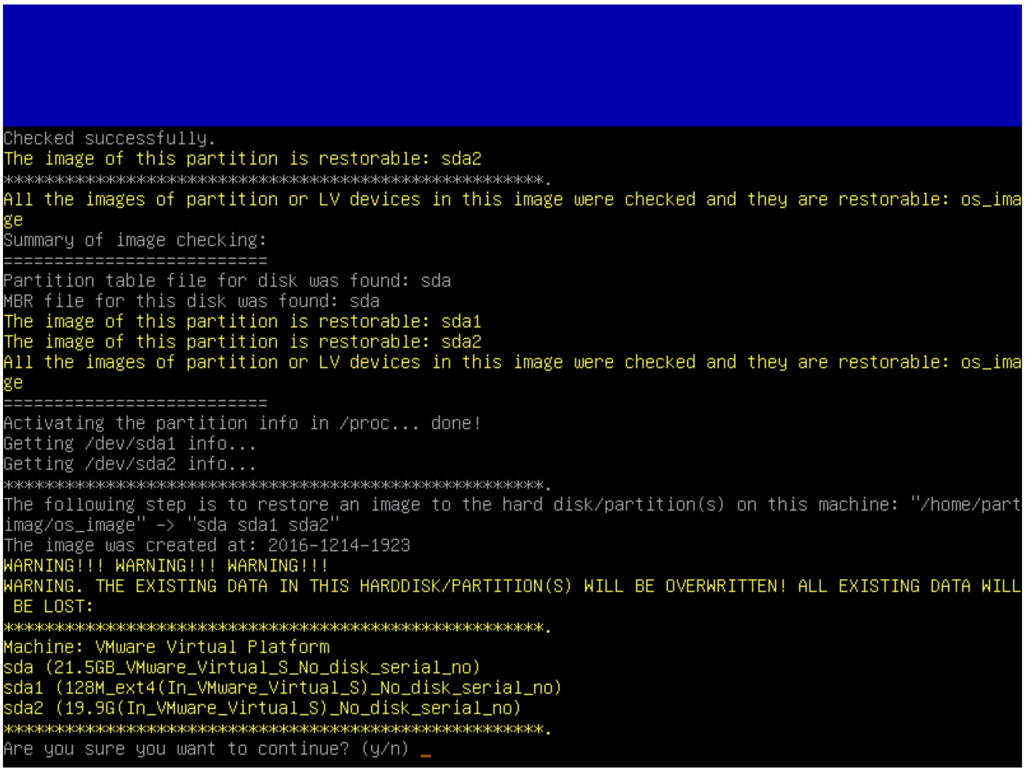
```
        Moxa System Save & Restore Utility (V2.5.0-1)

 clonezilla live restore disk
 clonezilla live restore disk(expert mode)
 clonezilla live save disk
 clonezilla live save disk(expert mode)

                 Press [Tab] to edit options
```

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



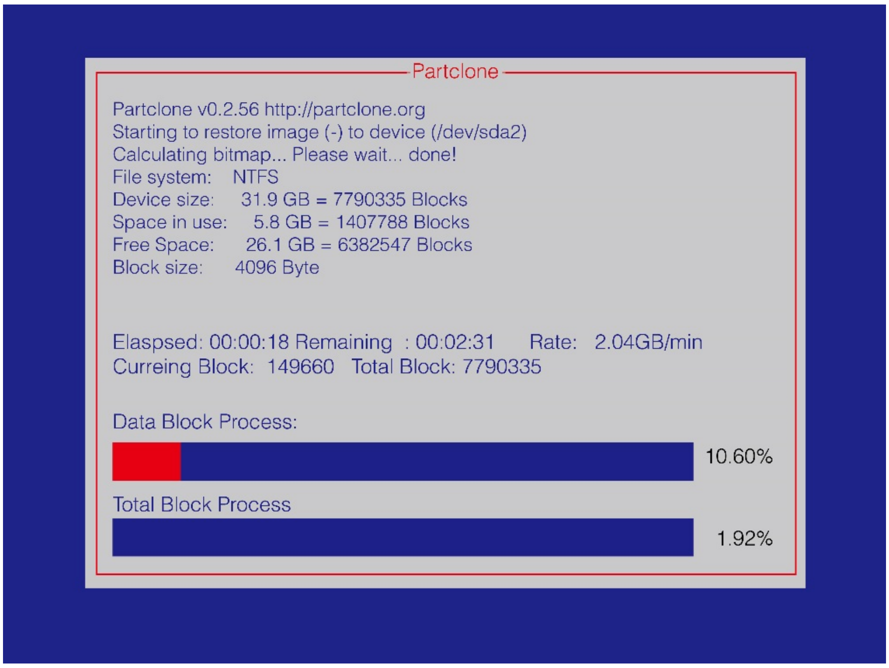
```
[    5.153522] sd 0:0:0:0: [sda] Attached SCSI disk
[    5.163726] sd 0:0:1:0: [sdb] Attached SCSI disk
[    5.287941] sd 0:0:0:0: Attached scsi generic sg0 type 0
[    5.310750] sd 0:0:1:0: Attached scsi generic sg1 type 0
[    5.334915] sr 1:0:0:0: Attached scsi generic sg2 type 5
Begin: Loading essential drivers ... [    5.690577] Atheros(R) L2 Ethernet Driver - version 2.2.3
[    5.692430] Copyright (c) 2007 Atheros Corporation.
[    5.776770] Broadcom NetXtreme II 5771x 10Gigabit Ethernet Driver bnx2x 1.62.00-6 (2011/01/30)
[    5.914014] Btrfs loaded
[    5.955475] device-mapper: uevent: version 1.0.3
[    5.961407] device-mapper: ioctl: 4.19.1-ioctl (2011-01-07) initialised: dm-devel@redhat.com
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... [    6.178946] Uniform Multi-Platform E-IDE driver
[    6.186189] ide_generic: please use "probe_mask=0x3f" module parameter for probing all legacy ISA
 IDE ports
[    6.913744] FAT: utf8 is not a recommended IO charset for FAT filesystems, filesystem will be cas
e sensitive!
[    7.047997] aufs: module is from the staging directory, the quality is unknown, you have been war
ned.
[    7.072516] aufs 2.1-standalone.tree-38-rcN-20110228
Begin: Running /scripts/live-premount ... done.
[    7.213433] loop: module loaded
[    7.509770] squashfs: version 4.0 (2009/01/31) Phillip Lougher
Begin: Running /scripts/live-realpremount ... done.
Begin: Mounting "/live/image/live/filesystem.squashfs" on "//filesystem.squashfs" via "/dev/loop0" .
.. done.
done.
Begin: Running /scripts/live-bottom
 ... Begin: Configuring fstab ... done.
Begin: Preconfiguring networking ... done.
Begin: Loading preseed file ... done.
Begin: Running /scripts/init-bottom ... done.
INIT: version 2.88 booting
Using makefile-style concurrent boot in runlevel S.
live-config: hostname user-setup sudo locales tzdata keyboard-configuration sysvinit sysv-rc initram
fs-tools util-linux login openssh-server_
```

3. Enter y to continue the restore process.



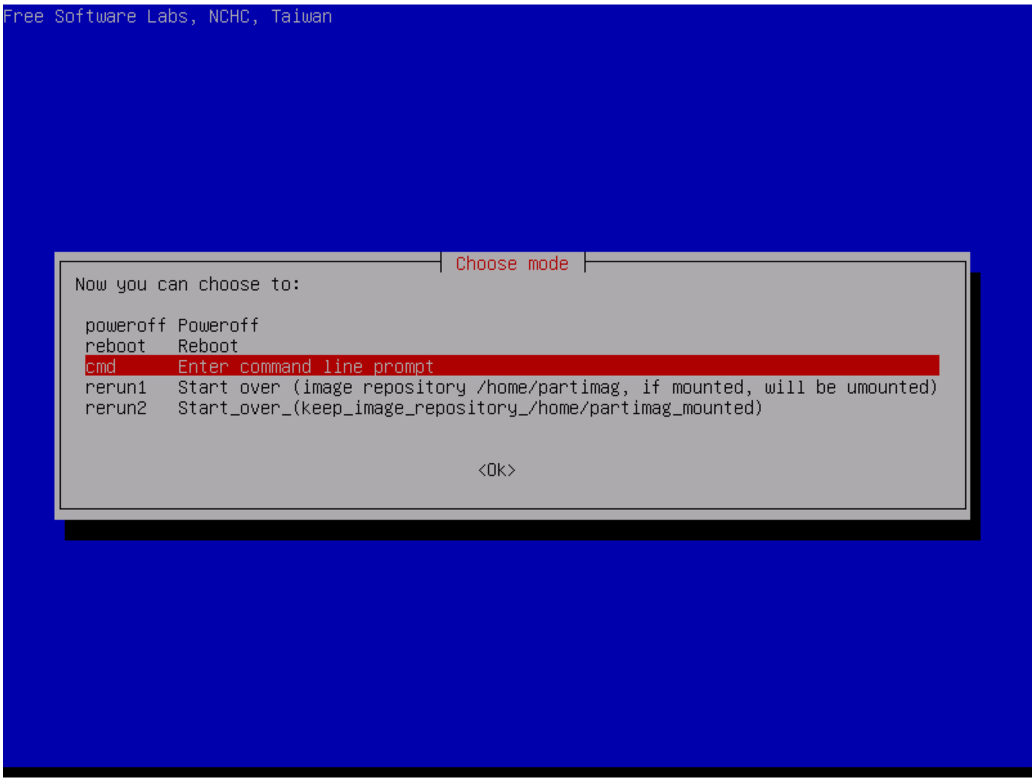
```
Checked successfully.
The image of this partition is restorable: sda2
*****************************************************.
All the images of partition or LV devices in this image were checked and they are restorable: os_ima
ge
Summary of image checking:
==========================
Partition table file for disk was found: sda
MBR file for this disk was found: sda
The image of this partition is restorable: sda1
The image of this partition is restorable: sda2
All the images of partition or LV devices in this image were checked and they are restorable: os_ima
ge
==========================
Activating the partition info in /proc... done!
Getting /dev/sda1 info...
Getting /dev/sda2 info...
*****************************************************.
The following step is to restore an image to the hard disk/partition(s) on this machine: "/home/part
imag/os_image" -> "sda sda1 sda2"
The image was created at: 2016-1214-1923
WARNING!!! WARNING!!! WARNING!!!
WARNING. THE EXISTING DATA IN THIS HARDDISK/PARTITION(S) WILL BE OVERWRITTEN! ALL EXISTING DATA WILL
 BE LOST:
*****************************************************.
Machine: VMware Virtual Platform
sda (21.5GB_VMware_Virtual_S_No_disk_serial_no)
sda1 (128M_ext4(In_VMware_Virtual_S)_No_disk_serial_no)
sda2 (19.9G(In_VMware_Virtual_S)_No_disk_serial_no)
*****************************************************.
Are you sure you want to continue? (y/n) _
```

4. Wait for the process to finish.



```
Partclone

Partclone v0.2.56 http://partclone.org
Starting to restore image (-) to device (/dev/sda2)
Calculating bitmap... Please wait... done!
File system:    NTFS
Device size:     31.9 GB = 7790335 Blocks
Space in use:     5.8 GB = 1407788 Blocks
Free Space:      26.1 GB = 6382547 Blocks
Block size:      4096 Byte


Elaspsed: 00:00:18 Remaining  : 00:02:31     Rate:  2.04GB/min
Curreing Block:  149660   Total Block: 7790335

Data Block Process:
                                                   10.60%
Total Block Process
                                                    1.92%
```

5. Select Poweroff to power off the computer.



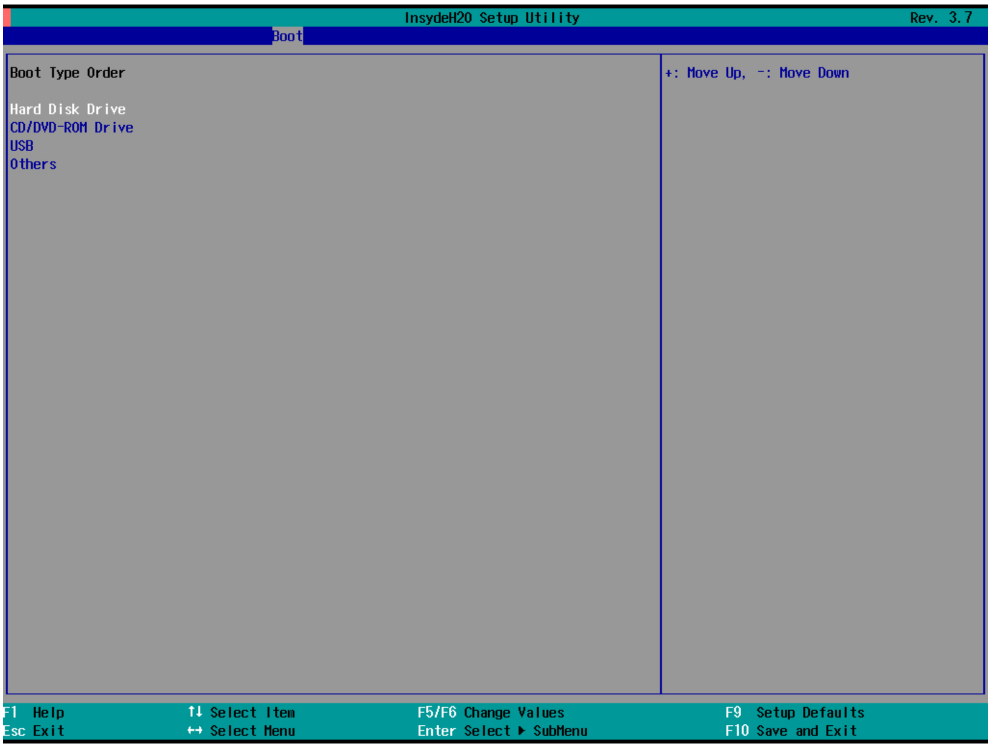
```
Free Software Labs, NCHC, Taiwan

                              Choose mode
 Now you can choose to:

  poweroff Poweroff
  reboot   Reboot
  cmd      Enter command line prompt
  rerun1   Start over (image repository /home/partimag, if mounted, will be umounted)
  rerun2   Start_over_(keep_image_repository_/home/partimag_mounted)


                                 <Ok>
```

6. 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.



```
InsydeH20 Setup Utility                                              Rev. 3.7
                                   Boot
Boot Type Order                                      +: Move Up, -: Move Down

Hard Disk Drive
CD/DVD-ROM Drive
USB
Others

F1  Help           ↑↓ Select Item      F5/F6 Change Values         F9  Setup Defaults
Esc Exit           ↔ Select Menu       Enter Select ▶ SubMenu      F10 Save and Exit
```

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

Step 5: Reboot the Computer

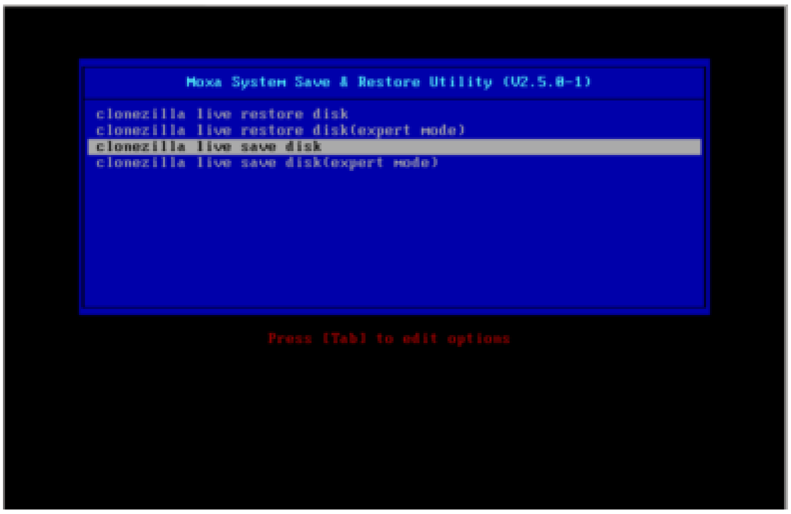
You need to wait about 10 to 15 minutes for the system to restart, since the system configuration files will be initiated while booting up for the first time. **Do not turn off the computer or shut down the computer** while the system is restarting; otherwise, the IIS service will be terminated. When 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 restore in case the system crashes. Before saving the system to the USB drive, we suggest you remove all files under **\home\partimag\** on the USB drive. In addition, change the BIOS settings to make the USB drive the first boot priority.

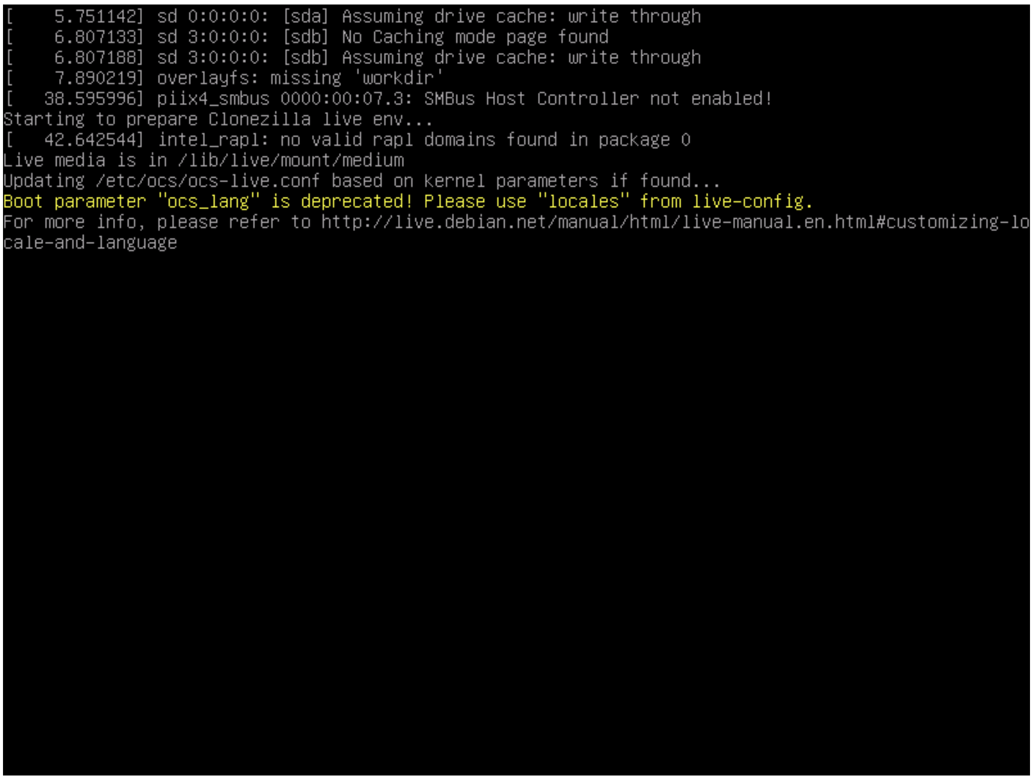
When the system has been launched, take the following steps:

1. Select clonezilla live save disk.



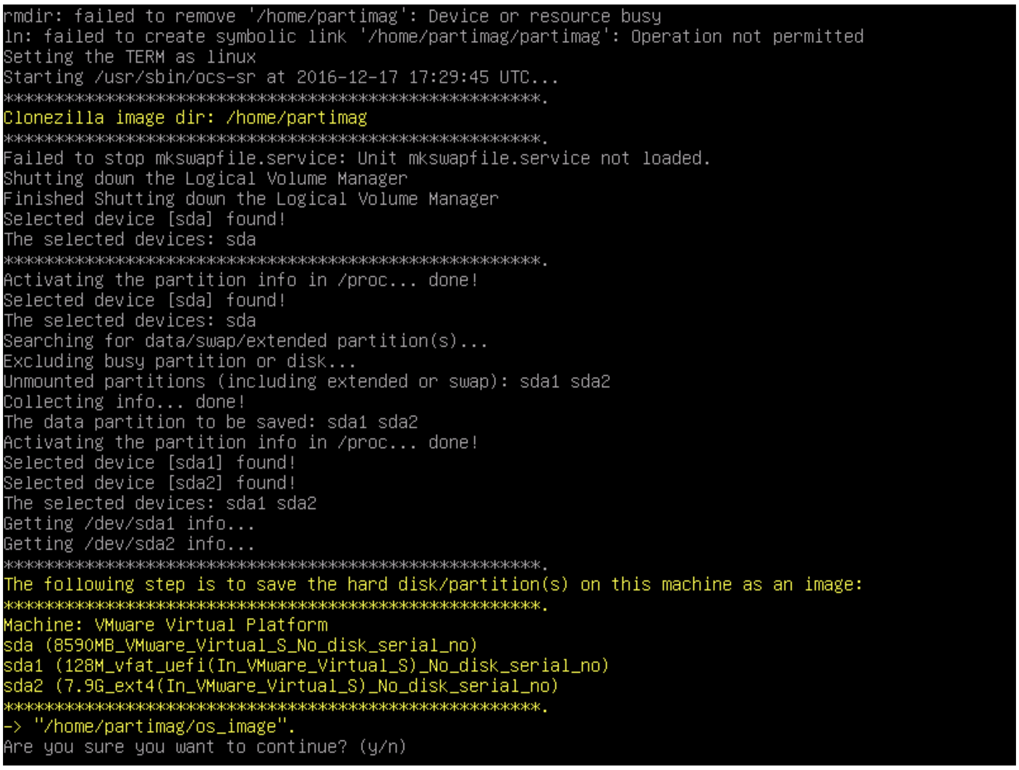
```
Moxa System Save & Restore Utility (V2.5.0-1)

clonezilla live restore disk
clonezilla live restore disk(expert mode)
clonezilla live save disk
clonezilla live save disk(expert mode)

Press [Tab] to edit options
```

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



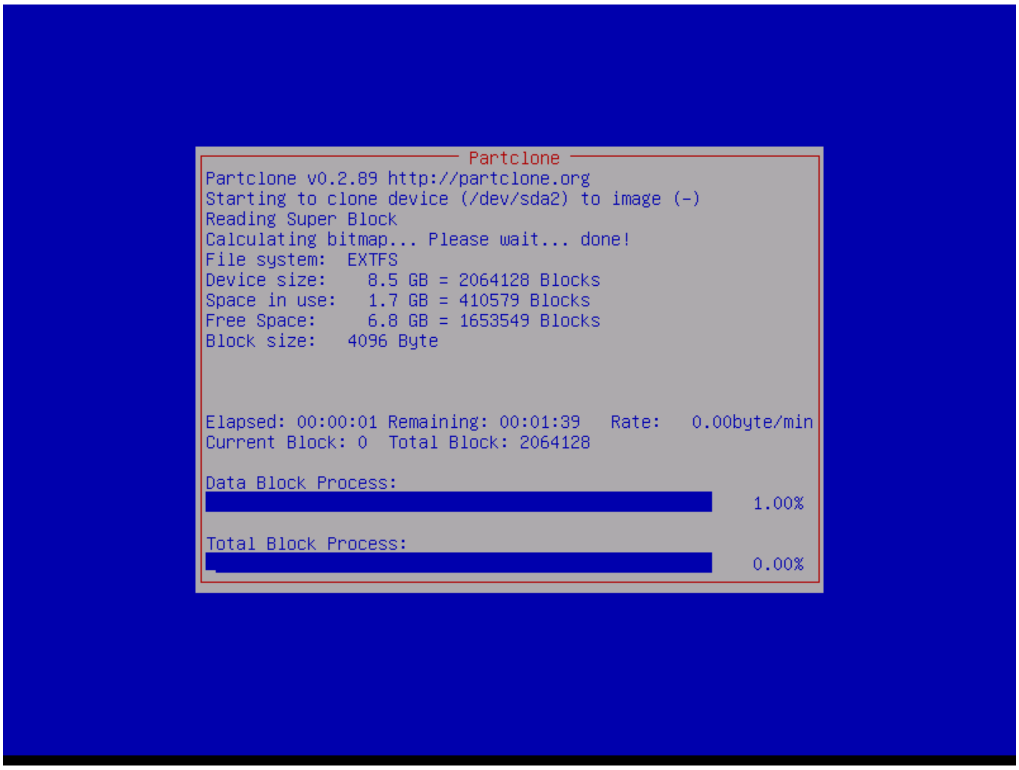
```
[    5.751142] sd 0:0:0:0: [sda] Assuming drive cache: write through
[    6.807133] sd 3:0:0:0: [sdb] No Caching mode page found
[    6.807188] sd 3:0:0:0: [sdb] Assuming drive cache: write through
[    7.890219] overlayfs: missing 'workdir'
[   38.595996] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
Starting to prepare Clonezilla live env...
[   42.642544] intel_rapl: no valid rapl domains found in package 0
Live media is in /lib/live/mount/medium
Updating /etc/ocs/ocs-live.conf based on kernel parameters if found...
Boot parameter "ocs_lang" is deprecated! Please use "locales" from live-config.
For more info, please refer to http://live.debian.net/manual/html/live-manual.en.html#customizing-lo
cale-and-language
```

3. Enter **y** to continue.



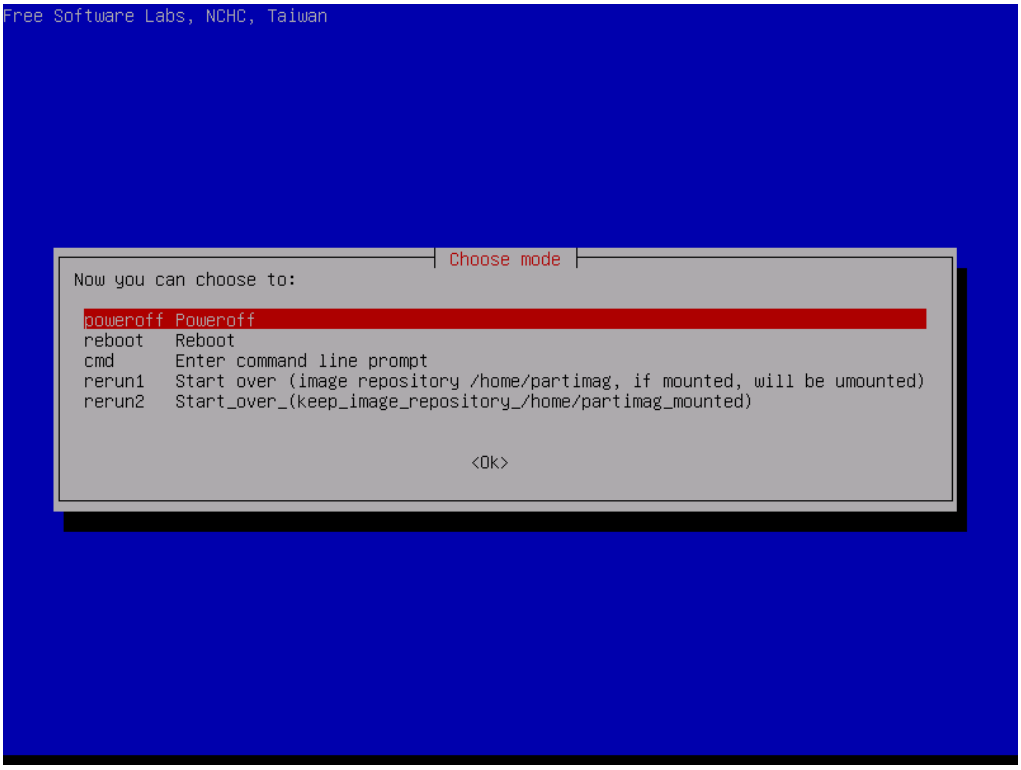
```
rmdir: failed to remove '/home/partimag': Device or resource busy
ln: failed to create symbolic link '/home/partimag/partimag': Operation not permitted
Setting the TERM as linux
Starting /usr/sbin/ocs-sr at 2016-12-17 17:29:45 UTC...
******************************************************.
Clonezilla image dir: /home/partimag
******************************************************.
Failed to stop mkswapfile.service: Unit mkswapfile.service not loaded.
Shutting down the Logical Volume Manager
Finished Shutting down the Logical Volume Manager
Selected device [sda] found!
The selected devices: sda
******************************************************.
Activating the partition info in /proc... done!
Selected device [sda] found!
The selected devices: sda
Searching for data/swap/extended partition(s)...
Excluding busy partition or disk...
Unmounted partitions (including extended or swap): sda1 sda2
Collecting info... done!
The data partition to be saved: sda1 sda2
Activating the partition info in /proc... done!
Selected device [sda1] found!
Selected device [sda2] found!
The selected devices: sda1 sda2
Getting /dev/sda1 info...
Getting /dev/sda2 info...
******************************************************.
The following step is to save the hard disk/partition(s) on this machine as an image:
******************************************************.
Machine: VMware Virtual Platform
sda (8590MB_VMware_Virtual_S_No_disk_serial_no)
sda1 (128M_vfat_uefi(In_VMware_Virtual_S)_No_disk_serial_no)
sda2 (7.9G_ext4(In_VMware_Virtual_S)_No_disk_serial_no)
******************************************************.
-> "/home/partimag/os_image".
Are you sure you want to continue? (y/n)
```

4. Wait for the process to finish.



```
Partclone
Partclone v0.2.89 http://partclone.org
Starting to clone device (/dev/sda2) to image (-)
Reading Super Block
Calculating bitmap... Please wait... done!
File system:  EXTFS
Device size:    8.5 GB = 2064128 Blocks
Space in use:   1.7 GB = 410579 Blocks
Free Space:     6.8 GB = 1653549 Blocks
Block size:   4096 Byte


Elapsed: 00:00:01 Remaining: 00:01:39   Rate:   0.00byte/min
Current Block: 0  Total Block: 2064128

Data Block Process:
                                                     1.00%

Total Block Process:
                                                     0.00%
```

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



```
Free Software Labs, NCHC, Taiwan

                          Choose mode
 Now you can choose to:

  poweroff Poweroff
  reboot   Reboot
  cmd      Enter command line prompt
  rerun1   Start over (image repository /home/partimag, if mounted, will be umounted)
  rerun2   Start_over_(keep_image_repository_/home/partimag_mounted)

                              <Ok>
```

A

Software Components

Name	Version	Description
acl	2.2.52-2	Access control list utilities
acpi	1.7-1	displays information on ACPI devices
acpi-support-base	0.142-6	scripts for handling base ACPI events such as the power button
acpid	1:2.0.23-2	Advanced Configuration and Power Interface event daemon
adduser	3.113+nmu3	add and remove users and groups
anacron	2.3-23	cron-like program that doesn't go by time
apache2	2.4.10-10+deb8u5	Apache HTTP Server
apache2-bin	2.4.10-10+deb8u5	Apache HTTP Server (modules and other binary files)
apache2-data	2.4.10-10+deb8u5	Apache HTTP Server (common files)
apache2-utils	2.4.10-10+deb8u5	Apache HTTP Server (utility programs for web servers)
apt	1.0.9.8.3	Command-line package manager
apt-listchanges	2.85.13+nmu1	package change history notification tool
apt-utils	1.0.9.8.3	package management related utility programs
aptitude	0.6.11-1+b1	terminal-based package manager
aptitude-common	0.6.11-1	architecture independent files for the aptitude package manager
aptitude-doc-en	0.6.11-1	English manual for aptitude, a terminal-based package manager
at	3.1.16-1	Delayed job execution and batch processing
avahi-autoipd	0.6.31-5	Avahi IPv4LL network address configuration daemon
base-files	8+deb8u5	Debian base system miscellaneous files
base-passwd	3.5.37	Debian base system master password and group files
bash	4.3-11+b1	GNU Bourne Again SHell
bash-completion	1:2.1-4	programmable completion for the bash shell
bc	1.06.95-9	GNU bc arbitrary precision calculator language
bind9-host	1:9.9.5.dfsg-9+deb8u6	Version of 'host' bundled with BIND 9.X
binutils	2.25-5	GNU assembler, linker and binary utilities
bluetooth	5.23-2	Bluetooth support
bluez	5.23-2+b1	Bluetooth tools and daemons
bridge-utils	1.5-9	Utilities for configuring the Linux Ethernet bridge
bsd-mailx	8.1.2-0.20141216cvs-2	simple mail user agent
bsdmainutils	9.0.6	collection of more utilities from FreeBSD
bsdutils	1:2.25.2-6	basic utilities from 4.4BSD-Lite

Name	Version	Description
build-essential	11.7	Informational list of build-essential packages
busybox	1:1.22.0-9+deb8u1	Tiny utilities for small and embedded systems
bzip2	1.0.6-7+b3	high-quality block-sorting file compressor - utilities
ca-certificates	20141019+deb8u1	Common CA certificates
console-setup	1.123	console font and keymap setup program
console-setup-linux	1.123	Linux specific part of console-setup
coreutils	8.23-4	GNU core utilities
cpio	2.11+dfsg-4.1+deb8u1	GNU cpio -- a program to manage archives of files
cpp	4:4.9.2-2	GNU C preprocessor (cpp)
cpp-4.9	4.9.2-10	GNU C preprocessor
crda	3.13-1	wireless Central Regulatory Domain Agent
cron	3.0pl1-127+deb8u1	process scheduling daemon
da720-misc	1.0.0	Moxa da720 Led/Relay/On Board Serial Interface device driver
da720-mxser	1.0.0	mxser driver for MC-1100 series
da720-setinterface	1.0.0	utility for mxser
da720-watchdog	1.0.0	Moxa da720 watchdog device driver
dash	0.5.7-4+b1	POSIX-compliant shell
dbus	1.8.20-0+deb8u1	simple inter process messaging system (daemon and utilities)
dc	1.06.95-9	GNU dc arbitrary precision reverse-polish calculator
debconf	1.5.56	Debian configuration management system
debconf-i18n	1.5.56	full internationalization support for debconf
debian-archive-keyring	2014.3	GnuPG archive keys of the Debian archive
debian-faq	5.0.3	Debian Frequently Asked Questions
debianutils	4.4+b1	Miscellaneous utilities specific to Debian
dictionaries-common	1.23.17	spelling dictionaries - common utilities
diffutils	1:3.3-1+b1	File comparison utilities
discover	2.1.2-7	hardware identification system
discover-data	2.2013.01.11	Data lists for Discover hardware detection system
dmidecode	2.12-3	SMBIOS/DMI table decoder
dmsetup	2:1.02.90-2.2+deb8u1	Linux Kernel Device Mapper userspace library
dnsutils	1:9.9.5.dfsg-9+deb8u6	Clients provided with BIND
doc-debian	6.2	Debian Project documentation and other documents
docutils-common	0.12+dfsg-1	text processing system for reStructuredText - common data
docutils-doc	0.12+dfsg-1	text processing system for reStructuredText - documentation
dpkg	1.17.27	Debian package management system
dpkg-dev	1.17.27	Debian package development tools
e2fslibs:amd64	1.42.12-1.1	ext2/ext3/ext4 file system libraries
e2fsprogs	1.42.12-1.1	ext2/ext3/ext4 file system utilities
easy-rsa	2.2.2-1	Simple shell based CA utility
emacsen-common	2.0.8	Common facilities for all emacsen
ethtool	1:3.16-1	display or change Ethernet device settings
fakeroot	1.20.2-1	tool for simulating superuser privileges
file	1:5.22+15-2+deb8u1	Determines file type using "magic" numbers

Name	Version	Description
findutils	4.4.2-9+b1	utilities for finding files--find, xargs
firmware-linux-free	3.3	Binary firmware for various drivers in the Linux kernel
firmware-misc-nonfree	20160110-1~bpo8+1	Binary firmware for various drivers in the Linux kernel
fontconfig	2.11.0-6.3	generic font configuration library - support binaries
fontconfig-config	2.11.0-6.3	generic font configuration library - configuration
fonts-dejavu-core	2.34-1	Vera font family derivate with additional characters
g++	4:4.9.2-2	GNU C++ compiler
g++-4.9	4.9.2-10	GNU C++ compiler
gcc	4:4.9.2-2	GNU C compiler
gcc-4.8-base:amd64	4.8.4-1	GCC, the GNU Compiler Collection (base package)
gcc-4.9	4.9.2-10	GNU C compiler
gcc-4.9-base:amd64	4.9.2-10	GCC, the GNU Compiler Collection (base package)
geoip-database	20150317-1	IP lookup command line tools that use the GeoIP library (country database)
gettext-base	0.19.3-2	GNU Internationalization utilities for the base system
gnupg	1.4.18-7+deb8u1	GNU privacy guard - a free PGP replacement
gnupg-agent	2.0.26-6	GNU privacy guard - password agent
gnupg2	2.0.26-6	GNU privacy guard - a free PGP replacement (new v2.x)
gpgv	1.4.18-7+deb8u1	GNU privacy guard - signature verification tool
grep	2.20-4.1	GNU grep, egrep and fgrep
groff-base	1.22.2-8	GNU troff text-formatting system (base system components)
grub-common	2.02~beta2-22+deb8u1	GRand Unified Bootloader (common files)
grub-pc	2.02~beta2-22+deb8u1	GRand Unified Bootloader, version 2 (PC/BIOS version)
grub-pc-bin	2.02~beta2-22+deb8u1	GRand Unified Bootloader, version 2 (PC/BIOS binaries)
grub2-common	2.02~beta2-22+deb8u1	GRand Unified Bootloader (common files for version 2)
gzip	1.6-4	GNU compression utilities
hdparm	9.43-2	tune hard disk parameters for high performance
hicolor-icon-theme	0.13-1	default fallback theme for FreeDesktop.org icon themes
host	1:9.9.5.dfsg-9+deb8u6	Transitional package
hostname	3.15	utility to set/show the host name or domain name
iamerican	3.3.02-6	American English dictionary for ispell (standard version)
ibritish	3.3.02-6	British English dictionary for ispell (standard version)
ienglish-common	3.3.02-6	Common files for British and American ispell

Name	Version	Description
		dictionaries
ifenslave	2.6	configure network interfaces for parallel routing (bonding)
ifstat	1.1-8+b1	InterFace STATistics Monitoring
ifupdown	0.7.53.1	high level tools to configure network interfaces
info	5.2.0.dfsg.1-6	Standalone GNU Info documentation browser
init	1.22	System-V-like init utilities - metapackage
init-system-helpers	1.22	helper tools for all init systems
initramfs-tools	0.120+deb8u2	generic modular initramfs generator
initscripts	2.88dsf-59	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	5.2.0.dfsg.1-6	Manage installed documentation in info format
installation-report	2.58	system installation report
iproute	1:3.16.0-2	transitional dummy package for iproute2
iproute2	3.16.0-2	networking and traffic control tools
ipsec-tools	1:0.8.2+20140711-2+de	IPsec utilities
iptables	1.4.21-2+b1	administration tools for packet filtering and NAT
iputils-ping	3:20121221-5+b2	Tools to test the reachability of network hosts
irqbalance	1.1.0-2~bpo8+1	Daemon to balance interrupts for SMP systems
isc-dhcp-client	4.3.1-6+deb8u2	DHCP client for automatically obtaining an IP address
isc-dhcp-common	4.3.1-6+deb8u2	common files used by all of the isc-dhcp packages
isc-dhcp-server	4.3.1-6+deb8u2	ISC DHCP server for automatic IP address assignment
iso-codes	3.57-1	ISO language, territory, currency, script codes and their translations
ispell	3.3.02-6	International Ispell (an interactive spelling corrector)
iw	3.17-1	tool for configuring Linux wireless devices
kbd	1.15.5-2	Linux console font and keytable utilities
keyboard-configuration	1.123	system-wide keyboard preferences
klibc-utils	2.0.4-2	small utilities built with klibc for early boot
kmod	18-3	tools for managing Linux kernel modules
krb5-locales	1.12.1+dfsg-19+deb8u2	Internationalization support for MIT Kerberos
laptop-detect	0.13.7	attempt to detect a laptop
less	458-3	pager program similar to more
libacl1:amd64	2.2.52-2	Access control list shared library
libalgorithm-c3-perl	0.09-1	Perl module for merging hierarchies using the C3 algorithm
libalgorithm-diff-perl	1.19.02-3	module to find differences between files
libalgorithm-diff-xs-perl	0.04-3+b1	module to find differences between files (XS accelerated)
libalgorithm-merge-perl	0.08-2	Perl module for three-way merge of textual

Name	Version	Description
		data
libapache2-mod-php5	5.6.24+dfsg-0+deb8u1	server-side, HTML-embedded scripting language (Apache 2 module)
libapr1:amd64	1.5.1-3	Apache Portable Runtime Library
libaprutil1:amd64	1.5.4-1	Apache Portable Runtime Utility Library
libaprutil1-dbd-sqlite3:amd64	1.5.4-1	Apache Portable Runtime Utility Library - SQLite3 Driver
libaprutil1-ldap:amd64	1.5.4-1	Apache Portable Runtime Utility Library - LDAP Driver
libapt-inst1.5:amd64	1.0.9.8.3	deb package format runtime library
libapt-pkg4.12:amd64	1.0.9.8.3	package management runtime library
libarchive-extract-perl	0.72-1	generic archive extracting module
libasan1:amd64	4.9.2-10	AddressSanitizer -- a fast memory error detector
libasprintf0c2:amd64	0.19.3-2	GNU library to use fprintf and friends in C++
libassuan0:amd64	2.1.2-2	IPC library for the GnuPG components
libatk1.0-0:amd64	2.14.0-1	ATK accessibility toolkit
libatk1.0-data	2.14.0-1	Common files for the ATK accessibility toolkit
libatomic1:amd64	4.9.2-10	support library providing __atomic built-in functions
libattr1:amd64	1:2.4.47-2	Extended attribute shared library
libaudit-common	1:2.4-1	Dynamic library for security auditing - common files
libaudit1:amd64	1:2.4-1+b1	Dynamic library for security auditing
libauthen-sasl-perl	2.1600-1	Authen::SASL - SASL Authentication framework
libavahi-client3:amd64	0.6.31-5	Avahi client library
libavahi-common-data:amd64	0.6.31-5	Avahi common data files
libavahi-common3:amd64	0.6.31-5	Avahi common library
libbind9-90	1:9.9.5.dfsg-9+deb8u6	BIND9 Shared Library used by BIND
libblkid1:amd64	2.25.2-6	block device id library
libboost-iostreams1.55.0:amd64	1.55.0+dfsg-3	Boost.Iostreams Library
libbsd0:amd64	0.7.0-2	utility functions from BSD systems - shared library
libbz2-1.0:amd64	1.0.6-7+b3	high-quality block-sorting file compressor library - runtime
libc-bin	2.19-18+deb8u4	GNU C Library: Binaries
libc-dev-bin	2.19-18+deb8u4	GNU C Library: Development binaries
libc6:amd64	2.19-18+deb8u4	GNU C Library: Shared libraries
libc6-dev:amd64	2.19-18+deb8u4	GNU C Library: Development Libraries and Header Files
libcairo2:amd64	1.14.0-2.1+deb8u1	Cairo 2D vector graphics library
libcap-ng0:amd64	0.7.4-2	An alternate POSIX capabilities library
libcap2:amd64	1:2.24-8	POSIX 1003.1e capabilities (library)
libcap2-bin	1:2.24-8	POSIX 1003.1e capabilities (utilities)
libcgi-fast-perl	1:2.04-1	CGI subclass for work with FCGI
libcgi-pm-perl	4.09-1	module for Common Gateway Interface applications
libcilkrts5:amd64	4.9.2-10	Intel Cilk Plus language extensions (runtime)
libclass-accessor-perl	0.34-1	Perl module that automatically generates accessors
libclass-c3-perl	0.26-1	pragma for using the C3 method resolution

Name	Version	Description
		order
libclass-c3-xs-perl	0.13-2+b1	Perl module to accelerate Class::C3
libclass-isa-perl	0.36-5	report the search path for a class's ISA tree
libcloog-isl4:amd64	0.18.2-1+b2	Chunky Loop Generator (runtime library)
libcomerr2:amd64	1.42.12-1.1	common error description library
libcpan-meta-perl	2.142690-1	Perl module to access CPAN distributions metadata
libcryptsetup4:amd64	2:1.6.6-5	disk encryption support - shared library
libcups2:amd64	1.7.5-11+deb8u1	Common UNIX Printing System(tm) - Core library
libcurl3-gnutls:amd64	7.38.0-4+deb8u3	easy-to-use client-side URL transfer library (GnuTLS flavour)
libcwidget3:amd64	0.5.17-2	high-level terminal interface library for C++ (runtime files)
libdaemon0:amd64	0.14-6	lightweight C library for daemons - runtime library
libdata-optlist-perl	0.109-1	module to parse and validate simple name/value option pairs
libdata-section-perl	0.200006-1	module to read chunks of data from a module's DATA section
libdatrie1:amd64	0.2.8-1	Double-array trie library
libdb5.3:amd64	5.3.28-9	Berkeley v5.3 Database Libraries [runtime]
libdbus-1-3:amd64	1.8.20-0+deb8u1	simple interprocess messaging system (library)
libdebconfclient0:amd64	0.192	Debian Configuration Management System (C-implementation library)
libdevmapper1.02.1:amd64	2:1.02.90-2.2+deb8u1	Linux Kernel Device Mapper userspace library
libdiscover2	2.1.2-7	hardware identification library
libdns-export100	1:9.9.5.dfsg-9+deb8u6	Exported DNS Shared Library
libdns100	1:9.9.5.dfsg-9+deb8u6	DNS Shared Library used by BIND
libdpkg-perl	1.17.27	Dpkg perl modules
libedit2:amd64	3.1-20140620-2	BSD editline and history libraries
libencode-locale-perl	1.03-1	utility to determine the locale encoding
libestr0	0.1.9-1.1	Helper functions for handling strings (lib)
libevent-2.0-5:amd64	2.0.21-stable-2	Asynchronous event notification library
libexpat1:amd64	2.1.0-6+deb8u2	XML parsing C library - runtime library
libfakeroot:amd64	1.20.2-1	tool for simulating superuser privileges - shared libraries
libfcgi-perl	0.77-1+b1	helper module for FastCGI
libffi6:amd64	3.1-2+b2	Foreign Function Interface library runtime
libfile-fcntllock-perl	0.22-1+b1	Perl module for file locking with fcntl(2)
libfile-listing-perl	6.04-1	module to parse directory listings
libfont-afm-perl	1.20-1	Font::AFM - Interface to Adobe Font Metrics files
libfontconfig1:amd64	2.11.0-6.3	generic font configuration library - runtime
libfreetype6:amd64	2.5.2-3+deb8u1	FreeType 2 font engine, shared library files
libfuse2:amd64	2.9.3-15+deb8u2	Filesystem in Userspace (library)
libgc1c2:amd64	1:7.2d-6.4	conservative garbage collector for C and C++
libgcc-4.9-dev:amd64	4.9.2-10	GCC support library (development files)
libgcc1:amd64	1:4.9.2-10	GCC support library
libgcrypt20:amd64	1.6.3-2+deb8u1	LGPL Crypto library - runtime library
libgdbm3:amd64	1.8.3-13.1	GNU dbm database routines (runtime

Name	Version	Description
		version)
libgdk-pixbuf2.0-0:amd64	2.31.1-2+deb8u4	GDK Pixbuf library
libgdk-pixbuf2.0-common	2.31.1-2+deb8u4	GDK Pixbuf library - data files
libgeoip1:amd64	1.6.2-4	non-DNS IP-to-country resolver library
libglib2.0-0:amd64	2.42.1-1+b1	GLib library of C routines
libglib2.0-data	2.42.1-1	Common files for GLib library
libgmp10:amd64	2:6.0.0+dfsg-6	Multiprecision arithmetic library
libgnutls-deb0-28:amd64	3.3.8-6+deb8u3	GNU TLS library - main runtime library
libgnutls-openssl27:amd64	3.3.8-6+deb8u3	GNU TLS library - OpenSSL wrapper
libgomp1:amd64	4.9.2-10	GCC OpenMP (GOMP) support library
libgpg-error0:amd64	1.17-3	library for common error values and messages in GnuPG components
libgpgme11:amd64	1.5.1-6	GPGME - GnuPG Made Easy (library)
libgpm2:amd64	1.20.4-6.1+b2	General Purpose Mouse - shared library
libgraphite2-3:amd64	1.3.6-1~deb8u1	Font rendering engine for Complex Scripts -- library
libgssapi-krb5-2:amd64	1.12.1+dfsg-19+deb8u2	MIT Kerberos runtime libraries - krb5 GSS-API Mechanism
libgtk2.0-0:amd64	2.24.25-3+deb8u1	GTK+ graphical user interface library
libgtk2.0-bin	2.24.25-3+deb8u1	programs for the GTK+ graphical user interface library
libgtk2.0-common	2.24.25-3+deb8u1	common files for the GTK+ graphical user interface library
libharfbuzz0b:amd64	0.9.35-2	OpenType text shaping engine (shared library)
libhogweed2:amd64	2.7.1-5+deb8u1	low level cryptographic library (public-key cryptos)
libhtml-form-perl	6.03-1	module that represents an HTML form element
libhtml-format-perl	2.11-1	module for transforming HTML into various formats
libhtml-parser-perl	3.71-1+b3	collection of modules that parse HTML text documents
libhtml-tagset-perl	3.20-2	Data tables pertaining to HTML
libhtml-tree-perl	5.03-1	Perl module to represent and create HTML syntax trees
libhttp-cookies-perl	6.01-1	HTTP cookie jars
libhttp-daemon-perl	6.01-1	simple http server class
libhttp-date-perl	6.02-1	module of date conversion routines
libhttp-message-perl	6.06-1	perl interface to HTTP style messages
libhttp-negotiate-perl	6.00-2	implementation of content negotiation
libicu52:amd64	52.1-8+deb8u3	International Components for Unicode
libidn11:amd64	1.29-1+deb8u1	GNU Libidn library, implementation of IETF IDN specifications
libintl-perl	1.23-1	Uniforum message translations system compatible i18n library
libio-html-perl	1.001-1	open an HTML file with automatic charset detection
libio-socket-ip-perl	0.32-1	module for using IPv4 and IPv6 sockets in a protocol-independent way
libio-socket-ssl-perl	2.002-2+deb8u1	Perl module implementing object oriented interface to SSL sockets

Name	Version	Description
libio-string-perl	1.08-3	Emulate IO::File interface for in-core strings
libirs-export91	1:9.9.5.dfsg-9+deb8u6	Exported IRS Shared Library
libisc-export95	1:9.9.5.dfsg-9+deb8u6	Exported ISC Shared Library
libisc95	1:9.9.5.dfsg-9+deb8u6	ISC Shared Library used by BIND
libisccc90	1:9.9.5.dfsg-9+deb8u6	Command Channel Library used by BIND
libisccfg-export90	1:9.9.5.dfsg-9+deb8u6	Exported ISC CFG Shared Library
libisccfg90	1:9.9.5.dfsg-9+deb8u6	Config File Handling Library used by BIND
libisl10:amd64	0.12.2-2	manipulating sets and relations of integer points bounded by linear cons
libitm1:amd64	4.9.2-10	GNU Transactional Memory Library
libiw30:amd64	30~pre9-8	Wireless tools - library
libjasper1:amd64	1.900.1-debian1-2.4+d	JasPer JPEG-2000 runtime library
libjbig0:amd64	2.1-3.1	JBIGkit libraries
libjpeg62-turbo:amd64	1:1.3.1-12	libjpeg-turbo JPEG runtime library
libjson-c2:amd64	0.11-4	JSON manipulation library - shared library
libk5crypto3:amd64	1.12.1+dfsg-19+deb8u2	MIT Kerberos runtime libraries - Crypto Library
libkeyutils1:amd64	1.5.9-5+b1	Linux Key Management Utilities (library)
libklibc	2.0.4-2	minimal libc subset for use with initramfs
libkmod2:amd64	18-3	libkmod shared library
libkrb5-3:amd64	1.12.1+dfsg-19+deb8u2	MIT Kerberos runtime libraries
libkrb5support0:amd64	1.12.1+dfsg-19+deb8u2	MIT Kerberos runtime libraries - Support library
libksba8:amd64	1.3.2-1+deb8u1	X.509 and CMS support library
liblcms2-2:amd64	2.6-3+b3	Little CMS 2 color management library
libldap-2.4-2:amd64	2.4.40+dfsg-1+deb8u2	OpenLDAP libraries
liblocale-gettext-perl	1.05-8+b1	module using libc functions for internationalization in Perl
liblockfile-bin	1.09-6	support binaries for and cli utilities based on liblockfile
liblockfile1:amd64	1.09-6	NFS-safe locking library
liblog-message-perl	0.8-1	powerful and flexible message logging mechanism
liblog-message-simple-perl	0.10-2	simplified interface to Log::Message
liblogging-stdlog0:amd64	1.0.4-1	easy to use and lightweight logging library
liblognorm1:amd64	1.0.1-3	Log normalizing library
liblsan0:amd64	4.9.2-10	LeakSanitizer -- a memory leak detector (runtime)
liblua5.1-0:amd64	5.1.5-7.1	Shared library for the Lua interpreter version 5.1
liblwp-mediatypes-perl	6.02-1	module to guess media type for a file or a URL
liblwp-protocol-https-perl	6.06-2	HTTPS driver for LWP::UserAgent
liblwres90	1:9.9.5.dfsg-9+deb8u6	Lightweight Resolver Library used by BIND
liblzma5:amd64	5.1.1alpha+20120614-2	XZ-format compression library
liblzo2-2:amd64	2.08-1.2	data compression library
libmagic1:amd64	1:5.22+15-2+deb8u1	File type determination library using "magic" numbers
libmailtools-perl	2.13-1	Manipulate email in perl programs
libmnl0:amd64	1.0.3-5	minimalistic Netlink communication library
libmodule-build-perl	0.421000-2	framework for building and installing Perl modules
libmodule-pluggable-perl	5.1-1	module for giving modules the ability to

Name	Version	Description
		have plugins
libmodule-signature-perl	0.73-1+deb8u2	module to manipulate CPAN SIGNATURE files
libmount1:amd64	2.25.2-6	device mounting library
libmpc3:amd64	1.0.2-1	multiple precision complex floating-point library
libmpfr4:amd64	3.1.2-2	multiple precision floating-point computation
libmro-compat-perl	0.12-1	mro::* interface compatibility for Perls < 5.9.5
libncurses5:amd64	5.9+20140913-1+b1	shared libraries for terminal handling
libncurses5-dev:amd64	5.9+20140913-1+b1	developer's libraries for ncurses
libncursesw5:amd64	5.9+20140913-1+b1	shared libraries for terminal handling (wide character support)
libnet-http-perl	6.07-1	module providing low-level HTTP connection client
libnet-smtp-ssl-perl	1.01-3	Perl module providing SSL support to Net::SMTP
libnet-ssleay-perl	1.65-1+b1	Perl module for Secure Sockets Layer (SSL)
libnetfilter-acct1:amd64	1.0.2-1.1	Netfilter acct library
libnettle4:amd64	2.7.1-5+deb8u1	low level cryptographic library (symmetric and one-way cryptos)
libnewt0.52:amd64	0.52.17-1+b1	Not Erik's Windowing Toolkit - text mode windowing with slang
libnfnetlink0:amd64	1.0.1-3	Netfilter netlink library
libnfsidmap2:amd64	0.25-5	NFS idmapping library
libnl-3-200:amd64	3.2.24-2	library for dealing with netlink sockets
libnl-genl-3-200:amd64	3.2.24-2	library for dealing with netlink sockets - generic netlink
libnuma1:amd64	2.0.10-1	Libraries for controlling NUMA policy
libonig2:amd64	5.9.5-3.2	Oniguruma regular expressions library
libopts25:amd64	1:5.18.4-3	automated option processing library based on autogen
libp11-kit0:amd64	0.20.7-1	Library for loading and coordinating access to PKCS#11 modules - runtime
libpackage-constants-perl	0.04-1	List constants defined in a package
libpam-modules:amd64	1.1.8-3.1+deb8u1+b1	Pluggable Authentication Modules for PAM
libpam-modules-bin	1.1.8-3.1+deb8u1+b1	Pluggable Authentication Modules for PAM - helper binaries
libpam-runtime	1.1.8-3.1+deb8u1	Runtime support for the PAM library
libpam0g:amd64	1.1.8-3.1+deb8u1+b1	Pluggable Authentication Modules library
libpango-1.0-0:amd64	1.36.8-3	Layout and rendering of internationalized text
libpangocairo-1.0-0:amd64	1.36.8-3	Layout and rendering of internationalized text
libpangoft2-1.0-0:amd64	1.36.8-3	Layout and rendering of internationalized text
libpaper-utils	1.1.24+nmu4	library for handling paper characteristics (utilities)
libpaper1:amd64	1.1.24+nmu4	library for handling paper characteristics
libparams-util-perl	1.07-2+b1	Perl extension for simple stand-alone param checking functions
libparse-debianchangelog-perl	1.2.0-1.1	parse Debian changelogs and output them in other formats
libpci3:amd64	1:3.2.1-3	Linux PCI Utilities (shared library)
libpcre3:amd64	2:8.35-3.3+deb8u4	Perl 5 Compatible Regular Expression Library - runtime files

Name	Version	Description
libpcsclite1:amd64	1.8.13-1	Middleware to access a smart card using PC/SC (library)
libperl-dev	5.20.2-3+deb8u6	Perl library: development files
libperl4-corelibs-perl	0.003-1	libraries historically supplied with Perl 4
libperl5.20	5.20.2-3+deb8u6	shared Perl library
libpipeline1:amd64	1.4.0-1	pipeline manipulation library
libpixman-1-0:amd64	0.32.6-3	pixel-manipulation library for X and cairo
libpkcs11-helper1:amd64	1.11-2	library that simplifies the interaction with PKCS#11
libpng12-0:amd64	1.2.50-2+deb8u2	PNG library - runtime
libpod-latex-perl	0.61-1	module to convert Pod data to formatted LaTeX
libpod-readme-perl	0.11-1	Perl module to convert POD to README file
libpopt0:amd64	1.16-10	lib for parsing cmdline parameters
libprocps3:amd64	2:3.3.9-9	library for accessing process information from /proc
libpsl0:amd64	0.5.1-1	Library for Public Suffix List (shared libraries)
libpth20:amd64	2.0.7-20	GNU Portable Threads
libpython-stdlib:amd64	2.7.9-1	interactive high-level object-oriented language (default python version)
libpython2.7-minimal:amd64	2.7.9-2	Minimal subset of the Python language (version 2.7)
libpython2.7-stdlib:amd64	2.7.9-2	Interactive high-level object-oriented language (standard library, versi
libqdbm14	1.8.78-5+b1	QDBM Database Libraries without GDBM wrapper[runtime]
libquadmath0:amd64	4.9.2-10	GCC Quad-Precision Math Library
libreadline6:amd64	6.3-8+b3	GNU readline and history libraries, run-time libraries
libregexp-common-perl	2013031301-1	module with common regular expressions
librtmp1:amd64	2.4+20150115.gita107c	toolkit for RTMP streams (shared library)
libsasl2-2:amd64	2.1.26.dfsg1-13+deb8u	Cyrus SASL - authentication abstraction library
libsasl2-modules:amd64	2.1.26.dfsg1-13+deb8u	Cyrus SASL - pluggable authentication modules
libsasl2-modules-db:amd64	2.1.26.dfsg1-13+deb8u	Cyrus SASL - pluggable authentication modules (DB)
libselinux1:amd64	2.3-2	SELinux runtime shared libraries
libsemanage-common	2.3-1	Common files for SELinux policy management libraries
libsemanage1:amd64	2.3-1+b1	SELinux policy management library
libsensors4:amd64	1:3.3.5-2	library to read temperature/voltage/fan sensors
libsepol1:amd64	2.3-2	SELinux library for manipulating binary security policies
libsigc++-2.0-0c2a:amd64	2.4.0-1	type-safe Signal Framework for C++ - runtime
libsigsegv2:amd64	2.10-4+b1	Library for handling page faults in a portable way
libslang2:amd64	2.3.0-2	S-Lang programming library - runtime version
libsmartcols1:amd64	2.25.2-6	smart column output alignment library

Name	Version	Description
libsnmp-base	5.7.2.1+dfsg-1	SNMP configuration script, MIBs and documentation
libsnmp30:amd64	5.7.2.1+dfsg-1	SNMP (Simple Network Management Protocol) library
libsoftware-license-perl	0.103010-3	module providing templated software licenses
libsqlite3-0:amd64	3.8.7.1-1+deb8u1	SQLite 3 shared library
libsqlite3-dev:amd64	3.8.7.1-1+deb8u1	SQLite 3 development files
libss2:amd64	1.42.12-1.1	command-line interface parsing library
libssh2-1:amd64	1.4.3-4.1+deb8u1	SSH2 client-side library
libssl1.0.0:amd64	1.0.1t-1+deb8u2	Secure Sockets Layer toolkit - shared libraries
libstdc++-4.9-dev:amd64	4.9.2-10	GNU Standard C++ Library v3 (development files)
libstdc++6:amd64	4.9.2-10	GNU Standard C++ Library v3
libsub-exporter-perl	0.986-1	sophisticated exporter for custom-built routines
libsub-install-perl	0.928-1	module for installing subroutines into packages easily
libsub-name-perl	0.12-1	module for assigning a new name to referenced sub
libswitch-perl	2.17-2	switch statement for Perl
libsystemd0:amd64	215-17+deb8u4	systemd utility library
libtasn1-6:amd64	4.2-3+deb8u2	Manage ASN.1 structures (runtime)
libterm-ui-perl	0.42-1	Term::ReadLine UI made easy
libtext-charwidth-perl	0.04-7+b3	get display widths of characters on the terminal
libtext-iconv-perl	1.7-5+b2	converts between character sets in Perl
libtext-soundex-perl	3.4-1+b2	implementation of the soundex algorithm
libtext-template-perl	1.46-1	perl module to process text templates
libtext-unidecode-perl	1.22-1	Text::Unidecode -- US-ASCII transliterations of Unicode text
libtext-wrapi18n-perl	0.06-7	internationalized substitute of Text::Wrap
libthai-data	0.1.21-1	Data files for Thai language support library
libthai0:amd64	0.1.21-1	Thai language support library
libtiff5:amd64	4.0.3-12.3+deb8u1	Tag Image File Format (TIFF) library
libtimedate-perl	2.3000-2	collection of modules to manipulate date/time information
libtinfo-dev:amd64	5.9+20140913-1+b1	developer's library for the low-level terminfo library
libtinfo5:amd64	5.9+20140913-1+b1	shared low-level terminfo library for terminal handling
libtirpc1:amd64	0.2.5-1	transport-independent RPC library
libtokyocabinet9:amd64	1.4.48-3	Tokyo Cabinet Database Libraries [runtime]
libtsan0:amd64	4.9.2-10	ThreadSanitizer -- a Valgrind-based detector of data races (runtime)
libubsan0:amd64	4.9.2-10	UBSan -- undefined behaviour sanitizer (runtime)
libudev1:amd64	215-17+deb8u4	libudev shared library
liburi-perl	1.64-1	module to manipulate and access URI strings
libusb-0.1-4:amd64	2:0.1.12-25	userspace USB programming library
libusb-1.0-0:amd64	2:1.0.19-1	userspace USB programming library
libustr-1.0-1:amd64	1.0.4-3+b2	Micro string library: shared library
libuuid1:amd64	2.25.2-6	Universally Unique ID library

Name	Version	Description
libwebp5:amd64	0.4.1-1.2+b2	Lossy compression of digital photographic images.
libwebpdemux1:amd64	0.4.1-1.2+b2	Lossy compression of digital photographic images.
libwebpmux1:amd64	0.4.1-1.2+b2	Lossy compression of digital photographic images.
libwrap0:amd64	7.6.q-25	Wietse Venema's TCP wrappers library
libwww-perl	6.08-1	simple and consistent interface to the world-wide web
libwww-robotrules-perl	6.01-1	database of robots.txt-derived permissions
libx11-6:amd64	2:1.6.2-3	X11 client-side library
libx11-data	2:1.6.2-3	X11 client-side library
libx86-1:amd64	1.1+ds1-10	x86 real-mode library
libxapian22	1.2.19-1+deb8u1	Search engine library
libxau6:amd64	1:1.0.8-1	X11 authorisation library
libxcb-render0:amd64	1.10-3+b1	X C Binding, render extension
libxcb-shm0:amd64	1.10-3+b1	X C Binding, shm extension
libxcb1:amd64	1.10-3+b1	X C Binding
libxcomposite1:amd64	1:0.4.4-1	X11 Composite extension library
libxcursor1:amd64	1:1.1.14-1+b1	X cursor management library
libxdamage1:amd64	1:1.1.4-2+b1	X11 damaged region extension library
libxdmcp6:amd64	1:1.1.1-1+b1	X11 Display Manager Control Protocol library
libxext6:amd64	2:1.3.3-1	X11 miscellaneous extension library
libxfixes3:amd64	1:5.0.1-2+b2	X11 miscellaneous 'fixes' extension library
libxi6:amd64	2:1.7.4-1+b2	X11 Input extension library
libxinerama1:amd64	2:1.1.3-1+b1	X11 Xinerama extension library
libxml-libxml-perl	2.0116+dfsg-1+deb8u1	Perl interface to the libxml2 library
libxml-namespacesupport-perl	1.11-1	Perl module for supporting simple generic namespaces
libxml-parser-perl	2.41-3	Perl module for parsing XML files
libxml-sax-base-perl	1.07-1	base class for SAX drivers and filters
libxml-sax-expat-perl	0.40-2	Perl module for a SAX2 driver for Expat (XML::Parser)
libxml-sax-perl	0.99+dfsg-2	Perl module for using and building Perl SAX2 XML processors
libxml2:amd64	2.9.1+dfsg1-5+deb8u1	GNOME XML library
libxmuu1:amd64	2:1.1.2-1	X11 miscellaneous micro-utility library
libxrandr2:amd64	2:1.4.2-1+b1	X11 RandR extension library
libxrender1:amd64	1:0.9.8-1+b1	X Rendering Extension client library
libxtables10	1.4.21-2+b1	netfilter xtables library
linux-base	4.3~bpo8+1	Linux image base package
linux-compiler-gcc-4.9-x86	4.6.4-1~bpo8+1	Compiler for Linux on x86 (meta-package)
linux-headers-4.6.0-0.bpo.1-amd64	4.6.4-1~bpo8+1	Header files for Linux 4.6.0-0.bpo.1-amd64
linux-headers-4.6.0-0.bpo.1-commo	4.6.4-1~bpo8+1	Common header files for Linux 4.6.0-0.bpo.1
linux-image-4.6.0-0.bpo.1-amd64	4.6.4-1~bpo8+1	Linux 4.6 for 64-bit PCs
linux-kbuild-4.6	4.6.4-1~bpo8+1	Kbuild infrastructure for Linux 4.6
linux-libc-dev:amd64	4.6.4-1~bpo8+1	Linux support headers for userspace development
lm-sensors	1:3.3.5-2	utilities to read temperature/voltage/fan sensors

Name	Version	Description
locales	2.19-18+deb8u4	GNU C Library: National Language (locale) data [support]
lockfile-progs	0.1.17	Programs for locking and unlocking files and mailboxes
login	1:4.2-3+deb8u1	system login tools
logrotate	3.8.7-1+b1	Log rotation utility
lsb-base	4.1+Debian13+nmu1	Linux Standard Base 4.1 init script functionality
lsb-release	4.1+Debian13+nmu1	Linux Standard Base version reporting utility
lshw	02.17-1.1	information about hardware configuration
lsof	4.86+dfsg-1	Utility to list open files
m4	1.4.17-4	macro processing language
make	4.0-8.1	utility for directing compilation
man-db	2.7.0.2-5	on-line manual pager
manpages	3.74-1	Manual pages about using a GNU/Linux system
manpages-dev	3.74-1	Manual pages about using GNU/Linux for development
mawk	1.3.3-17	a pattern scanning and text processing language
mime-support	3.58	MIME files 'mime.types' & 'mailcap', and support programs
mlocate	0.26-1	quickly find files on the filesystem based on their name
mount	2.25.2-6	Tools for mounting and manipulating filesystems
multiarch-support	2.19-18+deb8u4	Transitional package to ensure multiarch compatibility
mutt	1.5.23-3	text-based mail reader supporting MIME, GPG, PGP and threading
nano	2.2.6-3	small, friendly text editor inspired by Pico
ncurses-base	5.9+20140913-1	basic terminal type definitions
ncurses-bin	5.9+20140913-1+b1	terminal-related programs and man pages
ncurses-term	5.9+20140913-1	additional terminal type definitions
net-tools	1.60-26+b1	NET-3 networking toolkit
netbase	5.3	Basic TCP/IP networking system
netcat-traditional	1.10-41	TCP/IP swiss army knife
nfacct	1.0.1-1.1	netfilter accounting object tool
nfs-common	1:1.2.8-9	NFS support files common to client and server
ntp	1:4.2.6.p5+dfsg-7+deb	Network Time Protocol daemon and utility programs
ntpdate	1:4.2.6.p5+dfsg-7+deb	client for setting system time from NTP servers
opensc	0.14.0-2	Smart card utilities with support for PKCS#15 compatible cards
opensc-pkcs11:amd64	0.14.0-2	Smart card utilities with support for PKCS#15 compatible cards
openssh-client	1:6.7p1-5+deb8u3	secure shell (SSH) client, for secure access to remote machines
openssh-server	1:6.7p1-5+deb8u3	secure shell (SSH) server, for secure access from remote machines
openssh-sftp-server	1:6.7p1-5+deb8u3	secure shell (SSH) sftp server module, for

Name	Version	Description
		SFTP access from remote machines
openssl	1.0.1t-1+deb8u2	Secure Sockets Layer toolkit - cryptographic utility
openvpn	2.3.4-5+deb8u1	virtual private network daemon
os-prober	1.65	utility to detect other OSes on a set of drives
passwd	1:4.2-3+deb8u1	change and administer password and group data
patch	2.7.5-1	Apply a diff file to an original
pciutils	1:3.2.1-3	Linux PCI Utilities
perl	5.20.2-3+deb8u6	Larry Wall's Practical Extraction and Report Language
perl-base	5.20.2-3+deb8u6	minimal Perl system
perl-modules	5.20.2-3+deb8u6	Core Perl modules
php5	5.6.24+dfsg-0+deb8u1	server-side, HTML-embedded scripting language (metapackage)
php5-cli	5.6.24+dfsg-0+deb8u1	command-line interpreter for the php5 scripting language
php5-common	5.6.24+dfsg-0+deb8u1	Common files for packages built from the php5 source
php5-json	1.3.6-1	JSON module for php5
php5-readline	5.6.24+dfsg-0+deb8u1	Readline module for php5
pinentry-gtk2	0.8.3-2	GTK+-2-based PIN or pass-phrase entry dialog for GnuPG
pm-utils	1.4.1-15	utilities and scripts for power management
pmount	0.9.23-3+b1	mount removable devices as normal user
powermgmt-base	1.31+nmu1	Common utils and configs for power management
powertop	2.6.1-1	diagnose issues with power consumption and management
procmail	3.22-24	Versatile e-mail processor
procps	2:3.3.9-9	/proc file system utilities
psmisc	22.21-2	utilities that use the proc file system
python	2.7.9-1	interactive high-level object-oriented language (default version)
python-apt	0.9.3.12	Python interface to libapt-pkg
python-apt-common	0.9.3.12	Python interface to libapt-pkg (locales)
python-chardet	2.3.0-1	universal character encoding detector for Python2
python-debian	0.1.27	Python modules to work with Debian-related data formats
python-debianbts	1.12	Python interface to Debian's Bug Tracking System
python-defusedxml	0.4.1-2	XML bomb protection for Python stdlib modules (for Python 2)
python-docutils	0.12+dfsg-1	text processing system for reStructuredText (implemented in Python 2)
python-minimal	2.7.9-1	minimal subset of the Python language (default version)
python-pil:amd64	2.6.1-2+deb8u2	Python Imaging Library (Pillow fork)
python-pkg-resources	5.5.1-1	Package Discovery and Resource Access using pkg_resources
python-pygments	2.0.1+dfsg-1.1+deb8u1	syntax highlighting package written in Python

Name	Version	Description
python-reportbug	6.6.3	Python modules for interacting with bug tracking systems
python-roman	2.0.0-1	module for generating/analyzing Roman numerals for Python 2
python-six	1.8.0-1	Python 2 and 3 compatibility library (Python 2 interface)
python-soappy	0.12.22-1	SOAP Support for Python
python-support	1.0.15	automated rebuilding support for Python modules
python-wstools	0.4.3-2	WSDL parsing tools Python module
python2.7	2.7.9-2	Interactive high-level object-oriented language (version 2.7)
python2.7-minimal	2.7.9-2	Minimal subset of the Python language (version 2.7)
racoon	1:0.8.2+20140711-2+de	IPsec Internet Key Exchange daemon
read-edid	3.0.1-2	hardware information-gathering tool for VESA PnP monitors
readline-common	6.3-8	GNU readline and history libraries, common files
rename	0.20-3	Perl extension for renaming multiple files
reportbug	6.6.3	reports bugs in the Debian distribution
rpcbind	0.2.1-6+deb8u1	converts RPC program numbers into universal addresses
rsync	3.1.1-3	fast, versatile, remote (and local) file-copying tool
rsyslog	8.4.2-1+deb8u2	reliable system and kernel logging daemon
sed	4.2.2-4+b1	The GNU sed stream editor
sensible-utils	0.0.9	Utilities for sensible alternative selection
sgml-base	1.26+nmu4	SGML infrastructure and SGML catalog file support
shared-mime-info	1.3-1	FreeDesktop.org shared MIME database and spec
snmp	5.7.2.1+dfsg-1	SNMP (Simple Network Management Protocol) applications
snmpd	5.7.2.1+dfsg-1	SNMP (Simple Network Management Protocol) agents
sqlite3	3.8.7.1-1+deb8u1	Command line interface for SQLite 3
ssh	1:6.7p1-5+deb8u3	secure shell client and server (metapackage)
ssl-cert	1.0.35	simple debconf wrapper for OpenSSL
startpar	0.59-3	run processes in parallel and multiplex their output
sudo	1.8.10p3-1+deb8u3	Provide limited super user privileges to specific users
sysstat	11.0.1-1	system performance tools for Linux
systemd	215-17+deb8u4	system and service manager
systemd-sysv	215-17+deb8u4	system and service manager - SysV links
sysv-rc	2.88dsf-59	System-V-like run level change mechanism
sysvinit-utils	2.88dsf-59	System-V-like utilities
tar	1.27.1-2+b1	GNU version of the tar archiving utility
task-english	3.31+deb8u1	General English environment
task-laptop	3.31+deb8u1	laptop

Name	Version	Description
task-ssh-server	3.31+deb8u1	SSH server
tasksel	3.31+deb8u1	tool for selecting tasks for installation on Debian systems
tasksel-data	3.31+deb8u1	official tasks used for installation of Debian systems
tcpd	7.6.q-25	Wietse Venema's TCP wrapper utilities
texinfo	5.2.0.dfsg.1-6	Documentation system for on-line information and printed output
time	1.7-25	GNU time program for measuring CPU resource usage
traceroute	1:2.0.20-2+b1	Traces the route taken by packets over an IPv4/IPv6 network
tzdata	2016d-0+deb8u1	time zone and daylight-saving time data
ucf	3.0030	Update Configuration File(s): preserve user changes to config files
udev	215-17+deb8u4	/dev/ and hotplug management daemon
usbmount	0.0.22	automatically mount and unmount USB mass storage devices
usbutils	1:007-2	Linux USB utilities
util-linux	2.25.2-6	Miscellaneous system utilities
util-linux-locales	2.25.2-6	Locales files for util-linux
vbetool	1.1-3	run real-mode video BIOS code to alter hardware state
vim	2:7.4.488-7	Vi IMproved - enhanced vi editor
vim-common	2:7.4.488-7	Vi IMproved - Common files
vim-runtime	2:7.4.488-7	Vi IMproved - Runtime files
vim-tiny	2:7.4.488-7	Vi IMproved - enhanced vi editor - compact version
w3m	0.5.3-19	WWW browsable pager with excellent tables/frames support
wamerican	7.1-1	American English dictionary words for /usr/share/dict
watchdog	5.14-3	system health checker and software/hardware watchdog handler
wget	1.16-1	retrieves files from the web
whiptail	0.52.17-1+b1	Displays user-friendly dialog boxes from shell scripts
whois	5.2.7	intelligent WHOIS client
wireless-regdb	2014.11.18-1	wireless regulatory database
wireless-tools	30~pre9-8	Tools for manipulating Linux Wireless Extensions
wpasupplicant	2.3-1+deb8u3	client support for WPA and WPA2 (IEEE 802.11i)
xauth	1:1.0.9-1	X authentication utility
xdg-user-dirs	0.15-2	tool to manage well known user directories
xkb-data	2.12-1	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	1:1.2.8.dfsg-2+b1	compression library - runtime