Freescale Semiconductor
Application Note

Document Number: AN3875
Rev. 1, 08/2009

Building a Linux Image and Downloading onto i.MX Processors Using a Virtual Machine

Multimedia Applications Division
Freescale Semiconductor, Inc.
Detroit, MI

A virtual machine (VM) is a software implementation of a machine (computer) that executes programs like a real machine. This application note specifies the following:

- The details of how to create a virtual machine using open source software on a Windows machine.
- Installation of Debian Linux on this virtual machine.
- Installation of Linux Target Image Builder (LTIB) on the virtual machine and set up the host to TFTP files on the target.

Contents

1 Introduction

This document details the following steps using open source tools and Freescale provided board support packages.

- Creating a virtual machine using open source software on a Windows machine
- Installing Debian Linux on this virtual machine
- Installing Linux Target Image Builder (LTIB) on the virtual machine and setting up the host to TFTP files on the target

1.1 Scope

This document is meant to serve as a step-by-step guide for hardware and software engineers to create their own virtual machines, install Linux on it, and create their own Linux image to be downloaded onto i.MX processors.

1.2 Overview

This section includes a brief introduction to virtual machines, discusses different embedded operating systems, and presents how they fit into the i.MX development environment.

1.2.1 Virtual Machines

Virtual machines (VM) allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system. The main advantages of virtual machines are:

- Multiple OS environments can co-exist on the same computer, in strong isolation from each other.
- The virtual machine can provide an instruction set architecture (ISA) that is somewhat different from that of the real machine
- Application provisioning, maintenance, high availability and disaster recovery

Virtual machine converters can help the user resize the hard disk space or clone another virtual machine.

In case the board support package (BSP) installation on the virtual machine is corrupt or the operating system fails, the virtual machine can be reinstalled without affecting the host machine.

The disadvantage of virtual machines might be a slower execution compared to the host machine.

1.2.2 Board Support Package (BSP)

A BSP is implementation-specific support code for a given board, for example, the 3-stack platform for i.MX processors that conforms to a given operating system. It is commonly built with a bootloader that contains the minimal device support to load the operating system and device drivers for all the devices on the board. In order to develop applications for the i.MX processors, users can use the standard Freescale WinCE600 or Freescale Linux BSP for i.MX processors. The application development environment requires building customized Operating System images and downloading them onto the i.MX processor.

1.2.3 Operating System

Here we briefly discuss the pros and cons of Embedded Linux and Windows CE.

1.2.3.1 Embedded Linux

Linux is part of the Unix-like operating systems. It was initially developed in 1991 for IBM-compatible personal computer. Over the years Linux has become available for other architectures including Power Architecture® and ARM® technologies. Among the major benefits about Linux stands the fact that it isn’t a commercial operating system since its source code is available under the GPL (GNU General Public License), this means that the code is open source.

Linux is a modular operating system that uses a monolithic kernel. The Linux kernel takes care of the process control, networking, peripheral and file system access. However Linux does not covers applications like file-system utilities or windowing systems and graphical desktops, text editors, system administrators. All these utilities are provided freely under the GPL license too and they can be installed in Linux-based systems.

LTIB stands for Linux Target Image Builder and enables the user to customize the OS image for the i.MX processors..

1.2.3.2 Windows CE

Windows Embedded CE stands for Windows Embedded Consumer Electronics. It is a compact, modular, 32-bit, real-time operating system designed for use on devices with small memory requirements (small footprint devices). Windows Embedded CE is a multitasking, multithreaded OS.

Windows Embedded CE is based on the Win32 API from Microsoft, so programmers that are familiar with programming for traditional Windows platforms like Windows NT can begin programming on Windows Embedded CE applications with very little additional training. The Windows Embedded CE OS is independent of the hardware architecture, and can therefore be ported to a wide variety of CPUs, including ARM processors.

1.2.4 System Description

A personal computer with the following components is required for the procedures described in this document :

- Software
- Windows installed
- Hyperterminal
- Hardware
- Ethernet with internet connectivity
- Serial connector
- CD / USB
- i.MX 3-stack development system / product development kit

Figure 1 shows the various elements of the development environment. The 3-stack platform consists of the CPU card, Personality card and the Debug board. The CPU card has the i.MX processor, the power management chip and memory. The personality card has most of the other peripherals including the display. The debug card has the Ethernet and serial ports which are used to interface with the host machine – in our case the personal computer with virtual machine.

The virtual machine will be installed on a PC with Windows and needs an application such as VM Player to run. LTIB is then installed on the VIRTUAL MACHINE which has Linux installed on it. The Linux image for the ARM cores on the i.MX processors can be ported to the i.MX 3-stack platform by using the Ethernet connection. The Hyperterminal receives the debug messages from the i.MX processor using the serial port.

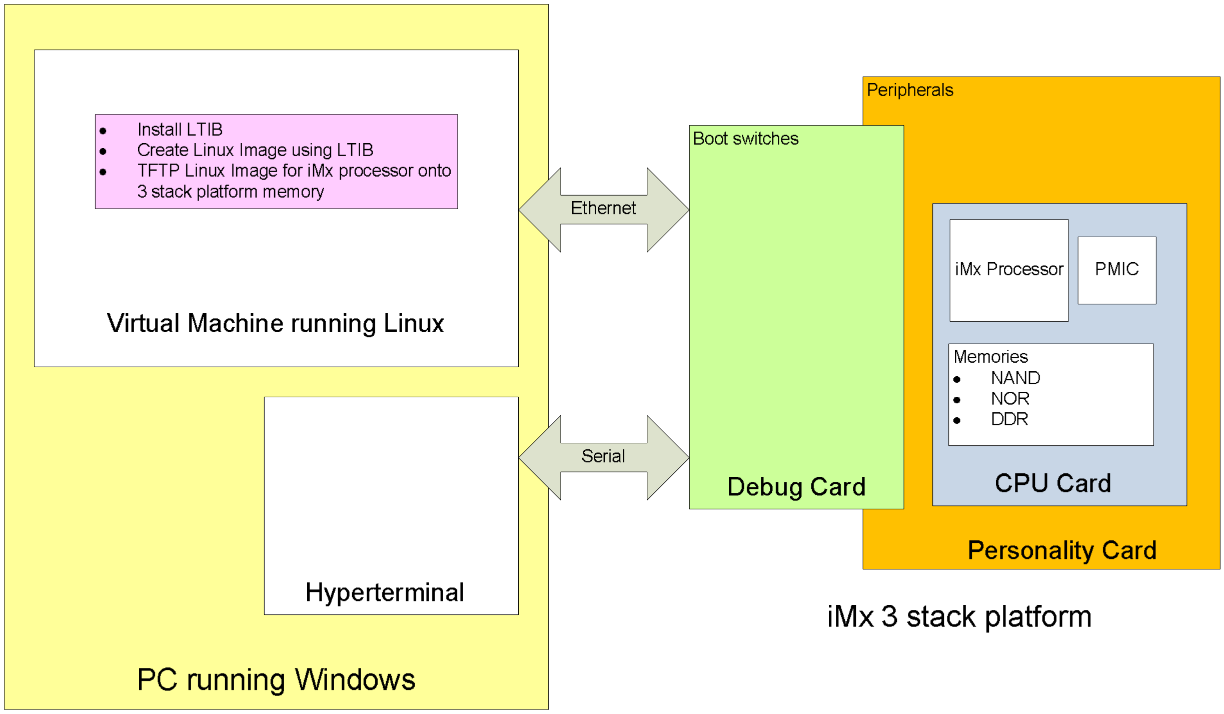


Figure 1. Block Diagram

Detailed documentation describing each of the above mentioned sections can be found in [5].

The development environment for this document has been created on a Windows XP Professional Edition machine. For users installing a different operating system in the virtual machine, differences are noted in the relevant sections.

2 Creating a Virtual Machine

In this section, instructions to create a virtual machine on the PC running Windows are presented. The first step is to create a virtual machine for VMware® Player. VMware Player is a free application which can run multiple operating systems simultaneously on a single PC. Using a web browser, navigate to www.easyvmx.com and the webpage shown in Figure 2 appears.

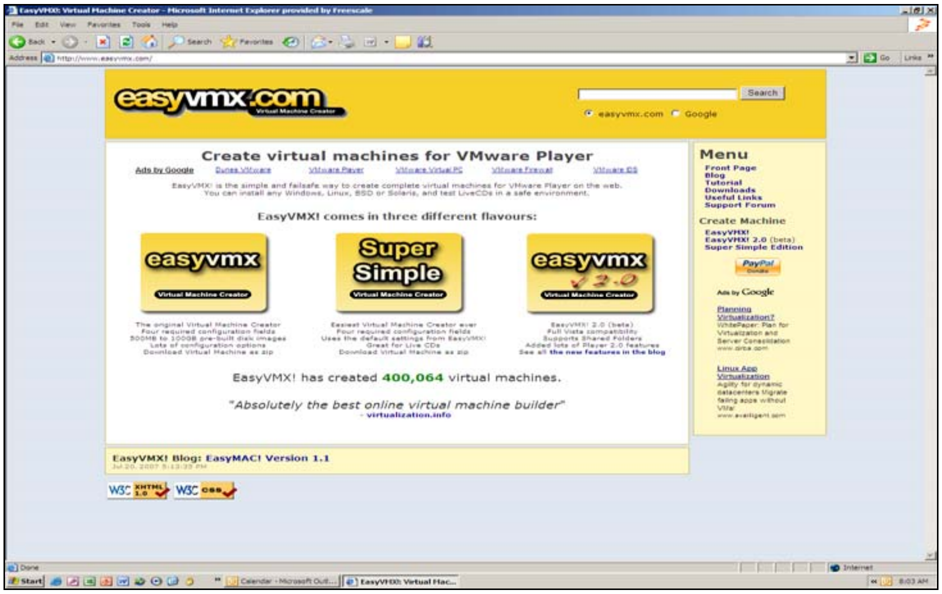


Figure 2. Easyvmx Website Screenshot

Click on the  button in Figure 2 and a new webpage, “Basic Configuration,” appears on the screen.

The following subsections describe configuration of the virtual machine’s parameters such as the name assigned to it, the hard disk space, networking device, sound and I/O controls, as well as the steps to download this virtual machine onto the personal computer.

2.1 Virtual Hardware

This document is meant to serve as a step by step guide for hardware and software engineers to create their own virtual machines, install Linux on it and create their own Linux image to be downloaded onto i.MX processors.

Under the option "Virtual Hardware," create a name, for example, WinCE_iMXmn or LTIB_iMXmn, based on the OS to be installed in the virtual machine.

CAUTION

For users with a different operating system make sure to choose the one that best describes the operating system in the “Select GuestOS” spin box as shown in Figure 3.

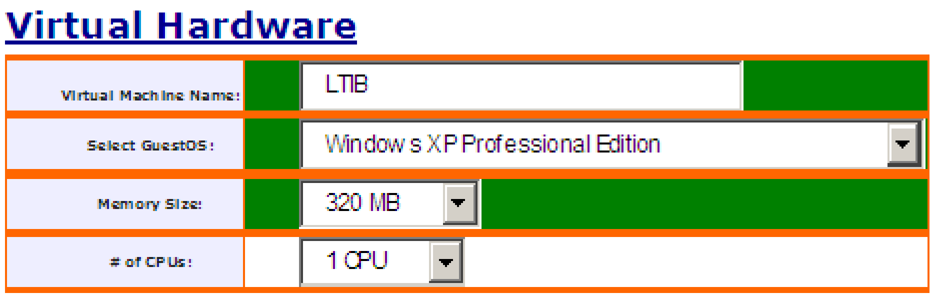


Figure 3. Virtual Machine Name

2.2 Hard Disk Drives

Under the option hard disk drive to allow the virtual machine to use up to 10 GB of hard disk space, select “Disk Size 10GB” from the spin box “Disk #1:” as shown in Figure 4. Note that, for WinCE users, more than 40 GB might be a more appropriate disk space.

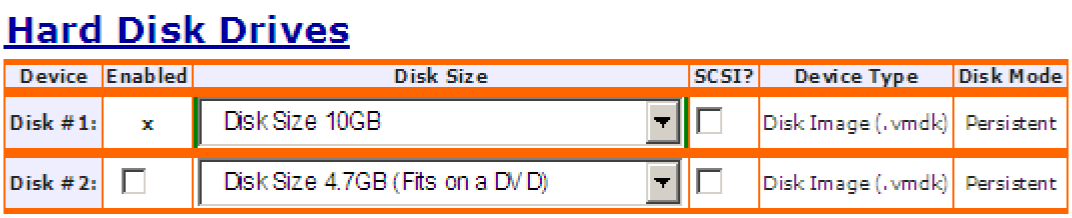


Figure 4. Virtual Machine Hard Disk Size Configuration

2.3 Network Configuration

Use the default network settings as shown in Figure 5.

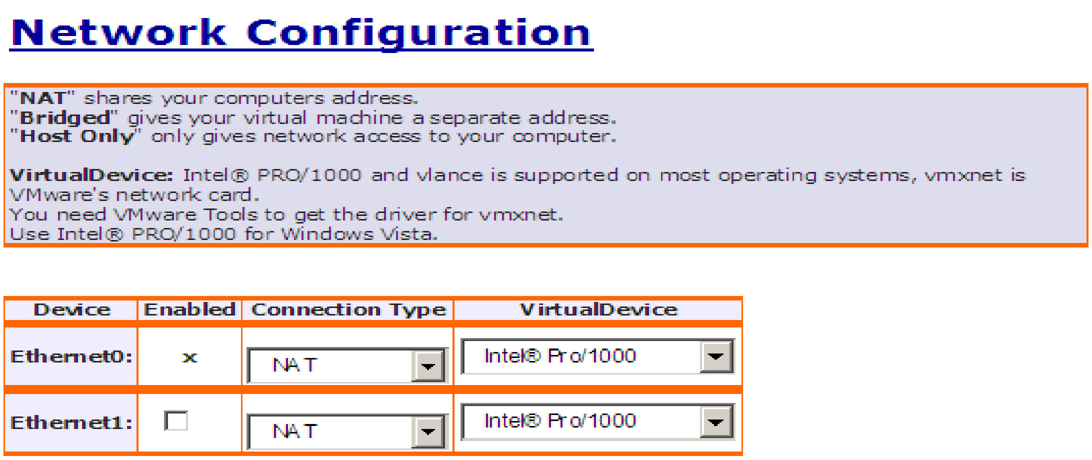


Figure 5. Virtual Machine Network Configuration

2.4 Sound and I/O Port Configuration

To enable USB auto connect on the virtual machine, select “Enable USB Autoconnect” option in the spin box “USB:” as shown in Figure 6 under the option Sound and I/O-Ports Configuration. To enable audio, the appropriate sound card in the system should be selected.

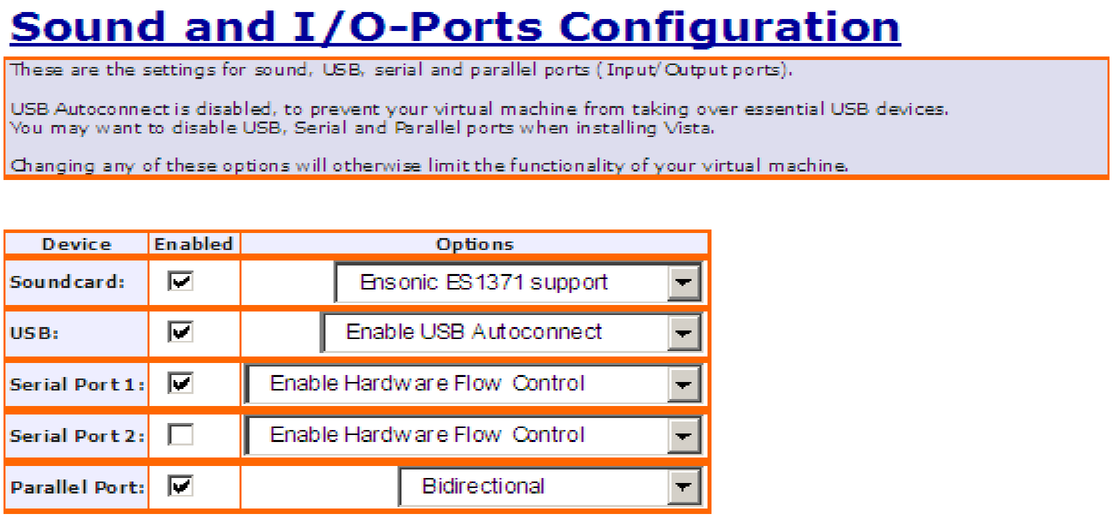
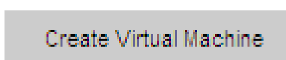


Figure 6. Virtual Machine Sound and I/O Configuration

Click on the button 

and a new webpage appears as shown in Figure 7.

2.5 Download Virtual Machine

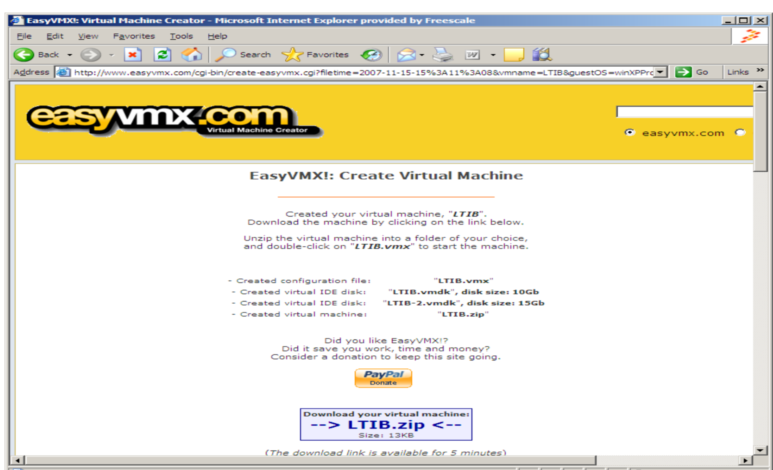


Figure 7. Download Virtual Machine

Click on the button 

to download the virtual machine. Save it on the hard disk and unzip it in a folder whose location is not going to be changed. This creates the virtual machine. Next, proceed to download the VMware Player.

3 Installing VM Player

The virtual machine which was configured in the previous section can be executed using the VM Player which can be downloaded from the VMware website. The appended webpage appears as shown in Figure 8.

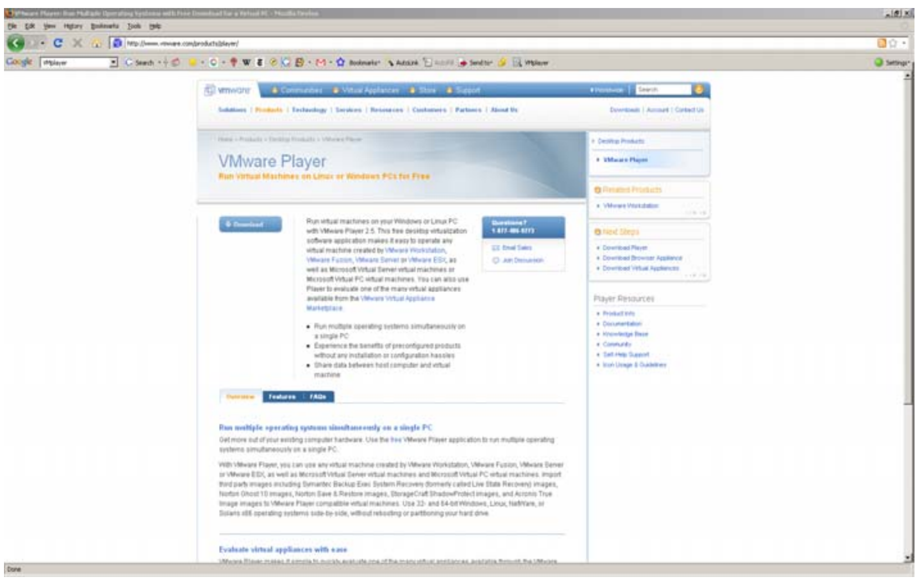


Figure 8. VMware Website Screenshot

The following subsections lead the user through installation of the VM Player on the personal computer.

3.1 Download VM Player

Click the “Download Now” button and follow the steps as directed by the subsequent web pages. Download the `VMware-player-2.5.2.exe` and save it onto the hard disk drive. See Figure 9.

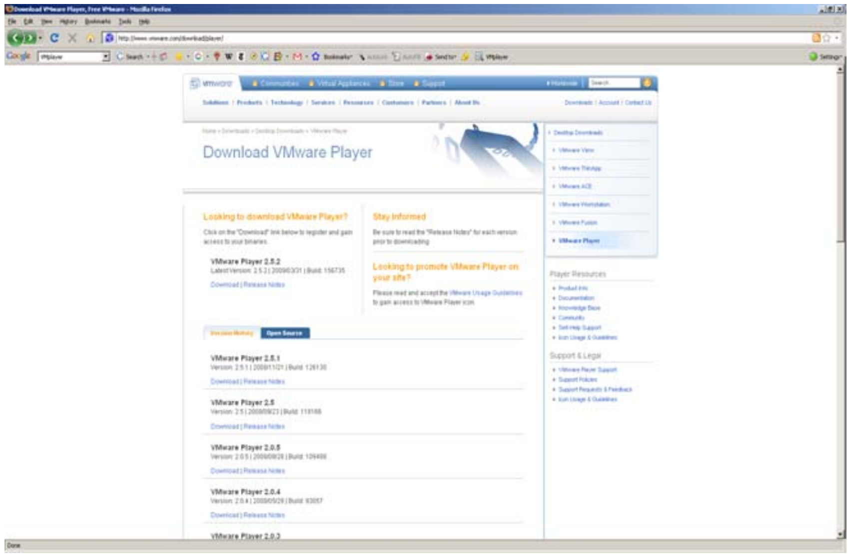


Figure 9. VM Player Releases Website Screenshot

CAUTION

The version of the VMware Player may be changed. Download this or any later version.

3.2 Install VM Player

Double click on the downloaded file to start the installation process. Next, click the "Run" button to install VMware Player as shown in Figure 10.

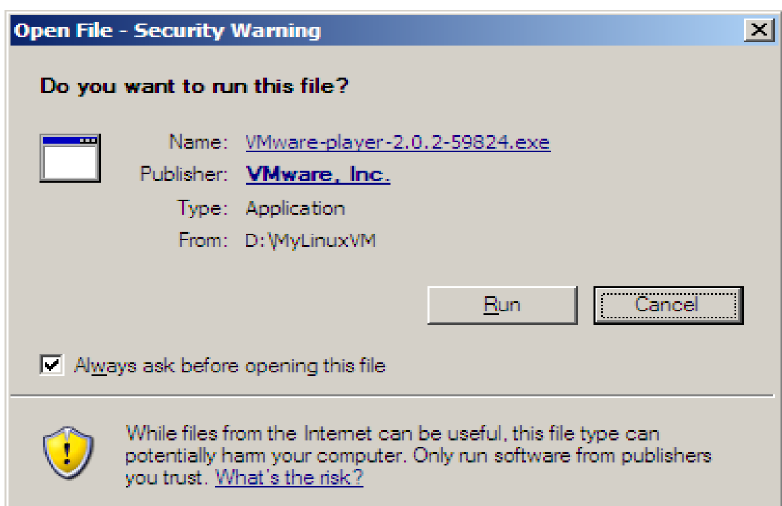


Figure 10. VM Player Download

The preparation for installation takes less than 30 seconds, as shown in Figure 11. If the process takes longer than two minutes, verify the integrity of the download file or simply download the VMware Player again.

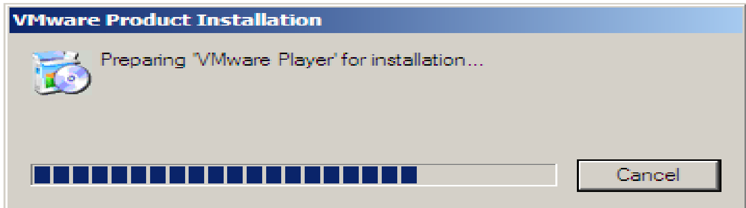


Figure 11. VM Player Download Ongoing

Once the preparation process is successful, a new window appears as shown in Figure 12.

Figure 12. VM Player Installation

Click the "Next>" button in this window, and a "Destination Folder" window appears, as shown in Figure 13.

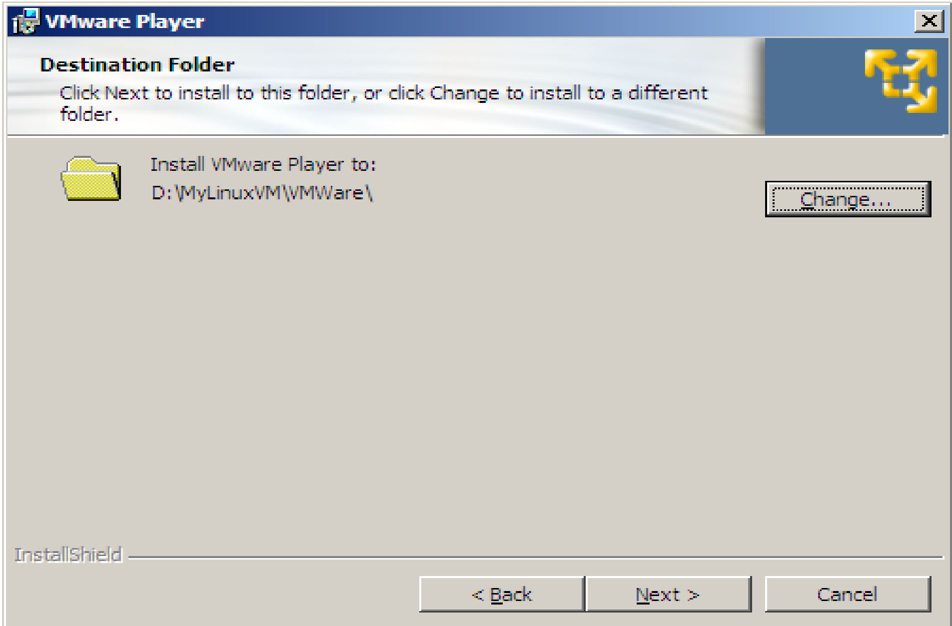


Figure 13. Virtual Machine Installation Folder

Choose the desired destination folder by clicking on the “Change” button in the “Destination Folder” window. Once the destination folder is selected, click the “Next>” button and the “Configure Shortcuts” window appears, as shown in Figure 14.

Check “Desktop” and then click the “Next >” button in the “Configure Shortcuts” window.

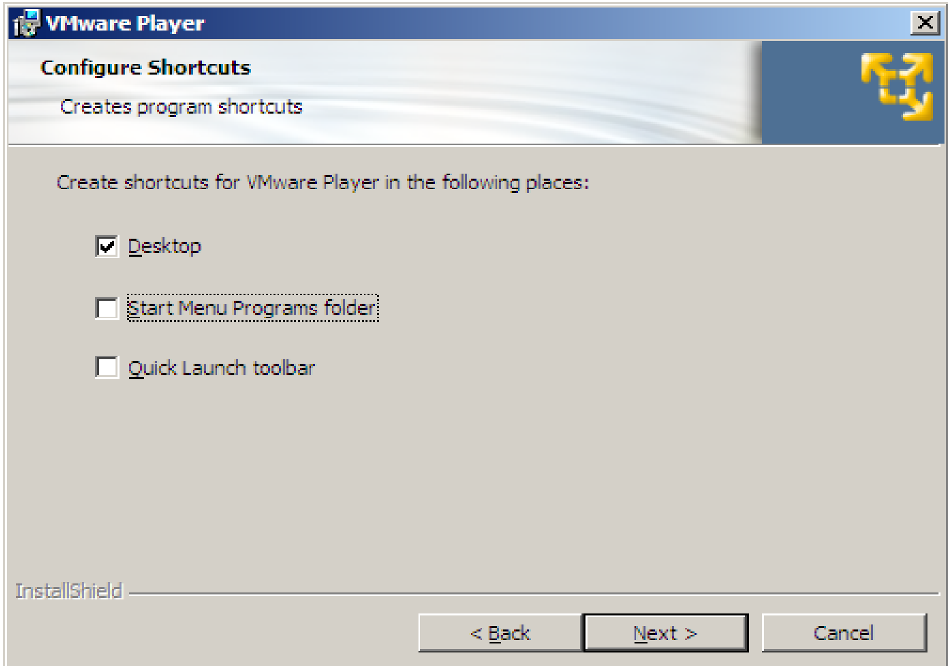


Figure 14. VM Player Shortcuts Configuration

VMware Player proceeds to install all the necessary components onto the computer, as shown in Figure 15.

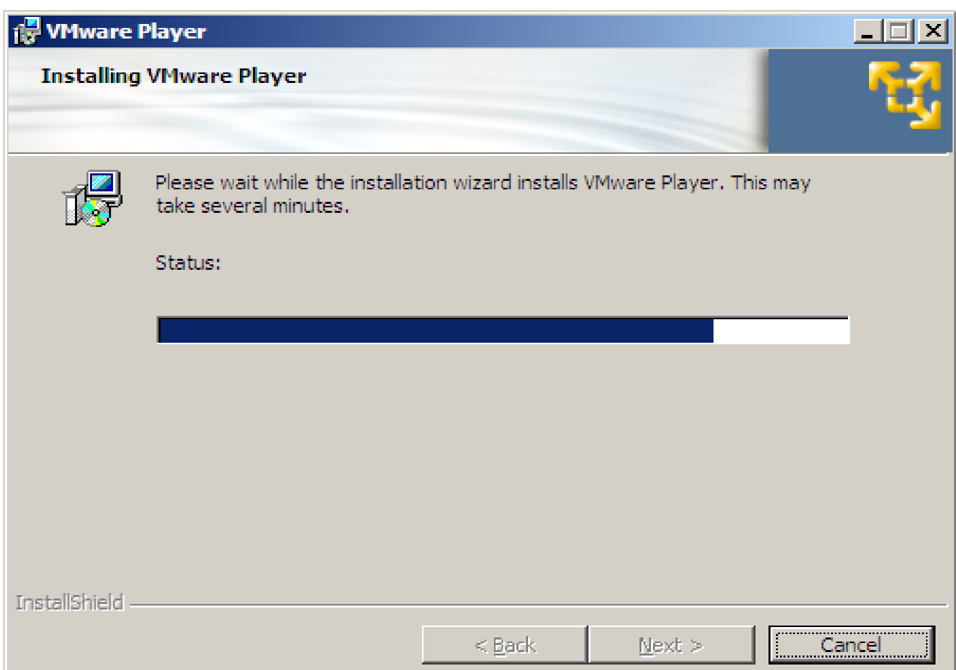


Figure 15. VM Player Installation Ongoing

After the installation process is done, the “Installation Wizard Completed” window appears, as shown in Figure 16. Click on the “Finish” button and reboot the system to end this process.

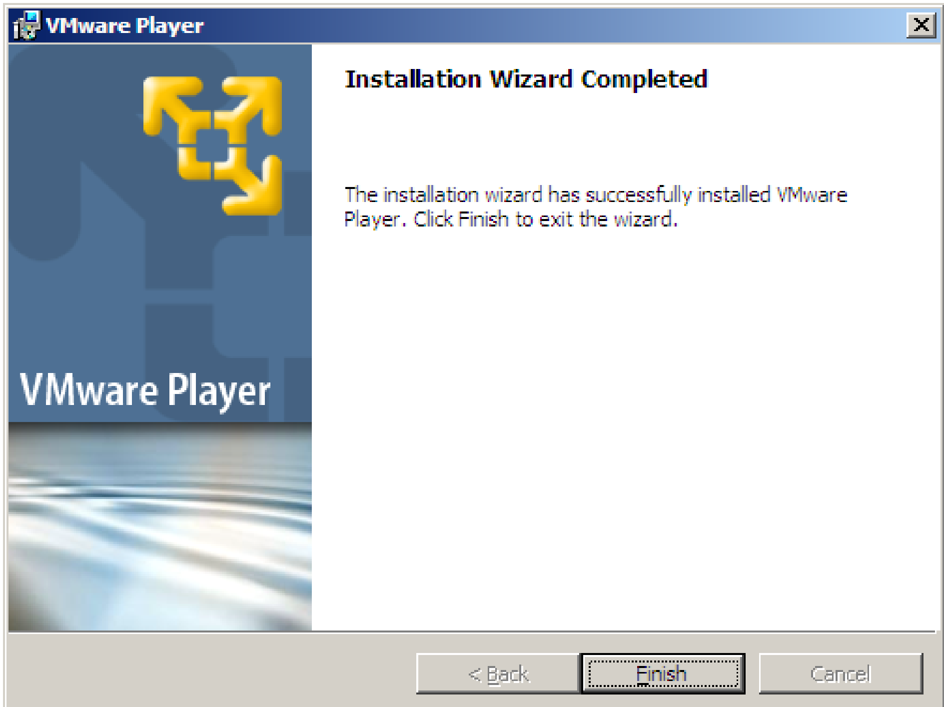


Figure 16. VM Player Installation Completed

4 Installing Debian in the Virtual Machine

Technical variations of Linux distributions include support for different hardware devices and systems or software package configurations. Organizational differences may be motivated by historical reasons. Other criteria include security, including how quickly security upgrades are available; ease of package management; and number of packages available. A good comparison of the different flavors of Linux can be found it in [6].

The following section leads the user through installing Debian Linux onto the virtual machine. Instructions are similar for any other Linux distribution. For users installing WinCE, Windows XP should be installed instead of Debian Linux onto the virtual machine.

This is a lengthy process which needs the Debian net installation CD. The Debian CD can be downloaded from www.debian.org in an ISO image format which has to be burnt onto a DVD. Version 4.0 or later is sufficient.

Start the VMware Player by clicking on the VMware Player shortcut on the desktop. A VMware Player window appears, as shown in Figure 17. Click on the “Open” icon and select the folder where of the unzipped virtual machine.

Figure 17. Start Virtual Machine Using VM Player

The following commands can be used at any point:

- To restart the virtual machine, 'CTRL + ALT + Insert' should be pressed simultaneously.
- To direct the input to the virtual machine from the Windows environment, 'CTRL + G' should be pressed simultaneously.
- To revert back to the Windows environment, 'CTRL + ALT' should be pressed simultaneously.

The following subsections enumerate the steps to install Linux on the virtual machine.

4.1 Open Virtual Machine

Select the `LTIB.vmx` file. The name of the virtual machine may vary depending on what name was provided during the creation of the virtual machine. Nonetheless, it still bears the file extension of "vmx." Insert the Debian net installation CD into the CDROM and click the "Open" button on the Open Virtual Machine window. See Figure 18.

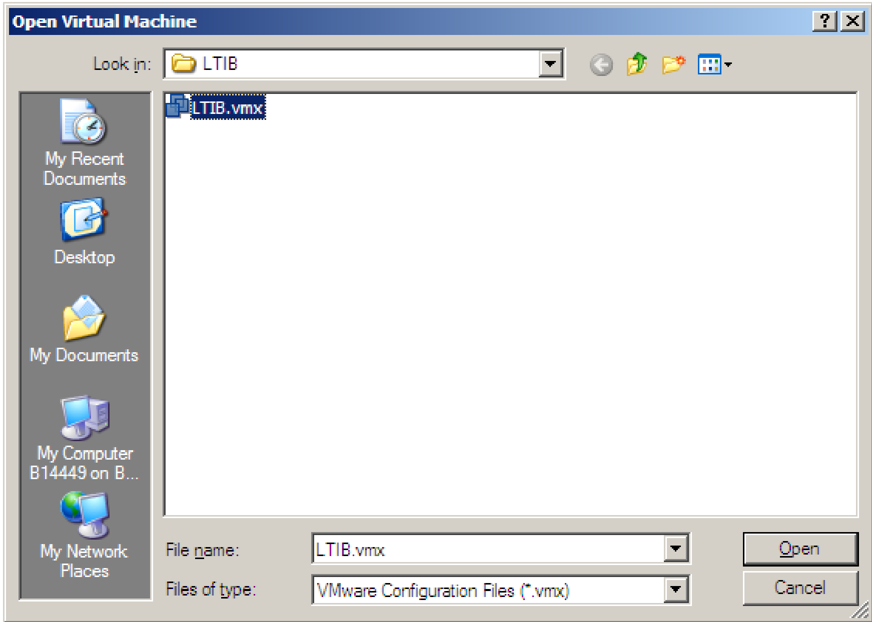


Figure 18. Selection of Virtual Machine Using VM Player

The Debian net installation begins now, as shown in Figure 19.

Figure 19. Installation of Debian Linux on Virtual Machine

4.2 Installing Linux

Linux installation begins with a startup screen as shown in Figure 20 (the VMware logo appears in the foreground while the OS is booting up in the background). Place the mouse cursor over in the VMware Player window and right click the mouse button to pass control to VMware Player. Press the “Enter” key, and the Linux installation proceeds.

Figure 20. Debian Linux Installation on Virtual Machine

Figure 21, Figure 22, and Figure 23 show, in order, the subsequent windows asking the user to enter configuration information. Choose the appropriate field using the up or down arrow key and press the “Enter” key to go to the next window.

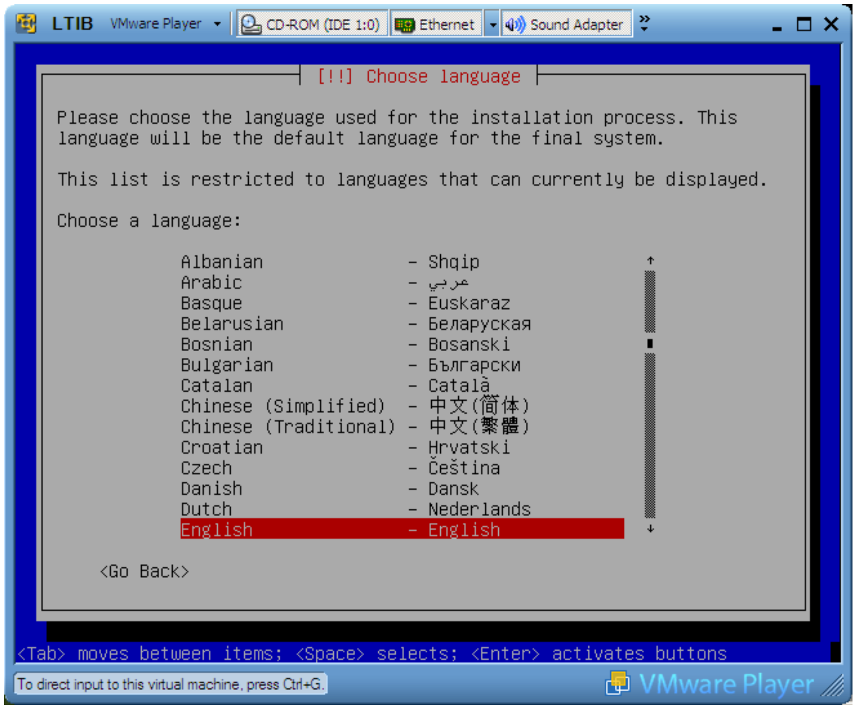


Figure 21. Language Selection

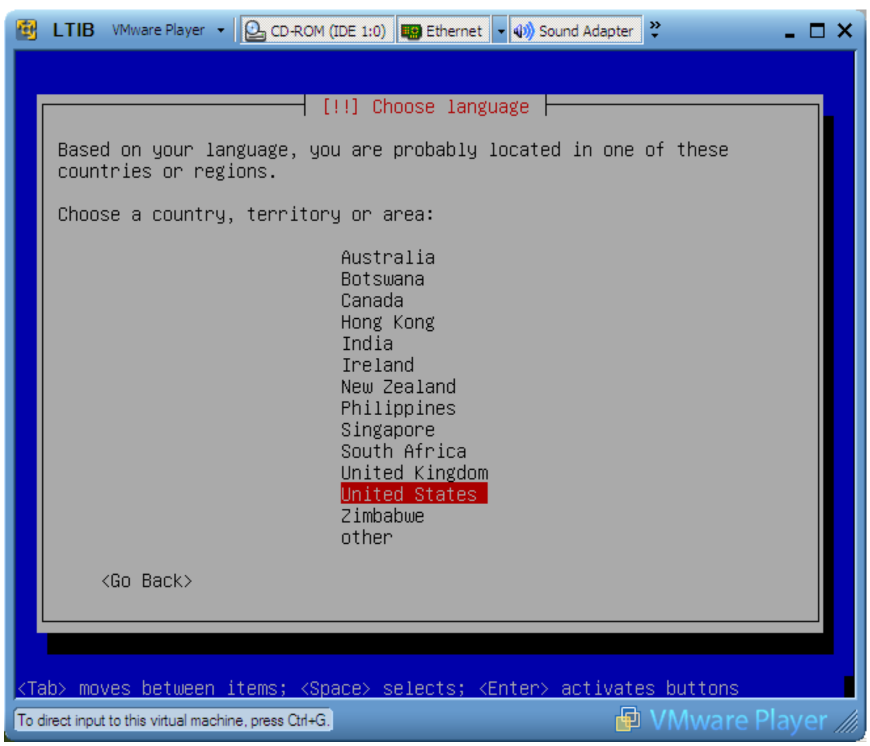


Figure 22. Region Selection

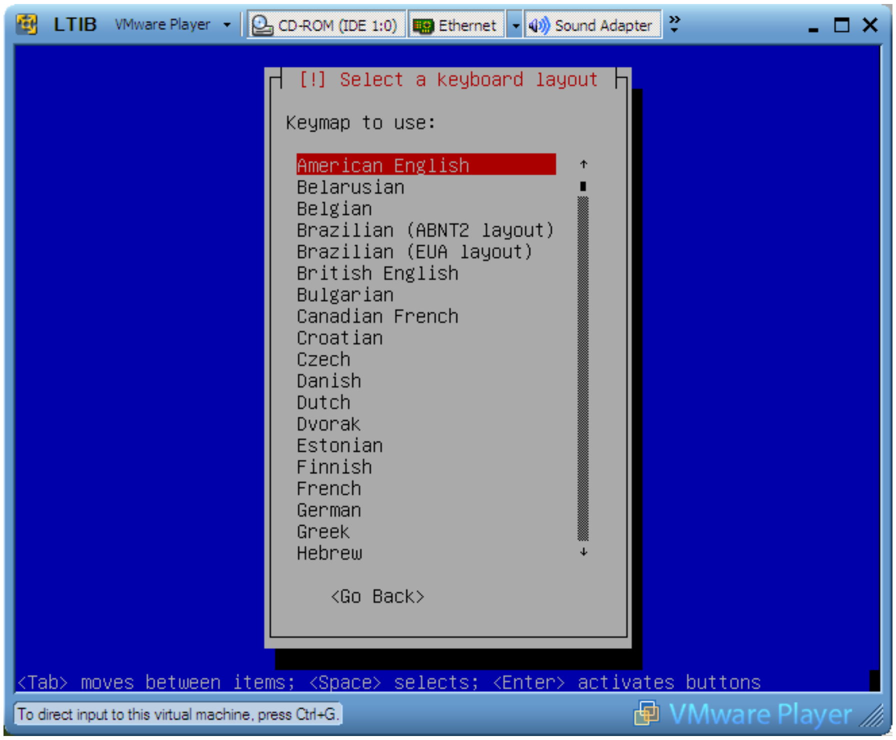


Figure 23. Keyboard Layout Selection

The installation wizard scans the installation CD and loads additional components as shown in Figure 24 and Figure 25.

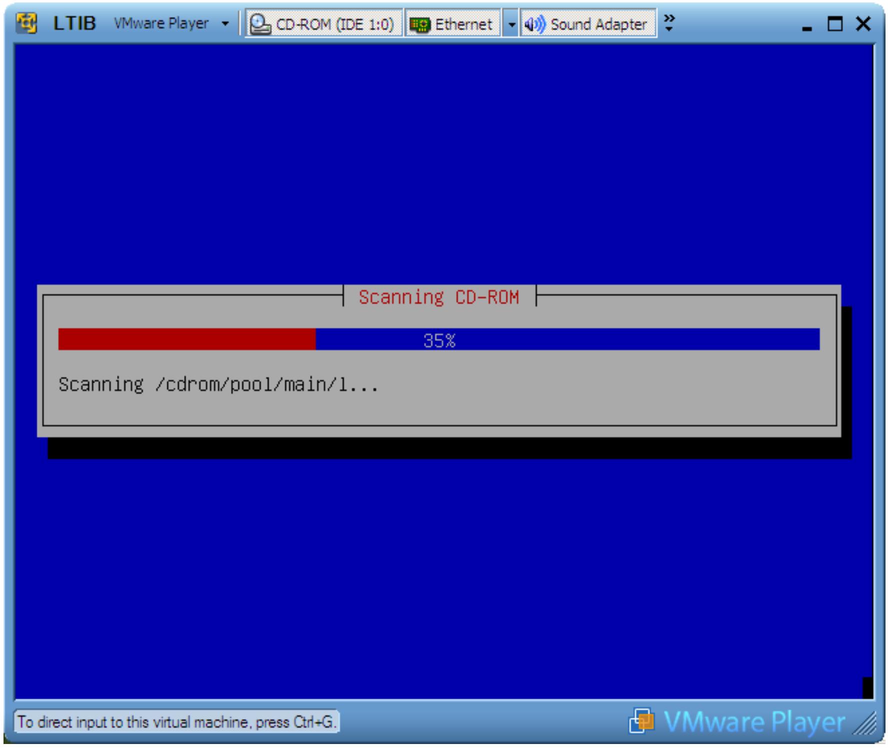


Figure 24. Debian Installation from CD

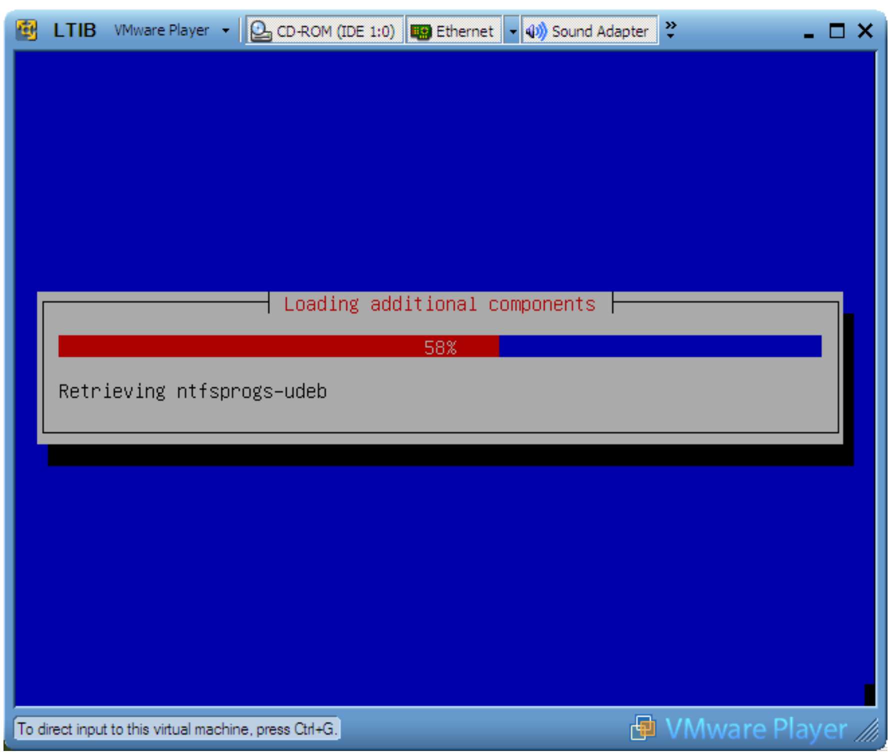


Figure 25. Loading Linux Components from CD

This is the network configuration stage. Enter a unique hostname and local domain for this system in each window and press the “Enter” key on the keyboard. See Figure 26 and Figure 27.

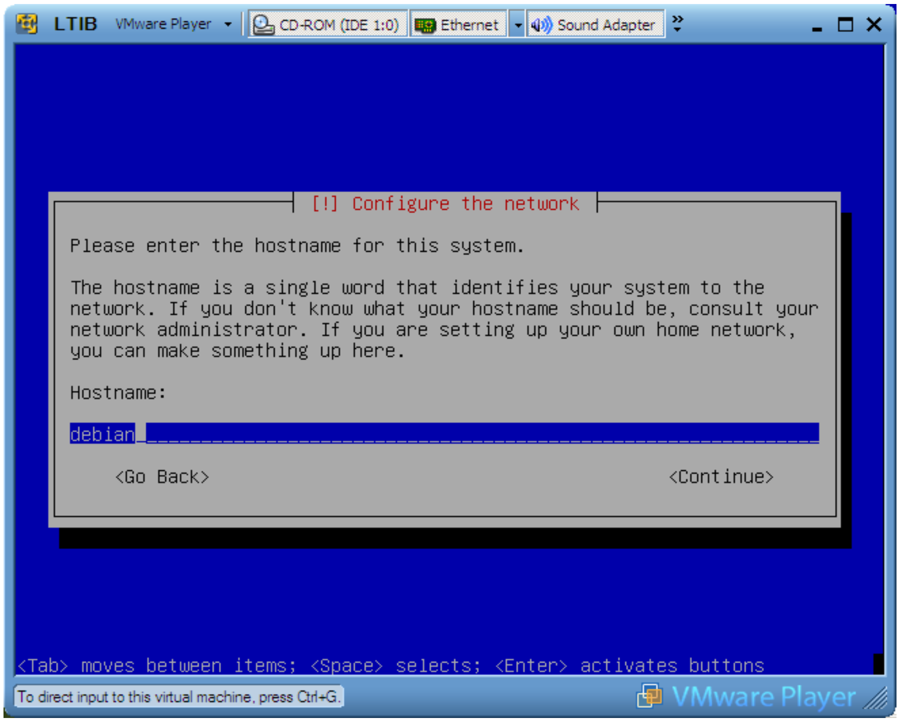


Figure 26. Network Configuration—Host Name

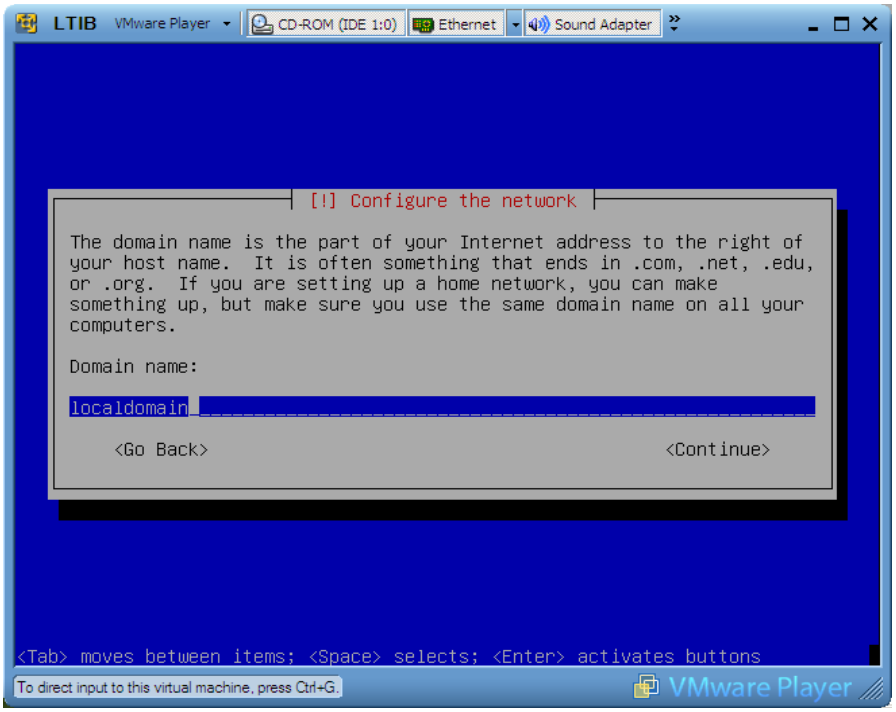


Figure 27. Network Configuration—Domain Name

The disk partition stage follows the network configuration stage. Select the options as shown from Figure 28 through Figure 32.

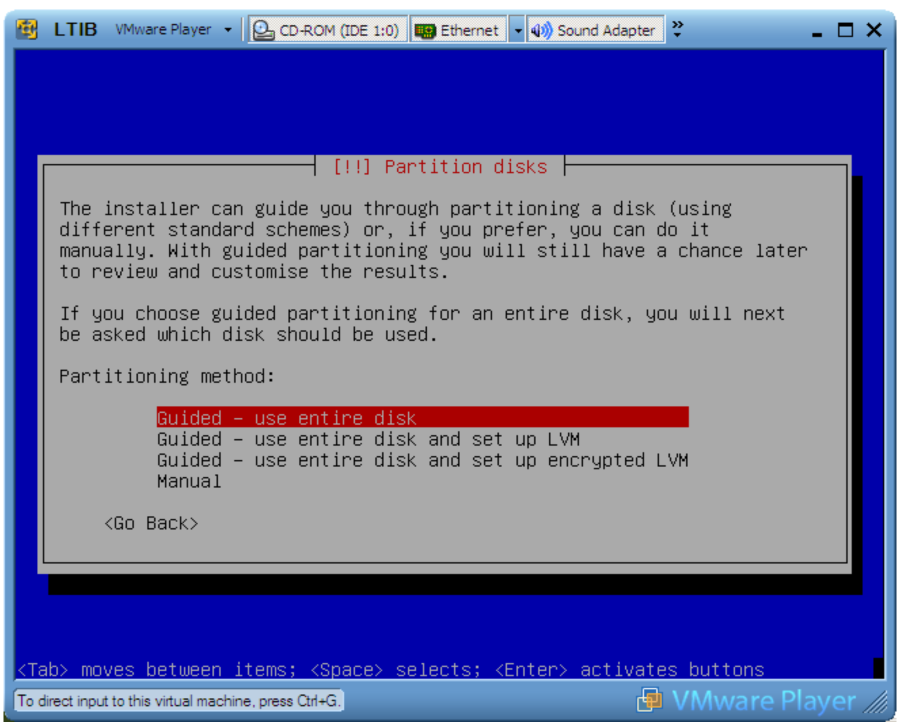


Figure 28. Disk Partitioning Guided – use entire disk

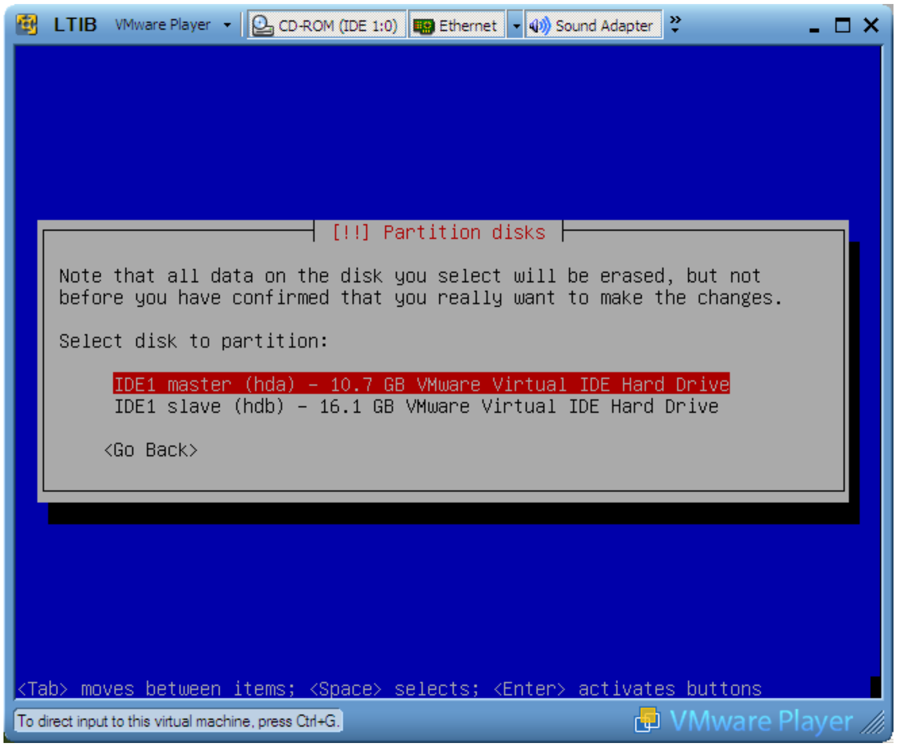


Figure 29. Disk Partitioning IDE master (hda) – 10.7 GB VMware IDE Hard Drive

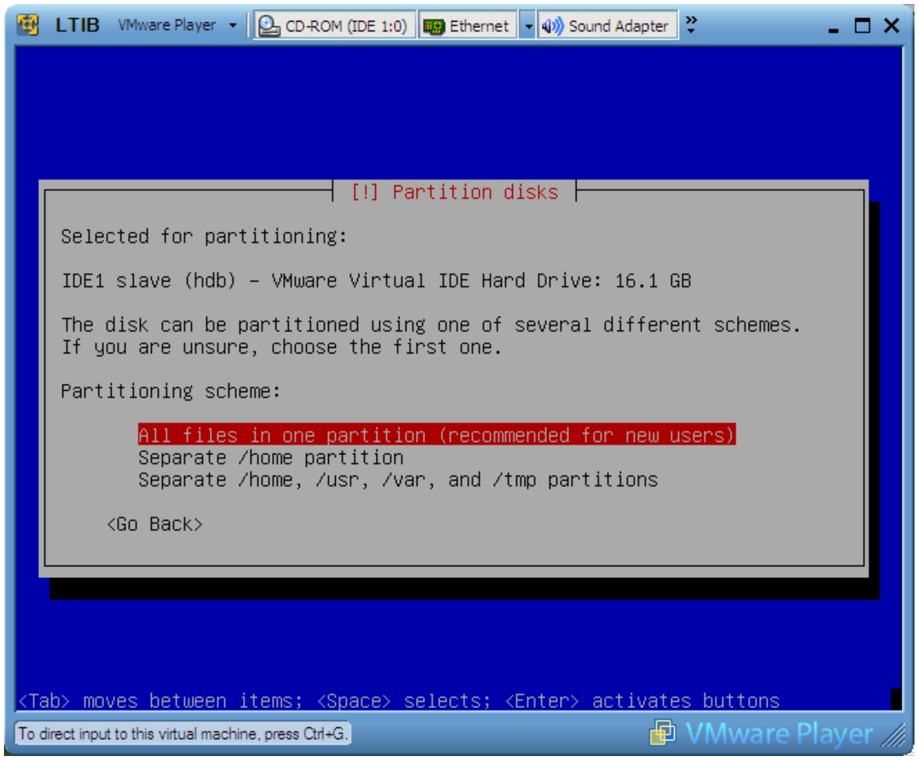


Figure 30. Disk Partitioning All files in one partition (recommended for new users)

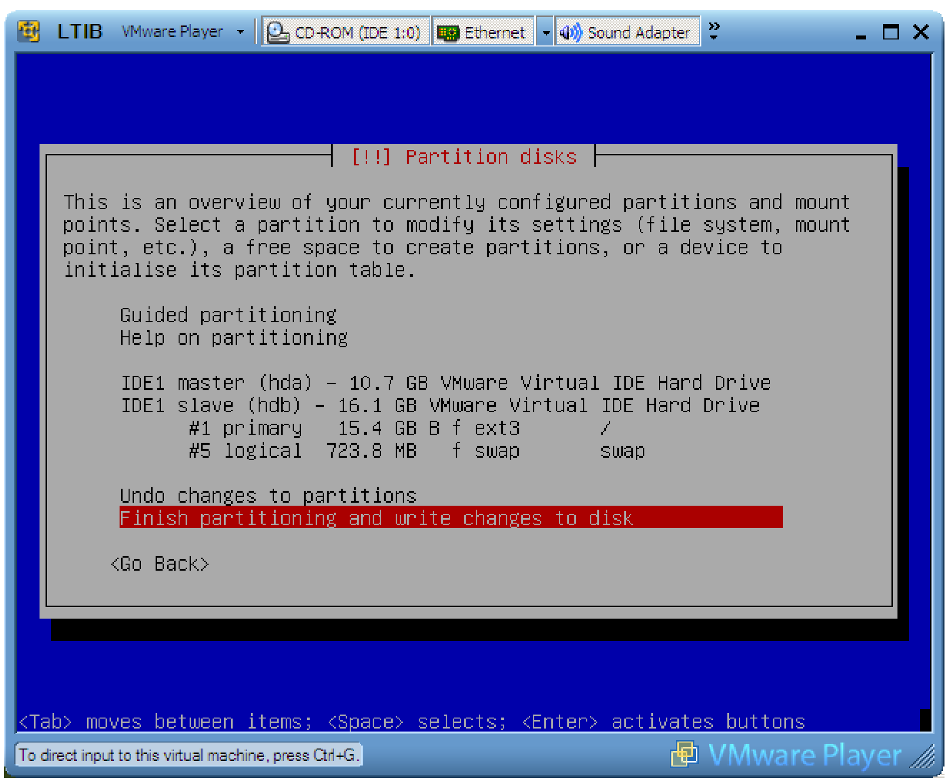


Figure 31. Disk Partitioning Finish Partitioning and Write Changes to Disk

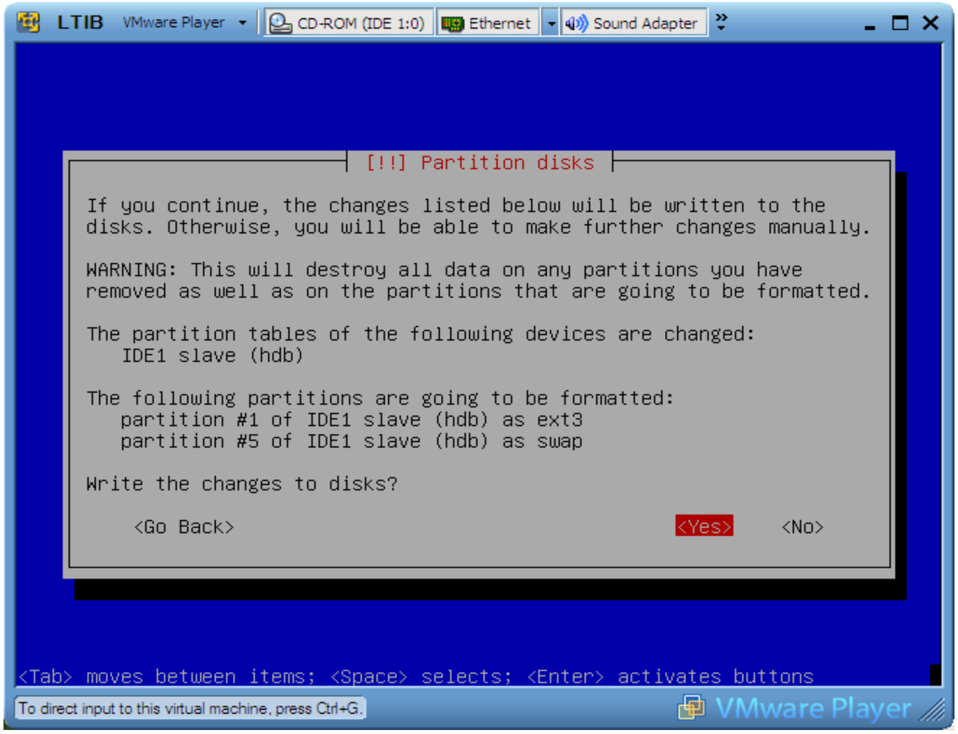


Figure 32. Disk Partitioning Yes

4.3 Setup Time Zone

Select the appropriate time zone with the up or down arrow key and then press the “Enter” key. See Figure 33.

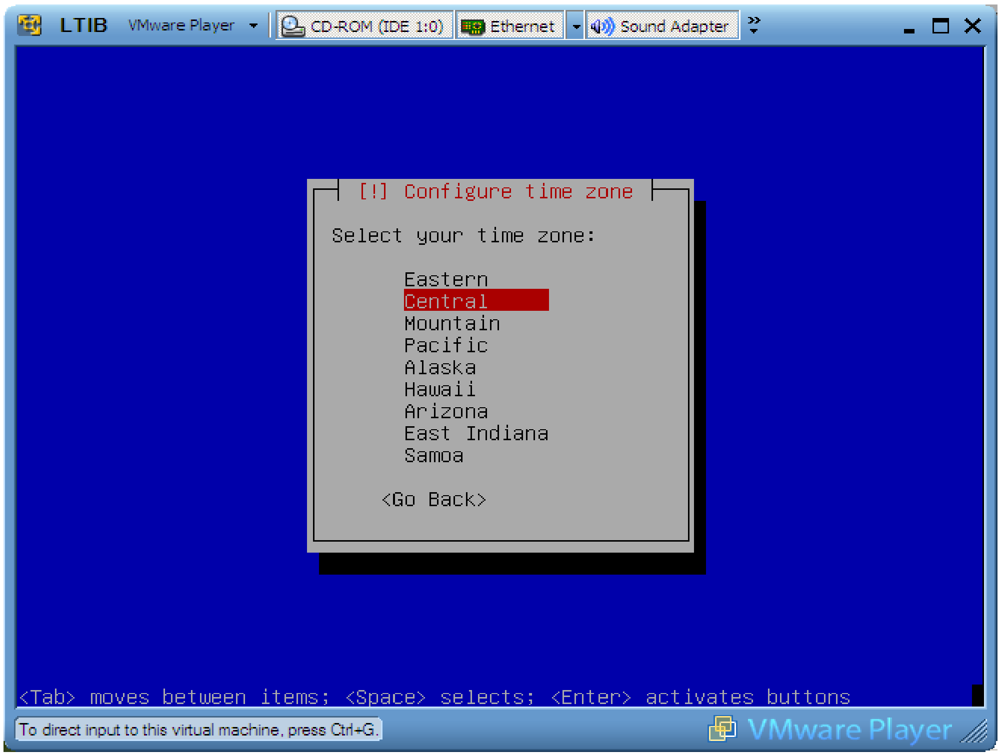


Figure 33. Time Zone Configuration

4.4 Setup Users and Passwords

The next six windows, shown in Figure 34 to Figure 39, show the set-up of root and user passwords and the login name. Make a note of the root and user password. These passwords are required later to login to the system and upgrade user status to super user status.

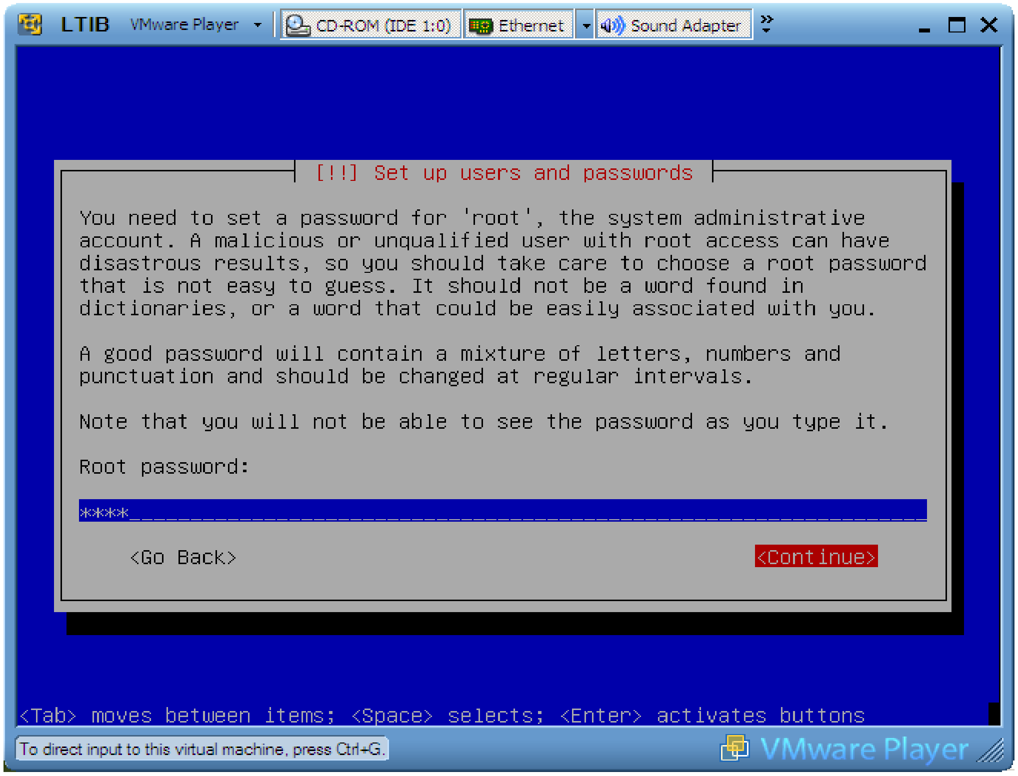


Figure 34. Root User Configuration

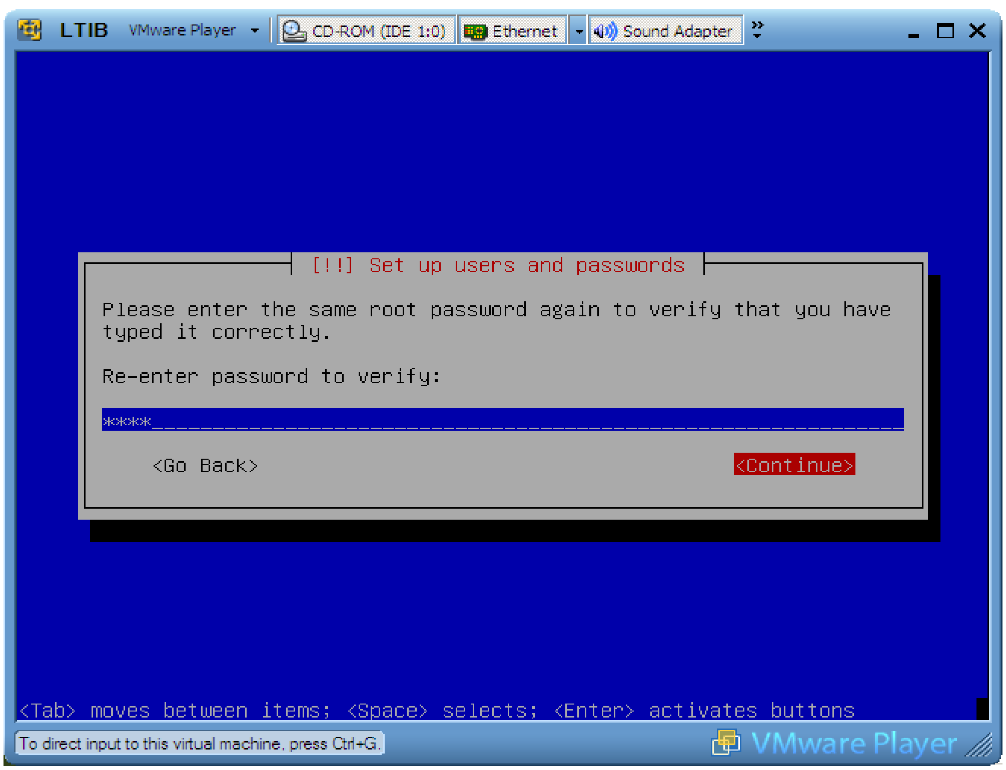


Figure 35. Root User Configuration—Re-enter Password

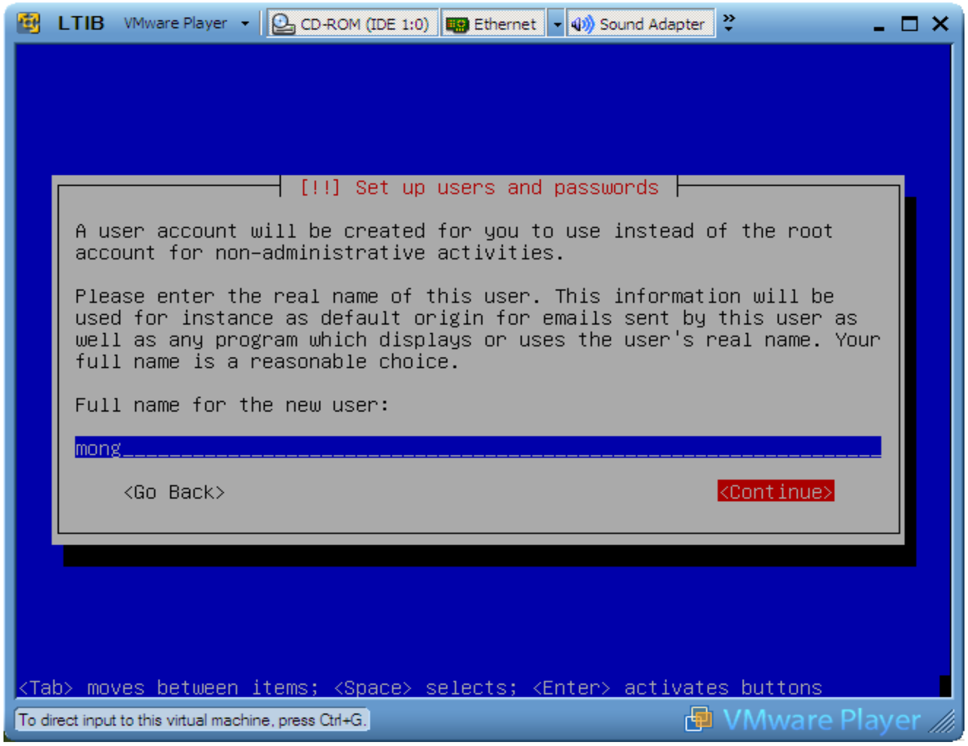


Figure 36. User Configuration

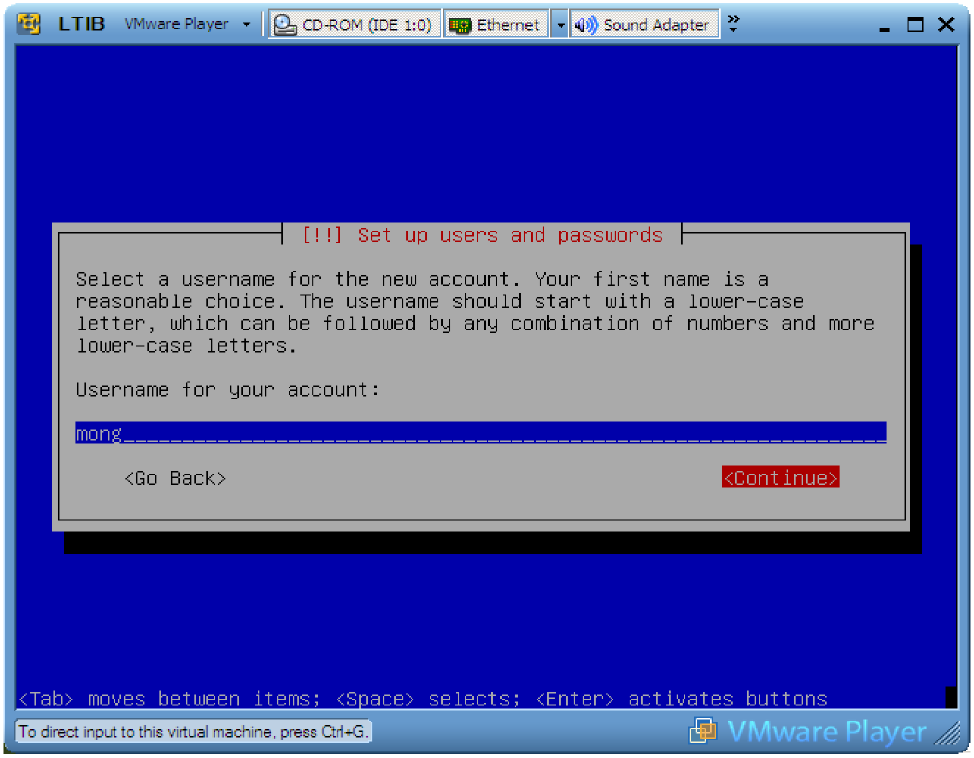


Figure 37. User Password Configuration

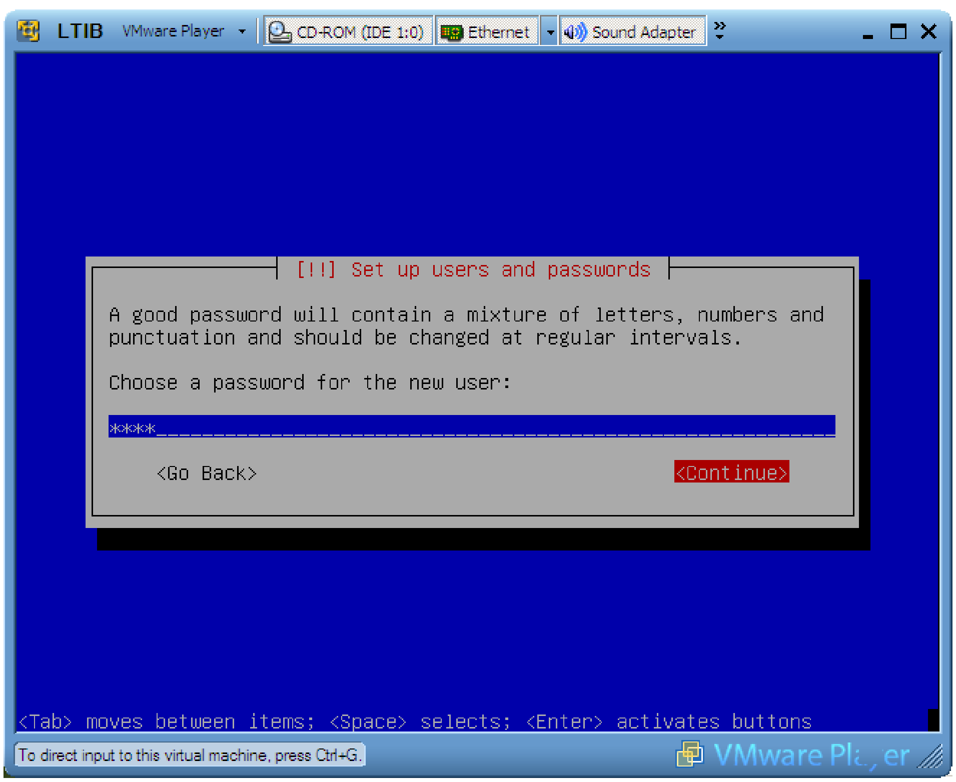


Figure 38. User Password Configuration

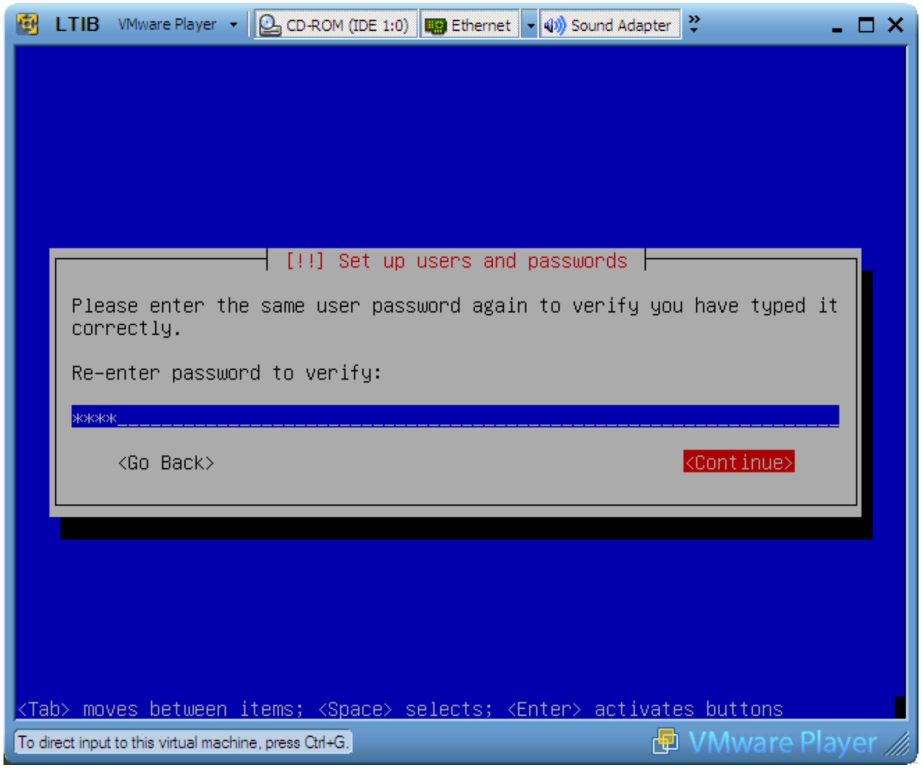


Figure 39. User Password Verification

The Debian installer installs the Linux base system on the VM as shown in Figure 40.

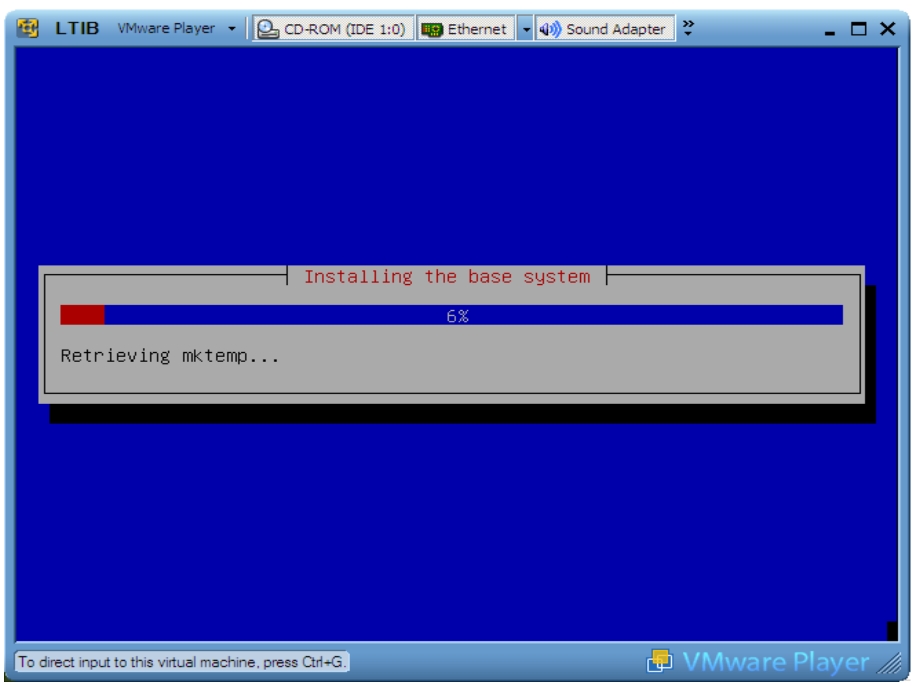


Figure 40. Installation of Debian Base Configuration

4.5 Configure the Package Manager

Debian installer prompts the user through a series of windows, shown in Figure 41 through Figure 44, to configure the package manager. The correct HTTP proxy information should be used in the text entry box provided in Figure 44.

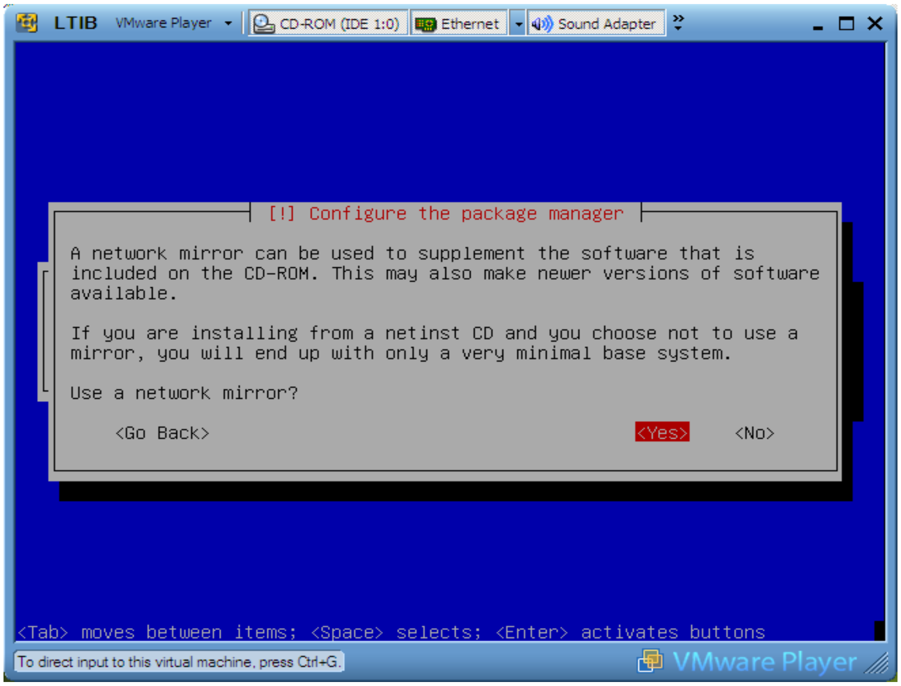


Figure 41. Package Manager Configuration

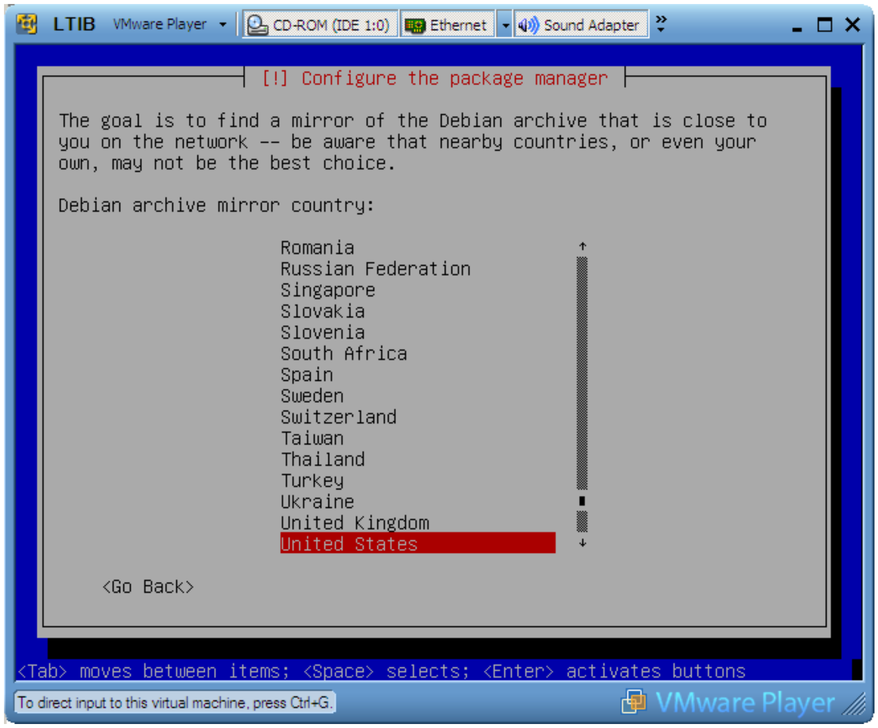


Figure 42. Package Manager Mirror Site Location

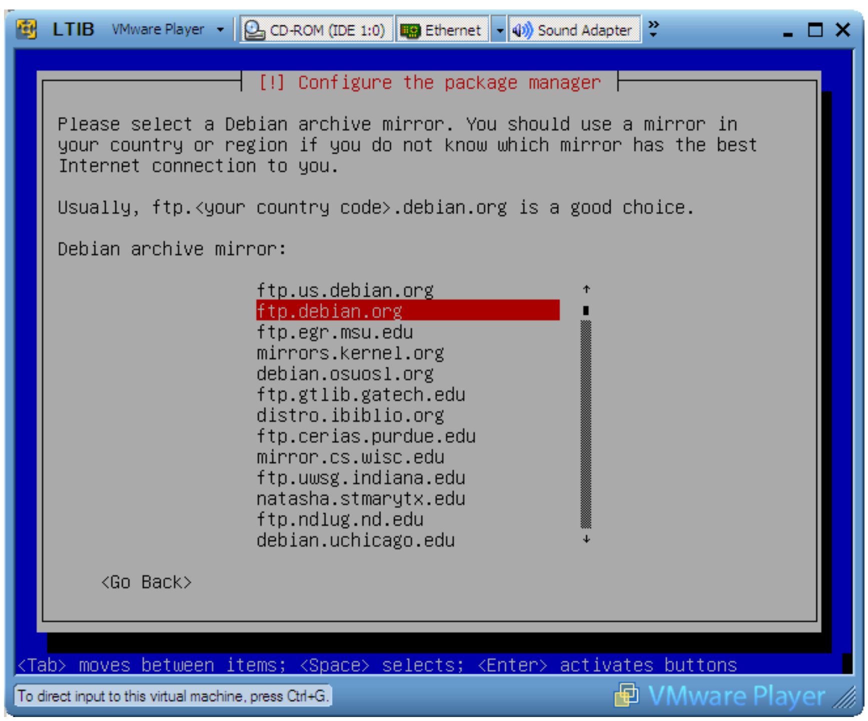


Figure 43. Package Manager Mirror Website Address

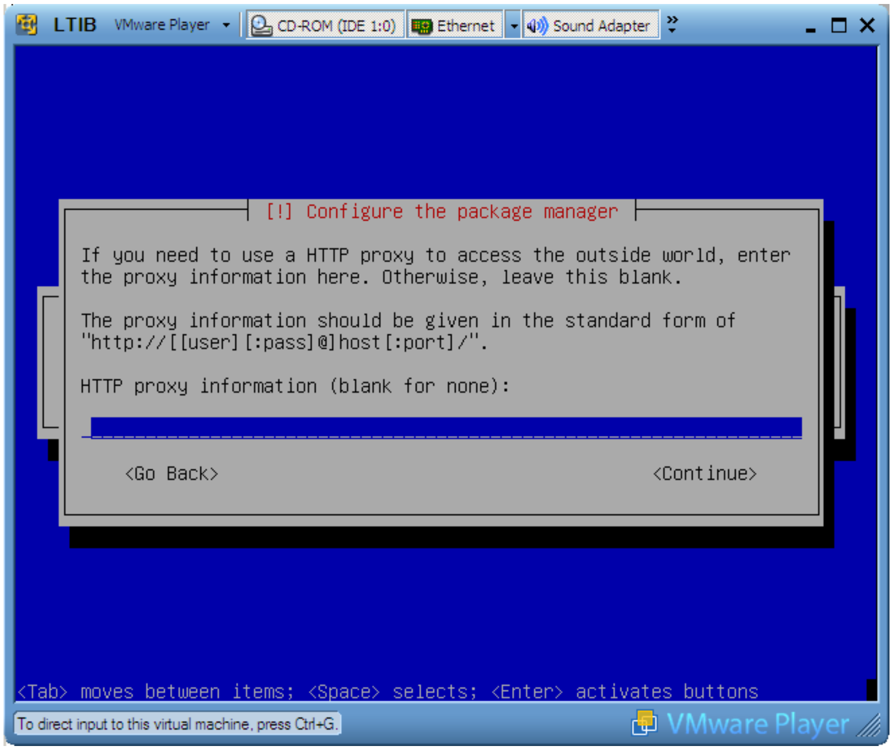


Figure 44. Debian Proxy Configuration

The Debian installer scans the mirror site and configures the applications.

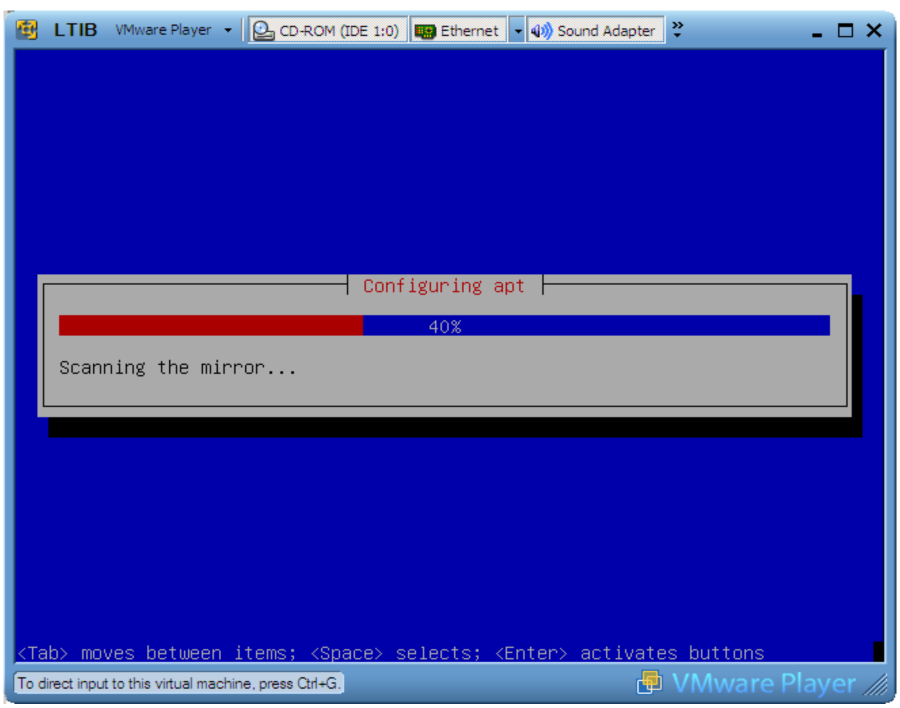


Figure 45. Debian apt Configuration

Select **“<YES>”** to participate in the package usage survey, or **“<NO>”** to decline.

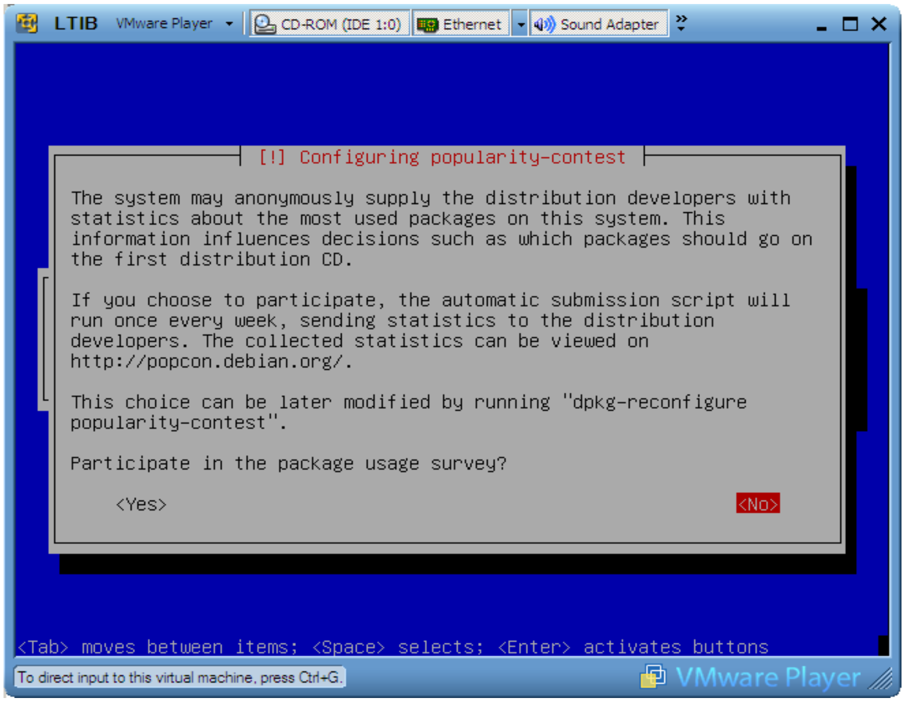


Figure 46. Package Usage Survey

4.6 Software Selection

Select “Desktop environment” and “Standard System,” as shown in Figure 47, then press “Enter."

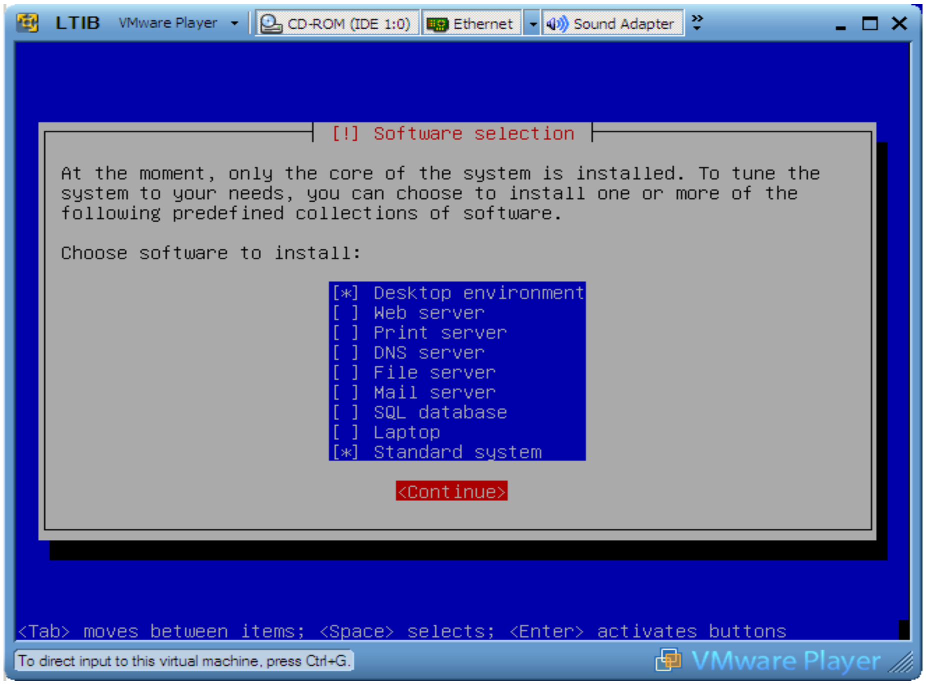


Figure 47. Debian Software Selection

Debian installer now installs the select software (see Figure 48). This process can take more than an hour depending on the mirror site and the network connection.

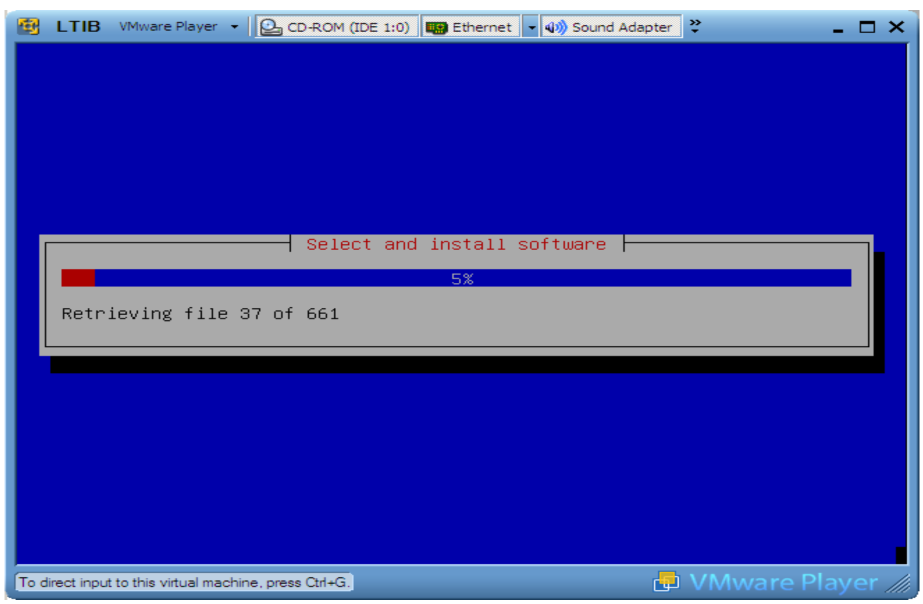


Figure 48. Debian Package Installation

Select **“<YES>”** for the option shown in Figure 49.

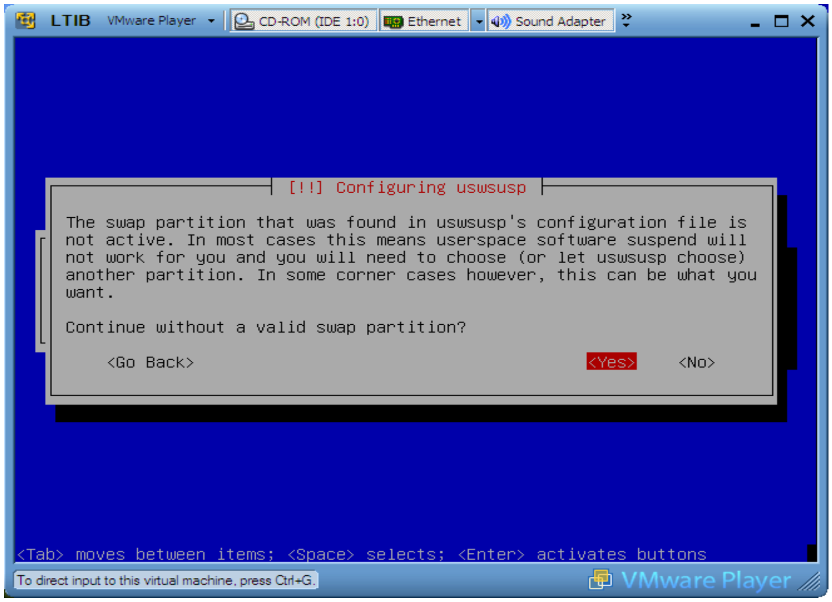


Figure 49. Debian Swap Partition Configuration

4.7 Configure XServer Xorg

Configure the video mode appropriately using the up and down arrow key and spacebar. Press “Enter” when certain that the display supports all the modes chosen. See Figure 50.

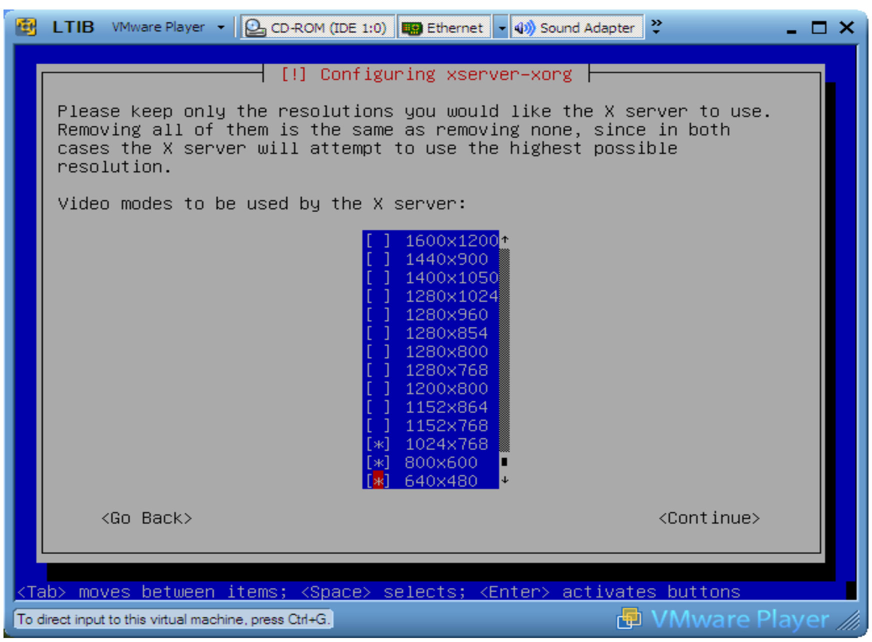


Figure 50. XServer Screen Resolution

As shown in Figure 51, Debian configures the system based on the information just provided.

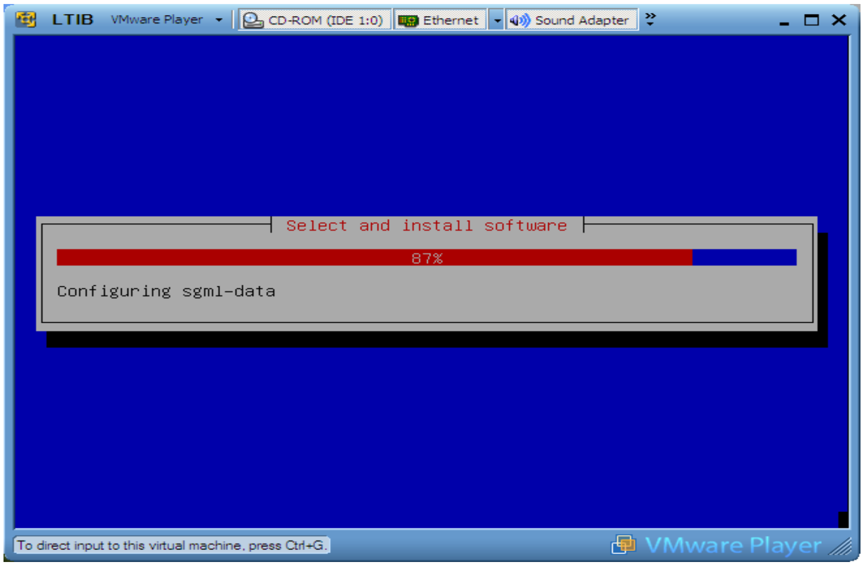


Figure 51. XServer Installation

4.8 Install GRUB Boot Loader

Debian now installs GRUB boot loader on the hard disk. Select “<YES>” and then press the “Enter” key. Select “<Continue>” as shown in Figure 52. Then press the “Enter” key.

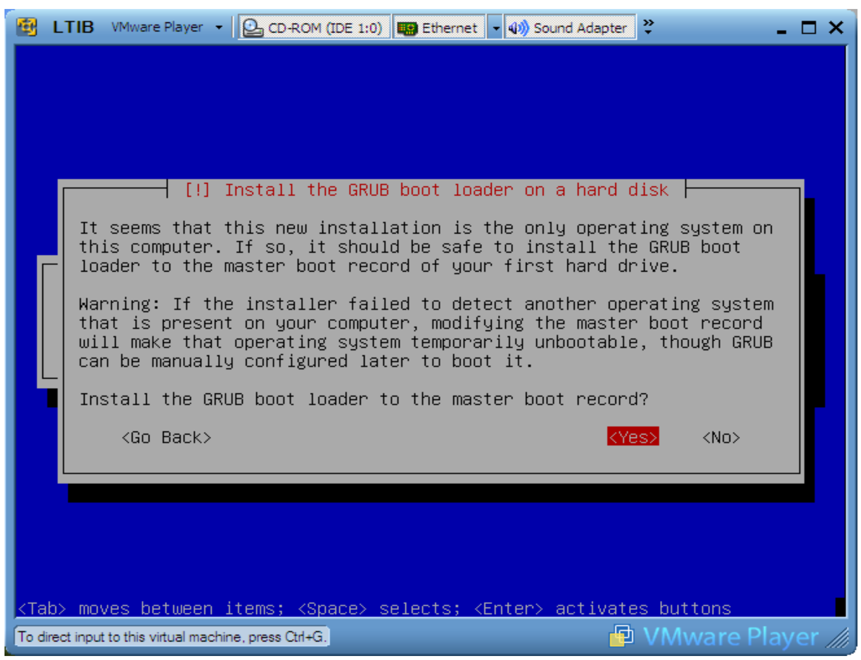


Figure 52. Grub Bootloader Installation

4.9 Finish the Installation

After completing the installation, the Debian installer ejects the installation. It reboots into the new system that has been successfully installed. See Figure 53 through Figure 55.

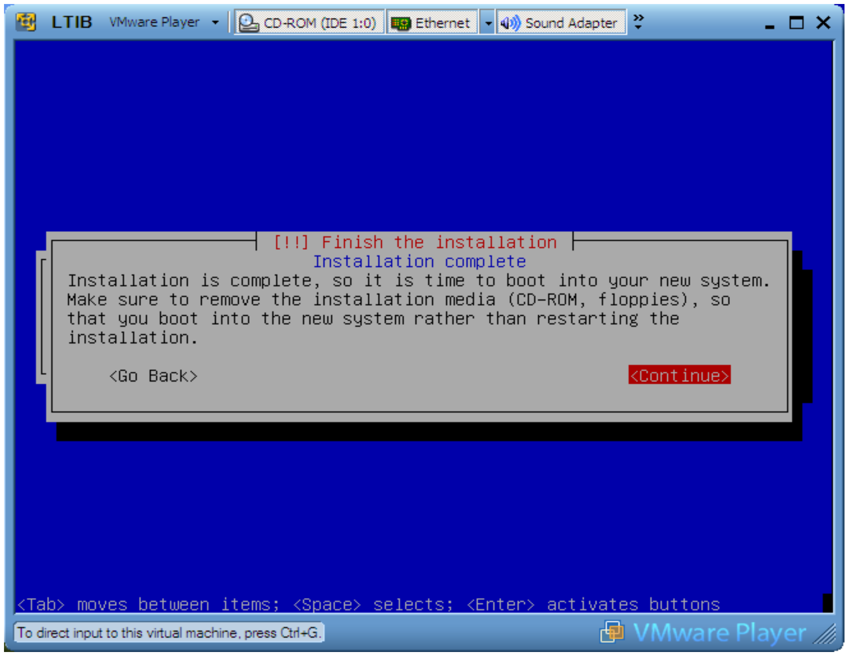


Figure 53. Launching Debian on Virtual Machine

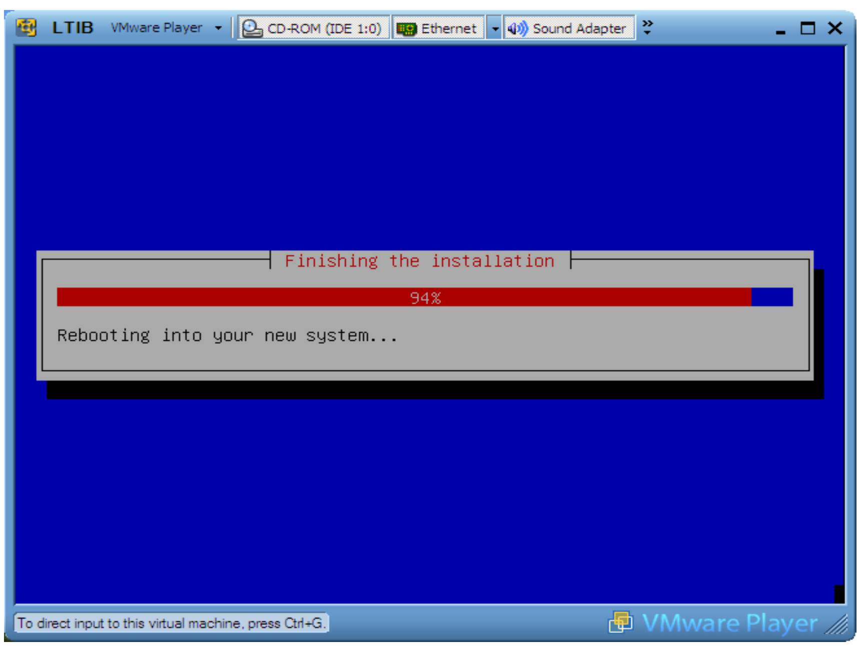


Figure 54. Reboot of Virtual Machine

![NXP logo]

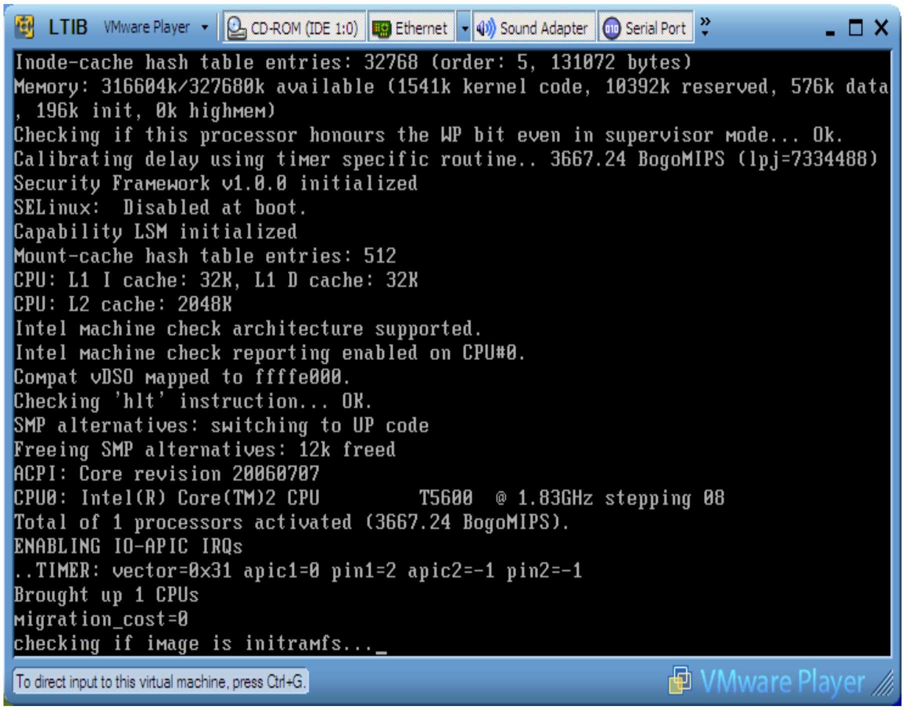


Figure 55. Debian Reboot on Virtual Machine

Enter the login name and user password as shown in Figure 56 and Figure 57, respectively. At this point, Debian Linux has been successfully installed in the virtual machine. Figure 58 shows the Debian desktop environment.

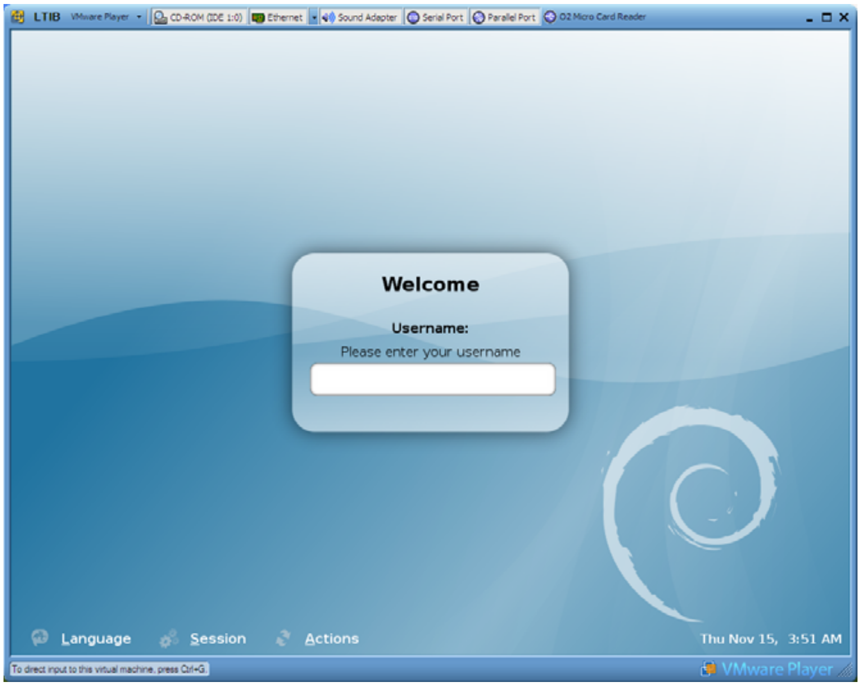


Figure 56. Debian User Logon

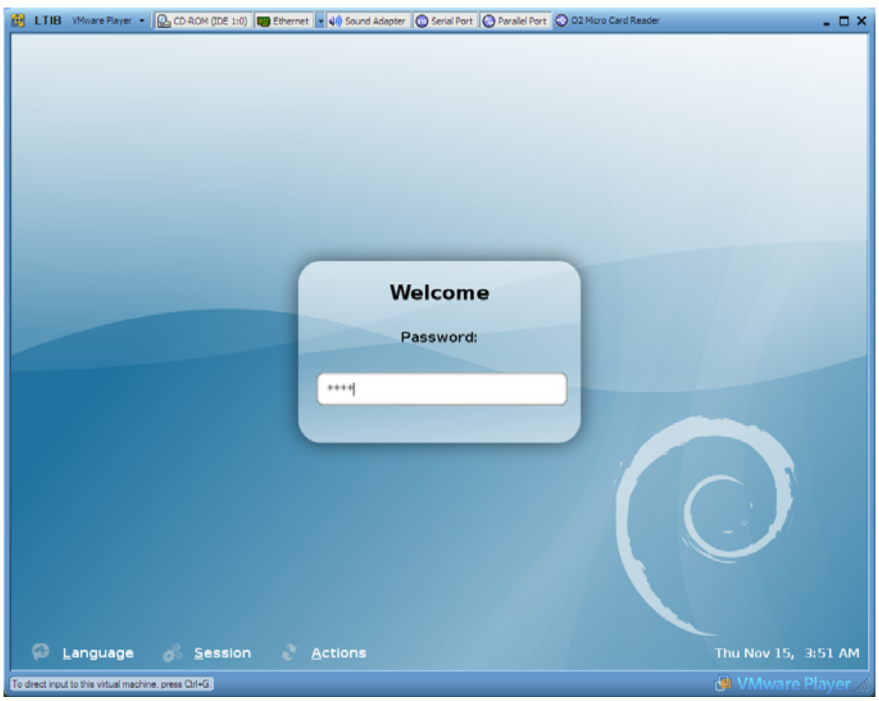


Figure 57. Debian User Password Screen

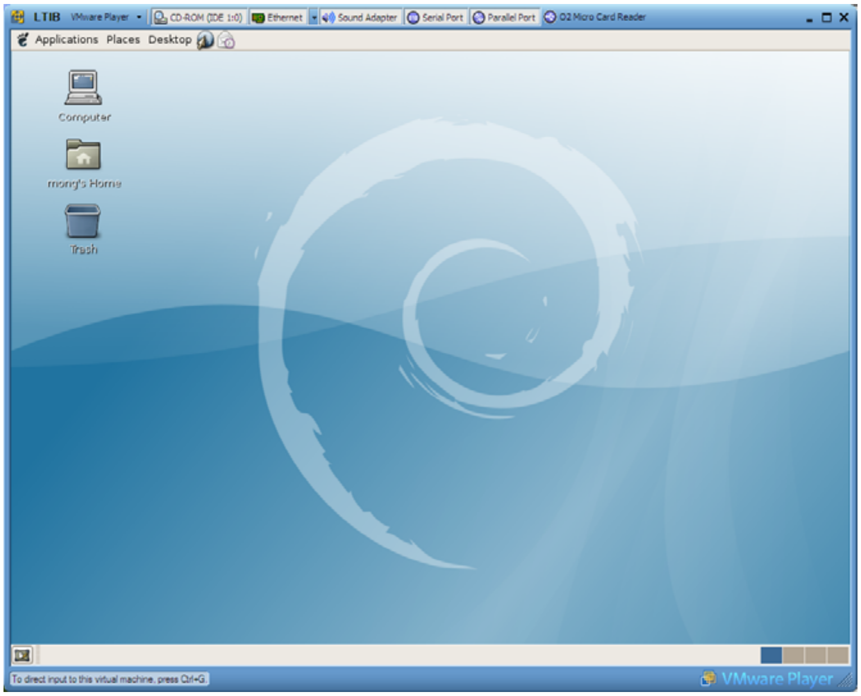


Figure 58. Debian Desktop Environment

5 Install Freescale LTIB for i.MX Family of Processors

After installing Linux on the virtual machine, LTIB has to be installed on this virtual machine. LTIB helps the user configure the Linux image to be downloaded on the target, that is, the i.MX 3-stack development platform.

The Freescale version of LTIB for the i.MX family of processors, can be downloaded from the website shown on the back cover of this document. Navigate to the desired i.MX processor, and under Board Support Packages, download “Linux BSPs for Freescale i.MX.” This file can be saved onto a USB drive or directly downloaded into the virtual machine. To be able to access the internet from the virtual machine, Network device setting should be chosen as bridged and proxy settings should be set appropriately as discussed in Section 5.4 .

From this point onwards, the “**$**” sign is the prompt of the Bash shell in the virtual machine and the “=>” sign prompt represents the Redboot prompt on the PDK.

This section obviously does not apply to WinCE users and they should follow the guidelines in the WinCE Software Development Kit.

The following subsections guide the user through installing LTIB, configuring the networking protocols to be able to connect to the target from the virtual machine, and downloading the images from the host, that is, the virtual machine, to the target.

5.1 Installing the Board Support Package

Follow the steps shown in Figure 59 and invoke the Bash shell or terminal window as shown in Figure 60. If the BSP was downloaded into the virtual machine, then it should be installed using the install command as shown below. If the BSP is in ISO format, it should be written to a CD and then the BSP can be installed using the install command as shown below. However, if the BSP was copied into a USB drive, insert the USB drive with the BSP software into the computer's USB type A connector.

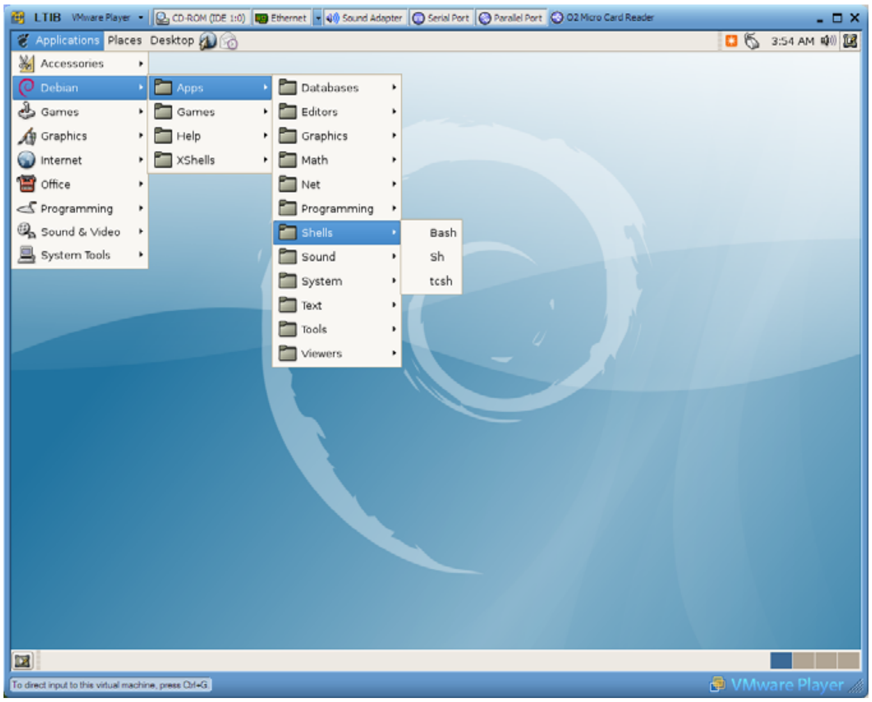


Figure 59. Launching Bash Shell in Debian

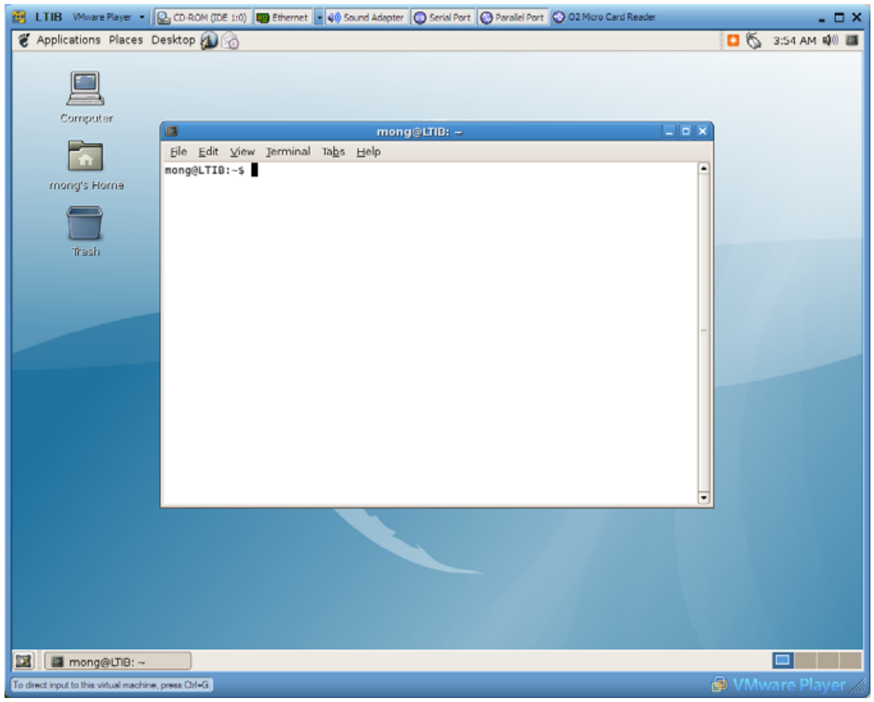


Figure 60. Bash Shell Prompt in Debian

The system detects the USB drive as shown in Figure 61.

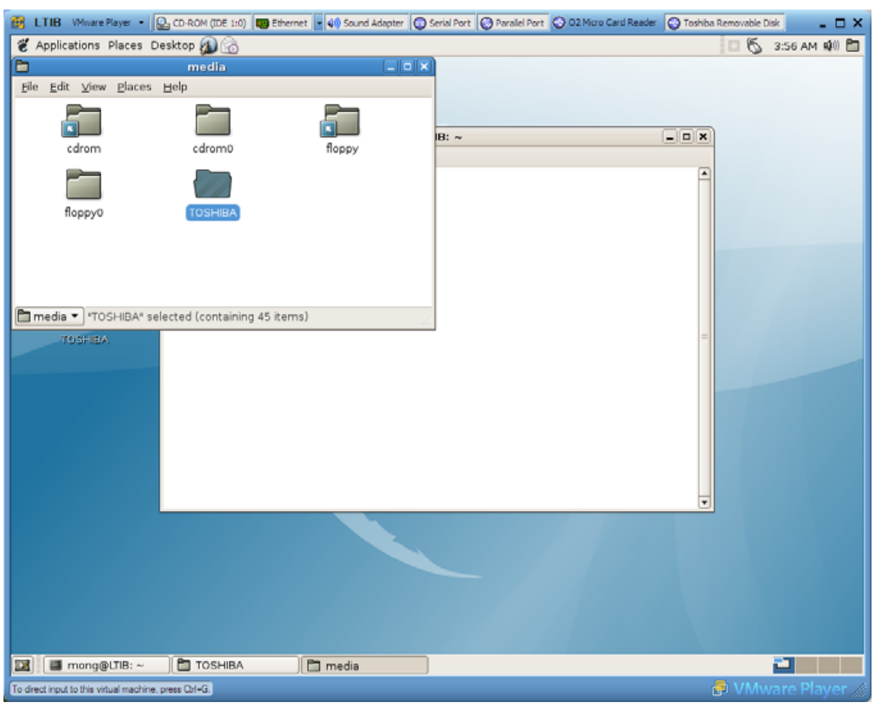


Figure 61. USB Folder Autodetection in Debian

The USB drive used may differ; therefore, the device name and the user name should be changed appropriately in the following sections.

Example 1. Mount LTIB in ISO format from USB if no tar.gz available

$ mkdir LTIBISO ← creates a directory

$ su ← upgrades to super user to use root commands

Password: ← enters the root password

$ mount –o loop /media/disk/imx_mn_ads_date /home/user/LTIBISO

The USB drive LED should flash.

$ exit ← returns to non-root status

$ LTIBISO/install ← installs the BSP software

The BSP EULA appears; to continue installing, accept the license.

The installation prompts, "Where do you want to install LTIB? (/home/user)." Press the "**Enter**" key to choose this and proceed.

It is important not to install LTIB as the super user to avoid issues with access permissions. In hosts with Ubuntu, the commands should be prepended with sudo to be the superuser.

5.2 Requirements for Running LTIB

When LTIB is running, it writes information into /opt/freescale/pkgs and /opt/freescale/ltib. LTIB must have permission to write into this directory. This can be ensured by appending the following command (in blue) is in the file /etc/sudoers.

Example 2. Modify /etc/sudoers File

```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

user ALL = NOPASSWD: /usr/bin/rpm, /opt/freescale/ltib/usr/bin/rpm
```

If the command is not in the sudoers file, the following instructions can be used to add this command to the file **/etc/sudoers**.

Example 3. Configure LTIB

$ su ← upgrades to super user to use root commands

Password: ← enters the root password

$ cd /etc ← change to /etc directory

$ nano sudoers ← uses nano text editor to add the command line

$ exit ← returns to non-root status

$ cd ~ ← returns to user home directory

$ cd ltib

$ ./ltib --configure ← run LTIB

When running LTIB, LTIB detects if all the required software packages are present in the system. If any software component is missing, LTIB generates a list of the missing software components needed for this process as shown in Figure 62.

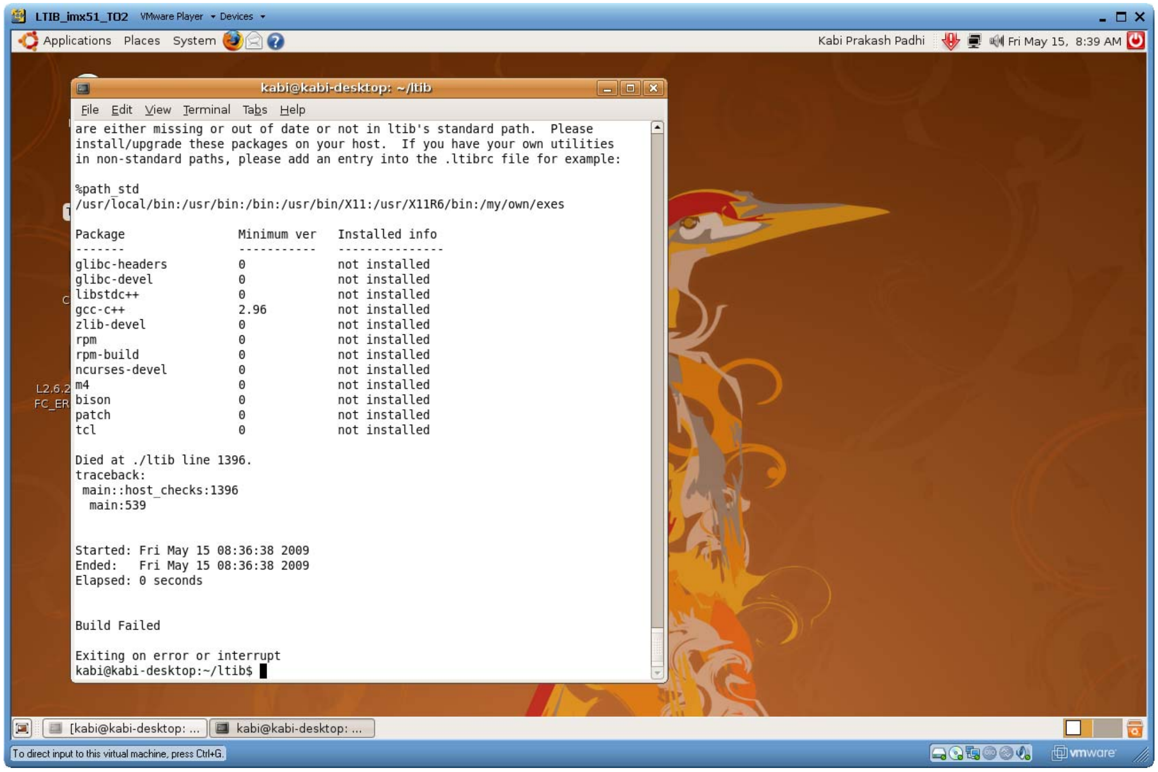


Figure 62. LTIB Missing Packages

The missing software components are as follows:

1) glibc-headers ← for 1 to 5 search for gcc and g++
2) glibc-devel selects gcc, gcc 3.4 and 4.1 , g++ and g++ 4.1
3) libstd++.devel
4) gcc-c++
5) zlib-devel ← search zlib and select zliblg-dev
6) rpm
7) rpm-build
8) ncurses-devel ← search ncurses and select libncurses-dev
9) bison
10) patch

11) tcl

12) m4

The Synaptic Package Manager can be used to install the missing software components. See Figure 63.

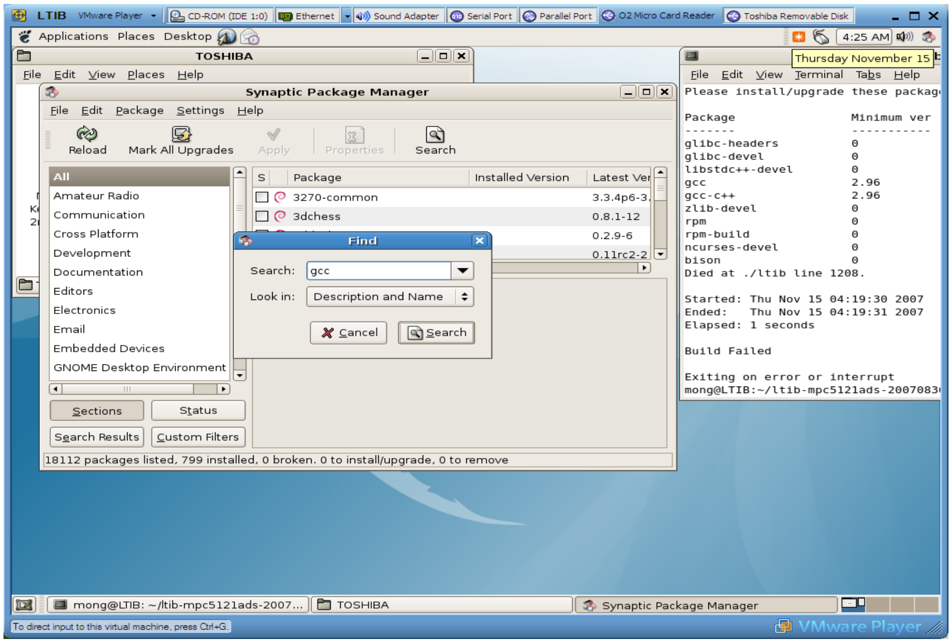


Figure 63. Synaptic Package Manager

After all the missing software components are found and installed, LTIB should be run again.

Example 4. Configure LTIB after installation of missing packages

$ ./ltib --configure ← run LTIB

When LTIB has successfully completed installation, the message in Figure 64 is shown.

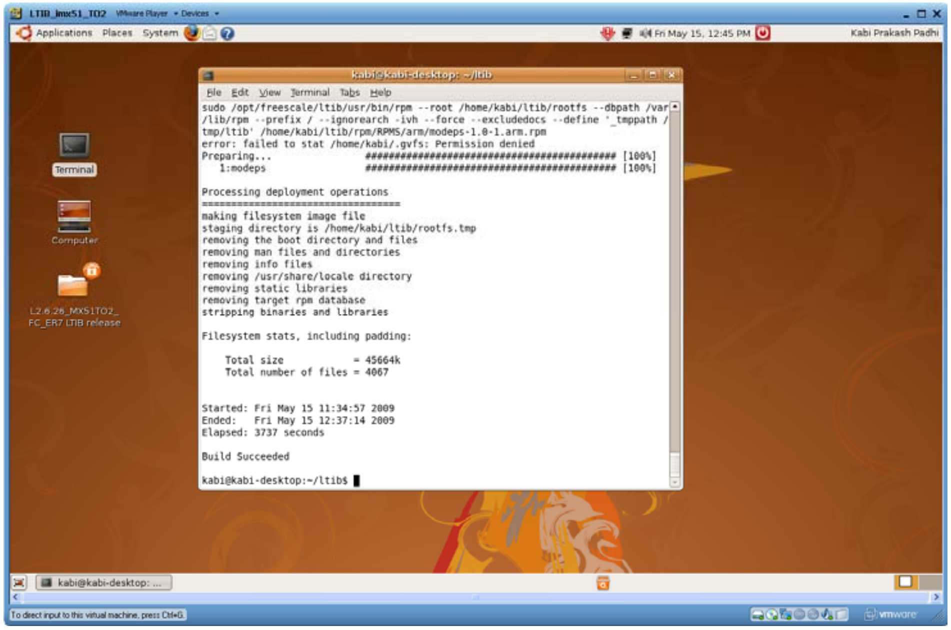


Figure 64. LTIB Build Successful

5.3 TFTP Host Setup

The Synaptic Package Manager should be used to install the TFTP and the NFS servers. The NFS server can be installed by using the NFS kernel server package.

The tftpboot directory should be created for tftp transfers between the host and the target.

Example 5. Create tftpboot folder

$ su ← upgrades to super user to use root commands

Password: ← enters the root password

$ cd /

$ mkdir –p /tftpboot

For convenience, the tftpboot/rootfs directory can be symbolically linked to an exportable directory.

Example 6. Link LTIB rootfs folder to tftpboot rootfs folder

$ ln –s /home/user/ltib/rootfs /tftpboot/rootfs

Copy over the kernel, boot loader, devices tree and root file system image to the /tftpboot directory

Example 7. Copy OS image and root file system to tftpboot folder

$ su ← upgrades to super user to use root commands

Password: ← enters the root password

$ cp /home/user/ltib/rootfs/boot/zImage /tftpboot

$ cp /home/user/ltib/rootfs/boot/rootfs.jffs2 /tftpboot

5.4 Setup Host IP Address

Under the Ethernet option of the VMware Player, the Ethernet should be connected and set to "Bridged" mode as shown in Figure 65. Next the following steps should be followed.

Example 8. Configure Ethernet

$ su ← upgrades to super user to use root commands

Password: ← enters the root password

$ ifconfig eth0:0 192.168.1.xx netmask 255.255.0.0 up

The 192.168.1.xx address can be used when both the target and the host are connected to an Ethernet hub or router. If not connected to a hub, the router can assign an IP address. See Figure 65.

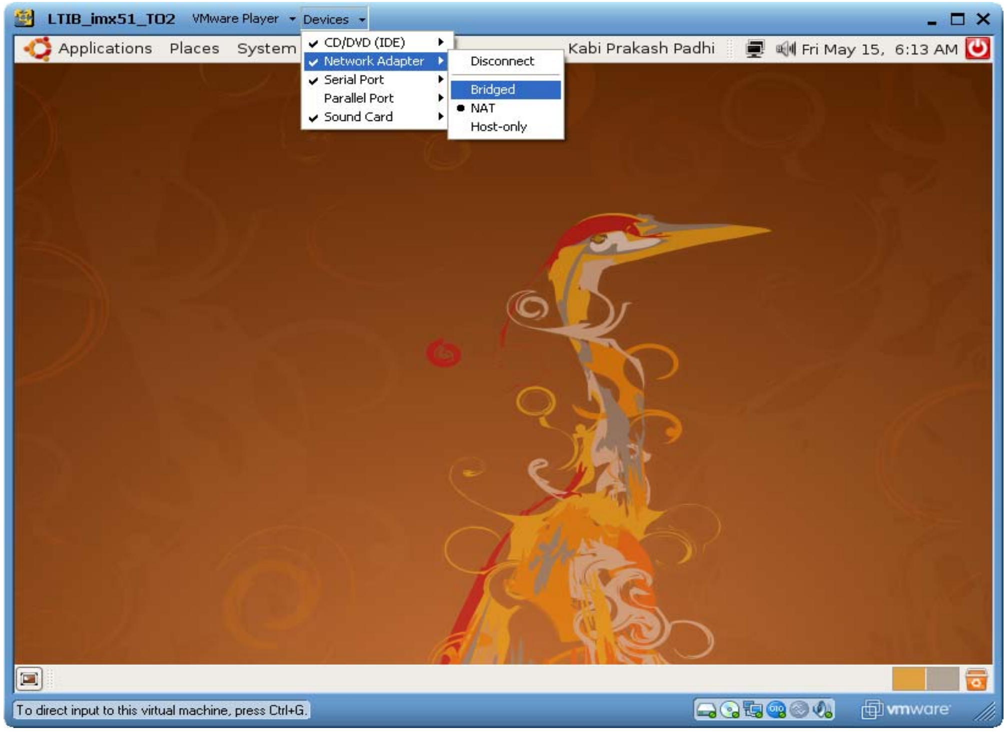


Figure 65. Bridging Network connections of Virtual Machine with Windows

5.5 Setup TFTP and NFS Server

The following line should be added to the file /etc/exports. If the exports file does not exist, it should be created.

Example 9. Modify /etc/exports File

```
/tftpboot/rootfs/ *(rw,no_root_squash,async)
```

NOTE

Make sure there is not space in between from the “*” to “)”

The file **/etc/xinetd.d/tftp** should be edited to enable tftp as appended. If the tftp file does not exist, it should be created.

If the directory xinetd.d does not exist, xinetd is probably not installed. The Synaptic Package Manager should be used to install xinetd if required.

Example 10. Modify /etc/xinetd.d/tftp File

```
service tftp
{
        disable         = no
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        server          = /usr/sbin/in.tftpd
        server_args     = /tftpboot
}
```

The server args parameter specifies the path for files to be tftped from host to target.

5.6 Restart TFTP and NFS Server

Restart the TFTP and NFS server on the host virtual machine as follows:

Example 11. Restart nfs and xinetd servers

```
$ su              ← upgrades to super user to use root commands
Password:         ← enters the root password

$ /etc/init.d/xinetd restart
$ /etc/init.d/nfs-kernel-server restart
```

5.6.1 Connect to TFTP Server

Sometimes the tftp server might not be connected to the host ipaddress. This can be verified by the following set of commands.

Example 12. Tftp status check

$ tftp

> status

Not connected

If the tftp server is not connected, the following commands should be used to connect to the tftp server and verify the connection status.

Example 13. Connect to tftp server on virtual machine

> connect 192.168.1.xx

> status

Connected to 192.168.1.xx

> get zImage

File transferred in 12 seconds.

In case tftp does not work, ensure file and folder permissions in the tftpboot folder are set to 777 using chmod. Also, the file sizes should be greater than 0.

5.7 Target Setup

The system should now be setup as shown in Figure 1 earlier. Power supplies are not shown in the diagram and should be connected as well.

a. Connect i.MX PDK 3-stack board as in the Quick Start guide.

b. Connect debug board to the network via the Ethernet port.

c. Connect the debug board power supply.

d. Connect the debug board to the host machine via serial port..

e. Start a serial communication program. Serial settings are 115200 baud, 8 data bits, none parity, one stop bit and none flow control.

f. Flash Redboot onto NAND / NOR using the ATK User Guide [4].

g. Power up the PDK board and see the redboot messages on the hyperterminal as follows:

Example 14. Redboot Log on Hyperterminal

```
++Searching for BBT table in the flash ...

Found version 1 Bbt0 at block 8191 (0x7ffc0000)
Total bad blocks: 0
... Read from 0x07ec0000-0x07f00000 at 0x00080000: ..
... Read from 0x07eb3000-0x07eb4000 at 0x000bf000: .

LAN92xx Driver version 1.1
SMSC LAN9217: ID = 0x117a REV = 0x0
[Warning] FEC not connect right PHY: ID=c0001c
FEC: [ HALF_DUPLEX ] [ disconnected ] [ 10M bps ]:
... waiting for BOOTP information
DHCP reply: 2, not 5
Ethernet eth0: MAC address 00:04:9f:00:8d:a0
IP: 192.168.1.xy/255.255.255.0, Gateway: 10.193.102.254
```

```
Default server: 0.0.0.0

Reset reason: Power-on reset
fis/fconfig from NAND
Boot switch: UNKNOWN
        NAND: ADDR CYCLES:4: MLC: 128B spare (4-bit ECC):
PAGE SIZE: 2K: BUS WIDTH: 8

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version FSL 200845 - built 07:05:22, Nov 10 2008

Platform: MX51 3-Stack (Freescale i.MX51 based) PASS 1.1 [x32 DDR]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited

RAM: 0x00000000-0x07f00000, [0x00095368-0x07eb1000] available
FLASH: 0x00000000 - 0x80000000, 8192 blocks of 0x00040000 bytes each.
== Executing boot script in 1.000 seconds - enter ^C to abort
^C
RedBoot>
```

It is essential for the target to get an IP address during redboot to be able to TFTP files onto the target. Follow steps listed in Sections 5.3 – 5.7 to ensure the target gets an IP address. Sometimes it might be necessary to reset the Ethernet hub for this.

5.8 TFTP files onto Target and Store in Flash

The kernel image and the root file system can be tftped to the target from the redboot prompt on the hypterminal. The hyperterminal can be either on the Windows machine or on the virtual machine. This section walks through the steps for this. First, the flash should be erased.

Example 15. Initialize Flash and tftp OS image onto target RAM

```
RedBoot> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0x00080000-0x000c0000: .
... Program from 0x07ec0000-0x07f00000 at 0x00080000: ..

RedBoot> load -r -b 0x100000 zImage.mx51 -h 192.168.1.xx (host IP address)
Using default protocol (TFTP)
        Raw file loaded 0x00100000-0x002b11d7, assumed entry at 0x00100000
```

The tftped file is then copied from RAM into flash.

Example 16. Copy OS image from RAM onto Flash

```
RedBoot> fis create kernel
... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..
... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..
... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..
... Erase from 0x000c0000-0x00280000: .......
... Program from 0x00100000-0x002b11d8 at 0x000c0000: ........
... Erase from 0x00080000-0x000c0000: .
... Program from 0x07ec0000-0x07f00000 at 0x00080000: ..
```

The same process is repeated for the Root file system.

Example 17. TFTP root file system onto RAM and copy into Flash

```
RedBoot> load -r -b 0x100000 rootfs_mx51_dual_display.jffs2 -h 192.168.1.xx (host IP address)
Using default protocol (TFTP)
```

```
Raw file loaded 0x00100000-0x0598f95f, assumed entry at 0x00100000


RedBoot> fis create -f 0x600000 root

... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..

... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..

... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..

... Erase from 0x00600000-0x05ec0000:
..........................................................................................................................................
..........................................................................................................................................
..................

... Program from 0x00100000-0x0598f960 at 0x00600000:
..........................................................................................................................................
..........................................................................

.......................................................................

... Erase from 0x00080000-0x000c0000: .

... Program from 0x07ec0000-0x07f00000 at 0x00080000: ..
```

Now the kernel image and the root file system are resident on the target Flash.

5.9 Redboot Configuration

Before running Linux, the redboot environment variables should be set correctly using the fconfig command. A detailed list of Redboot commands is listed in [3].

Setting the correct boot script allows redboot to invoke Linux after redboot has finished execution. The timeout should always be > 0 seconds.

Example 18. Redboot configuration

```
RedBoot> fconfig

Run script at boot: true

Boot script:

fis load kernel
```

exec -c "noinitrd console=ttymxc0,115200 root=/dev/mtdblock2 rw rootfstype=jffs2 ip=none"

Enter script, terminate with empty line

>> fis load kernel

>> exec -c "noinitrd console=ttymxc0,115200 root=/dev/mtdblock2 rw rootfstype=jffs2 ip=dhcp" (If no IP address is required in Linux, ip=none)

>>

Boot script timeout (1000ms resolution): 1

Use BOOTP for network configuration: true

Default server IP address:

Board specifics: 0

Console baud rate: 115200

Set eth0 network hardware address [MAC]: false

Set FEC network hardware address [MAC]: false

GDB connection port: 9000

Force console for special debug messages: false

Network debug at boot time: false

Default network device: lan92xx_eth0

Update RedBoot non-volatile configuration - continue (y/n)? y

... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..

... Erase from 0x00080000-0x000c0000: .

... Program from 0x07ec0000-0x07f00000 at 0x00080000: ..

Redboot>

5.10 Running Linux on Target

Now that Linux has been loaded onto flash and redboot has been configured to load Linux, Linux can be invoked from either the Redboot prompt or by doing a reset. The commands to invoke Linux from the Redboot prompt are the same as in the Redboot boot script. The log has been included in Appendix A.

5.11 References

[1] Mong Sim, “Development Environment for 5121e LTIB”, Application Note

[2] http://en.wikipedia.org/wiki/Virtual_machine

[3] http://ecos.sourceware.org/docs-latest/redboot/redboot-guide.html

[4] Advanced Took Kit (ATK) User guide

[5] http://www.imxdev.org

[6] http://en.wikipedia.org/wiki/Comparison_of_Linux_distributions

6 Revision History

Table 1 provides a revision history for this application note.

Table 1. Revision History

Revision	Date	Substantive Changes
1	8/12/2009	Added missing slash to path shown in Example 9, "Modify /etc/exports File." Added clarification after Example 13, "Connect to tftp server on virtual machine."
0	7/2/2009	Initial release

Appendix A i.MX51 Bootlog

RedBoot> reset

... Resetting.

++Searching for BBT table in the flash ...

.

Found version 1 Bbt0 at block 8191 (0x7ffc0000)

```
Total bad blocks: 0
... Read from 0x07ec0000-0x07f00000 at 0x00080000: ..
... Read from 0x07eb3000-0x07eb4000 at 0x000bf000: .

LAN92xx Driver version 1.1
SMSC LAN9217: ID = 0x117a REV = 0x0
[Warning] FEC not connect right PHY: ID=c0001c
FEC: [ HALF_DUPLEX ] [ disconnected ] [ 10M bps ]:
... waiting for BOOTP information
Ethernet eth0: MAC address 00:04:9f:00:8d:a0
IP: 192.168.1.xx/255.255.255.0, Gateway: 192.168.1.1
Default server: 0.0.0.0

Reset reason: WDOG reset
fis/fconfig from NAND
Boot switch: UNKNOWN
        NAND: ADDR CYCLES:4: MLC: 128B spare (4-bit ECC):
PAGE SIZE: 2K: BUS WIDTH: 8

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version FSL 200845 - built 07:05:22, Nov 10 2008

Platform: MX51 3-Stack (Freescale i.MX51 based) PASS 1.1 [x32 DDR]
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
Copyright (C) 2003, 2004, 2005, 2006 eCosCentric Limited
```

```
RAM: 0x00000000-0x07f00000, [0x00095368-0x07eb1000] available

FLASH: 0x00000000 - 0x80000000, 8192 blocks of 0x00040000 bytes each.

== Executing boot script in 1.000 seconds - enter ^C to abort

RedBoot> fis load kernel

... Read from 0x07ec0000-0x07eff000 at 0x00080000: ..

... Read from 0x00100000-0x002b11d8 at 0x000c0000: ........

RedBoot> exec -c "noinitrd console=ttymxc0,115200 root=/dev/mtdblock2 rw rootfstype=jffs2
ip=dhcp"

entry=0x90008000, target=0x90008000

Using base address 0x00100000 and length 0x001b11d8

Uncompressing Linux.. ..........................................................
    ...........................................................................

...........................................................

........ done, booting the kernel.


Linux version 2.6.26-00238-gb424051-dirty (r62914@sdclx4) (gcc version 4.1.2) #46 PREEMPT Tue
Dec 2 12:20:10 CST 2008

CPU: ARMv7 Processor [412fc081] revision 1 (ARMv7), cr=00c5387f

Machine: Freescale MX51 3-Stack Board

Memory policy: ECC disabled, Data cache writeback

CPU0: D VIPT write-back cache

CPU0: cache: 768 bytes, associativity 1, 8 byte lines, 64 sets

Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 32512

Kernel command line: noinitrd console=ttymxc0,115200 root=/dev/mtdblock2 rw rootfstype=jffs2
ip=dhcp

MXC IRQ initialized
```

PID hash table entries: 512 (order: 9, 2048 bytes)

MXC GPT timer initialized, rate = 8000000

Console: colour dummy device 80x30

Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)

Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)

Memory: 128MB = 128MB total

Memory: 126128KB available (3360K code, 298K data, 108K init)

SLUB: Genslabs=12, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1

Mount-cache hash table entries: 512

CPU: Testing write buffer coherency: ok

net_namespace: 192 bytes

NET: Registered protocol family 16

CPU is i.MX51 Revision 1.1

MXC GPIO hardware

3-Stack Debug board detected, rev = 0x0200

IRAM READY

Using SDMA I.API

MXC DMA API initialized

SCSI subsystem initialized

CSPI: mxc_spi-0 probed

usbcore: registered new interface driver usbfs

usbcore: registered new interface driver hub

usbcore: registered new device driver usb

MXC I2C driver

MXC HS I2C driver

ASoC version 0.20

MC13892 regulator successfully probed

mc13892 1-0008: Loaded

NET: Registered protocol family 2

IP route cache hash table entries: 1024 (order: 0, 4096 bytes)

TCP established hash table entries: 4096 (order: 3, 32768 bytes)

TCP bind hash table entries: 4096 (order: 2, 16384 bytes)

TCP: Hash tables configured (established 4096 bind 4096)

TCP reno registered

NET: Registered protocol family 1

LPMode driver module loaded

Static Power Management for Freescale i.MX51

PM driver module loaded

usb: Host 1 host (isp1504) registered

usb: DR host (utmi) registered

usb: DR gadget (utmi) registered

JFFS2 version 2.2. (NAND) Â© 2001-2006 Red Hat, Inc.

msgmni has been set to 246

io scheduler noop registered

io scheduler anticipatory registered

io scheduler deadline registered

io scheduler cfq registered (default)

mxc_ipu mxc_ipu: Channel already uninitialized 9

mxc_sdc_fb mxc_sdc_fb.0: mxcfb: Error initializing panel.

Console: switching to colour frame buffer device 30x20

```
mxc_sdc_fb mxc_sdc_fb.0: fb registered, using mode <NULL>
mxc_sdc_fb mxc_sdc_fb.1: fb registered, using mode <NULL>
mxc_sdc_fb mxc_sdc_fb.2: fb registered, using mode <NULL>
Serial: MXC Internal UART driver
mxcintuart.0: ttymxc0 at MMIO 0x73fbc000 (irq = 31) is a Freescale MXC
console [ttymxc0] enabled
mxcintuart.1: ttymxc1 at MMIO 0x73fc0000 (irq = 32) is a Freescale MXC
mxcintuart.2: ttymxc2 at MMIO 0x7000c000 (irq = 33) is a Freescale MXC
loop: module loaded
smsc911x: Driver version 2007-07-13.
eth%d: SMSC911x MAC Address: 00:04:9f:00:8d:a0
Linux video capture interface: v2.00
MXC Video Output MXC Video Output.0: Registered device video16
Driver 'sd' needs updating - please use bus_type methods
MXC MTD nand Driver 2.5
NAND device: Manufacturer ID: 0xec, Chip ID: 0xd3 (Samsung NAND 1GiB 3,3V 8-bit)
RedBoot partition parsing not available
Creating 5 MTD partitions on "NAND 1GiB 3,3V 8-bit":
0x00000000-0x00100000 : "bootloader"
0x00100000-0x00600000 : "nand.kernel"
0x00600000-0x08600000 : "nand.rootfs"
0x08600000-0x18600000 : "nand.userfs1"
0x18600000-0x40000000 : "nand.userfs2"
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
```

```
USB Mass Storage support registered.
MXC keypad loaded
clk: Unable to get requested clock: kpp_clk
input: mxckpd as /class/input/input0
mxc_rtc mxc_rtc.0: rtc core: registered mxc_rtc as rtc0
i2c /dev entries driver
MXC WatchDog Driver 2.0
clk: Unable to get requested clock: wdog_clk
MXC Watchdog # 0 Timer: initial timeout 60 sec
clk: Unable to get requested clock: scc_clk
SCC2: Driver Status is OK
Sahara base address: 0xfc2f8000
Sahara HW Version is 0x00000000
Sahara HW Version was not expected value.
Sahara going into cleanup
VPU initialized
Registered led device: pmic_ledsr
Registered led device: pmic_ledsg
Registered led device: pmic_ledsb
Advanced Linux Sound Architecture Driver Version 1.0.16.
DMA Sound Buffers Allocated:UseIram=1 buf->addr=1ffe9000 buf->area=c88b8000 size=24576
DMA Sound Buffers Allocated:UseIram=1 buf->addr=97e58000 buf->area=fdf2d000 size=24576
asoc: wm8903-hifi-dai <-> imx-ssi-3 mapping ok
wm8903-i2c 1-001a: WM8903 revision 0
ALSA device list:
```

```
        #0: i.MX_3STACK (WM8903)
TCP cubic registered
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
ieee80211: 802.11 data/management/control stack, git-1.1.13
ieee80211: Copyright (C) 2004-2005 Intel Corporation <jketreno@linux.intel.com>
VFP support v0.3: implementor 41 architecture 3 part 30 variant c rev 2
input: mxc_ts as /class/input/input1
mxc input touchscreen loaded
mxc_rtc mxc_rtc.0: setting system clock to 1970-01-01 00:00:00 UTC (0)
eth0: SMSC911x/921x identified at 0xc881e000, IRQ: 256
eth0: link down
Sending DHCP requests .<6>eth0: link up, 100Mbps, full-duplex, lpa 0x45E1
., OK
IP-Config: Got DHCP answer from 0.0.0.0, my address is 192.168.1.xy
IP-Config: Complete:
        device=eth0, addr=192.168.1.xy, mask=255.255.255.0, gw=192.168.1.254,
        host=192.168.1.xy, domain=ap.freescale.net, nis-domain=(none),
        bootserver=0.0.0.0, rootserver=0.0.0.0, rootpath=
JFFS2 doesn't use OOB.
VFS: Mounted root (jffs2 filesystem).
Freeing init memory: 108K
init started: BusyBox v1.6.1 () multi-call binary
starting pid 939, tty '': '/etc/rc.d/rcS'
```

```
Setting the hostname to freescale
Mounting /proc and /sys
Starting the hotplug events dispatcher udevd
Synthesizing initial hotplug events
Mounting filesystems
Rootfs is read-only, mounting rw filesystem
mount: mounting usbfs on /proc/bus/usb failed
Starting syslogd and klogd
Setting up networking on loopback device:

Warning: no IPADDR is set, please set this from the ltib
config screen, or directly in /etc/rc.d/rc.conf.
IP address setup bypassed

Starting inetd:
Starting the dropbear ssh server:
modprobe: cannot parse modules.dep
starting pid 1897, tty '': '/sbin/getty'
JFFS2 doesn't use OOB.

arm-none-linux-gnueabi-gcc (GCC) 4.1.2
root filesystem built on Wed, 19 Nov 2008 11:17:02 +0800
Freescale Semiconductor, Inc.

freescale login: bt_fd is 4
```

```
bind: No such file or directory
DeviceManager: no battery device
CHIP_TYPE=IMX 51
modprobe: cannot parse modules.dep

arm-none-linux-gnueabi-gcc (GCC) 4.1.2
root filesystem built on Wed, 19 Nov 2008 11:17:02 +0800
Freescale Semiconductor, Inc.

freescale login: root

BusyBox v1.6.1 () Built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@freescale ~$
```

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064

Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: AN3875
Rev. 1
08/2009