EXAM 101

LPIC1 Junior Level Administrator Part 1

Mega Guide

Prepare With Confidence

This PrepLogic Mega Guide was written by certified subject matter experts and published authors to provide you accurate, in-depth exam coverage. All exam objectives are covered in detail, giving you the knowledge and confidence you need to pass your exam.

PrepLogic

Be Prepared. Be Confident. Get Certified.

James Stanger - Author
Emmett Dulaney - Technical Editor

PrepLogic, Inc. • 1300 N. Westshore Boulevard, Suite # 125 • Tampa, Florida 33607
Phone: (813) 769-0920 (800) 418-6789 • Fax: (813) 769-0926 • Website: www.preplogic.com

Topic 101: Hardware & Architecture

Objective 1.101.1: Configure Fundamental BIOS Settings: The /proc/ file system

To fulfill this objective, candidates must be able to verify IRQ, DMA and I/O settings on the system. Stud ents also need to be able to change these settings. To perform these tasks, you first must understand the /proc/ file system.

The /proc/ file system

The /proc/ file system is created at each boot time. The files are mirrored (i.e., copied) from existing system settings and placed into the /proc/ directory.It reflects the system's current configuration and contains files and directories that the system uses to configure how it functions. Essential files and directories include those in Table 1:

File or Directory	Description
/proc/ioports	Provides detailed information concerning the hardware resources that the system is using. If a device is recognized, it will receive an I/O address. An example of an I/O address is: 03c0-03df. If the device does not have an I/O address, it has not been recognized.
/proc/interrupts	Describes the particular interrupt used by a device. Devices can share interrupts. If you are using a system with two CPUs, you will receive a listing of interrupts for each CPU.
/proc/dma	Describes channel that ISA and other devices register for Direct Memory Access (DMA).
/proc/pci	Contains information about operational PCI devices.
/proc/partitions	Provides detailed information about operational partitions.
/proc/cpuinfo	Statistics about the CPU(s) currently operating,
/proc/uptime	Shows how long the system has stayed operational. The first number shows uptime in seconds. The second number shows how many seconds the running system has been idle.
/proc/sys/net/ipv4/	Contains various files that allow you to view, enable or disable network resources. For example, changing the value of /proc/sys/net/ipv4/icmp_echo_ignore_all from 0 to 1 will prohibit the system from sending or receiving all ICMP packets, including those used by the "ping" command.

Table 1

To disable devices, you can take the following steps:

- Edit files in the /proc filesystem. Understand, however, that if you edit these files, the settings you generate will remain valid only until the next boot. You can, of course, create scripts to automatically alter files in the /proc/ directory.
- Change "jumper" settings on the devices you want to attach to the system.

As you add systems, verify that that there are no conflicting devices. For the exam, make sure that you understand the format of the files given in the above table. Also, make sure that you check on the /proc/ioports and /proc/interrupts files. It is possible for devices to share the same interrupt. However, they should not share the same I/O address.

Objective 1.101.1: Configure Fundamental BIOS Settings: Configuring Ports and Creating Headless Linux systems

Many times, an integrated peripheral can be enabled or disabled by going in to the BIOS settings and configuring the settings for each port. Ports for integrated peripherals can include:

- **Serial ports**: Often used to connect a "null modem" cable, which can be used to transfer data between systems, connect to a modem, or connect to a keyboard.
- **Parallel ports (IEEE 1284)**: Used in various situations, but most often to connect to a printing device. Although parallel ports are less used in production systems, it is important to understand that you can configure speeds for these ports. Available speeds include:
- **Compatibility mode**: The original specification, for simple 8-bit transfer.
- **Nibble mode**: Designed for printers, allows two-way communication at 4-bits at a time, but only one way at a time. In this way, transmission is unidirectional.
- **Byte mode**: A legacy mode that allows parallel ports to receive data more quickly. Disables the drivers that would place outgoing data onto the line.
- **Enhanced Parallel Port (EPP)**: Half-duplex communication, for non-printing devices.
- **Extended Capability Port (ECP)**: Enables half-duplex, bi-directional communication that allows compression. The fastest standard (2.5 Mbps).
- **SCSI devices**: Production Linux systems often use SCSI devices. The order in which the SCSI card is detected can determine the difference between a system that boots into Linux, and one that simply hangs.
- **Serial ATA (SATA) devices**: Serial devices include current hard drives.
- **IDE devices**: You can configure the order in which floppy, hard, CD and DVD drives are configured. Although floppy disks are very rare today, many servers still have them, and they are necessary for the initial booting of systems when performing updates. Also, IDE drives are in many older systems, which many beginners in Linux may be using. After all, many older systems that are just "lying around" can run Linux and are very useful in a home lab setting.
- **USB ports**: USB devices are not considered integrated devices by LPI. Nevertheless, you should understand that you can enable or disable the USB devices through BIOS.Headless Linux systems

There will also be times that you will have to use a "headless" Linux system. A "headless" system is a computer that does not have a keyboard or monitor attached. Sometimes, these headless systems are servers connected to a device that allows the server administrator or programmer to switch from one computer to the next. At other times, they are embedded Linux systems, where the operating system works behind the scenes, and there is no command-line or GUI interface for the user.

BIOS systems in headless computers require special configuration. For example, many x386 systems will not boot if a keyboard is not attached. To solve this problem, enter into the system's BIOS and configure it to ignore a missing keyboard.

You access a system's BIOS settings at boot during the boot time. During the boot time, the key you press to enter the system's BIOS configuration program differs from computer to computer. Sometimes, the F10 key will allow you to access the BIOS settings. With other systems, the DELETE key must be pressed. The system Power On Self Test (POST) screen will inform you.

Once you have accessed the BIOS configuration program, find the screen that allows the system to skip the keyboard check. Some systems will also cause problems if a graphics card is not installed. Similarly, find the settings that allow you to bypass checking for a card.

Troubleshooting headless systems

Below are some common issues when configuring headless system:

- **Keyboard and video will not work, even when they are attached:** Make sure that you have activated the serial port and that the mappings are correct in the /etc/inittab file.
- **The system won't boot to a GRUB or LILO prompt**: Many times, a system will not proceed if it detects that a keyboard is missing. If your system will not boot, make sure that you have disabled either the keyboard port, or that you have disabled the portion of the POST that checks for the keyboard. The option to look for is often called "Halt on," or "Keyboard halt."

Note: It is possible to use a null modem cable to test ports to make sure that they are usable.

Checking serial ports

When checking serial ports, use the dmesg command. You will be able to determine the ports that are running:

```
$ dmesg | grep ttyS
ttyS0 at 0x03f8 (irq = 4) is a 16550A
```

Communication programs

When communicating with the ports, you can use various applications, including:

minicom: The standard terminal.

gtkterm: A GUI-based alternative, based on Hyperterminal.

If you find that you can't log in as root on the terminal, the /etc/securetty file may not have been updated. Any terminal listed in this file will allow root logins. If the terminal is not listed, root cannot log in. To update the file, simply place the terminal name in it. You must be root. For example, if you wish to allow ttyS0 to allow root logins, you simply add the following line:

ttyS0

You may not wish to have the terminal to boot into X, assuming that the headless system is being used as a server. To have it boot into a standard terminal, enter the following in /etc/inittab:

id:3:initdefault:

Finally, it is possible to allow the serial terminal to communicate with the prompt issued by GRUB, enter the following lines in /boot/grub/grub.conf:

serial --unit=0 --speed=9600

terminal --timeout=2 serial console

1.101.3 Configure Modem and Sound cards: Using the setserial command

Make sure that you understand how to use the common commands to configure both a modem and sound card. The first step when configuring a modem or modem card is setting the proper port speed. To set the port speed, use the setserial command. A few examples are given below:

[root@jacob root]# setserial /dev/ttyS0 uart 16550A port 0x03f8 irq 4 baud_base 115200 spd_vhi

In the above command, you use the setserial command to configure the first terminal, called ttyS0. You can specify any port. The most common ports are ttyS0, ttyS1, ttyS2, and ttyS3. The additional commands specify the uart used, which in this case is the most common, 16550A. The UART is the Universal Asynchronous Receiver-Transmitter, which is responsible for managing serial transmissions between a computer and a peripheral, such as a modem. External modems rely on the UART that is placed on a PCI serial card, which is a modular component. If you are experiencing a problem with your UART, simply replace the serial card with an updated UART. Internal modems, however, have their own UART. In this case, you will have to replace the entire modem.

The command then specifies the I/O port address and IRQ to use (0x03f8 and 4, respectively). The baud base option sets the basic baud rate, which is usually set to 115200, the fastest baud rate the 16550A UART can support. Finally, the spd vhi option sets the UART to 115 Kbps. Additional setserial options are given below:

spd_normal: Tells the UART to run at 38.4 Kbps.

spd_hi: Sets the UART for 57.6 Kbps.

spd_vhi: Programs the UART to use a speed of 115 Kbps.

auto_irq: Use any available IRQ.

skip_test: Do not conduct a test of the UART.

autoconfig: Has the kernel automatically configure the serial port.

You can also use the setserial command to find out information about multiple ports on the system. To do this, use the -g option:

```
[root@jacob root]# setserial -g setserial -g /dev/ttyS*
```

The setserial command will then search all ports that begin with ttyS, and give the current settings. To disable a serial port, use the following command:

```
[root@jacob root]# setserial to set it to "uart none"
```

Winmodems

One of the most common legacy problems with Linux systems are modems that use proprietary software and the system's CPU to function properly. Such modems are not the industry standard, and they are called "Winmodems." Generally, these do not work with Linux systems. To detect a Winmodem, you can:

- Consult the Hardware Compatibility List (HCL) of your Linux distributor.
- Use the lspci command to view output concerning the device. If you see the text "Winmodem," purchase another modem.
- Internal modems can be Winmodems. External modems are never Winmodems. To solve the Winmodem problem, the best solution is to ignore the existing Winmodem and obtain an external modem.

Note: Winmodems are less of a problem now than they used to be 5 years ago. Still, consider the Winmodem problem if you are experiencing difficulties. Also, don't forget that you may be using a card with an older UART chip.

1.101.4 Setup non-IDE Devices: SCSI device names

The most common non-IDE device in Linux is the SCSI device. SCSI devices are serial devices and can be "daisy chained" together. Traditionally, SCSI devices have been considered to be faster and more reliable than SATA, and especially IDE devices. Many SCSI devices exist, including hard drives, printers, and scanners. In a production server environment, hard drives are the most common SCSI device. SATA devices have become much faster and reliable, recently, though SCSI remains an important standard.

SCSI device names

IDE and SATA devices are named after a certain pattern. The IDE device named /dev/hda is the first device. The device named /dev/hdb is the second device, and so forth. SCSI devices are named after the pattern shown in Table 2: