Table of Contents

LPIC 1 study guide in plain English

This book is originally released as a part of a series of free video tutorials for LPIC1 created by Jadi (in Persian language). But it is also possible to use it as a reference for self-study to prepare for LPIC1-101 & LPIC-102 exam.

If you want to download Farsi LPIC 101 & LPIC 102 exam videos use this torrent magnet link

Latest versions of this be book is always available at **http://j.mp/jadilpic1**

I've tried to write a short, easy to understand but still comprehensive exam guide. I'm not a native English speaker be ready for some possible language flaws.

Enjoy!

- Personal site: Jadi.ir
- Persian Weblog: jadi.net
- twitter: @jadi

Donations? http://jadi.ir/support-donate/ :D

101.1 Determine and configure hardware settings

Weight: 2

Candidates should be able to determine and configure fundamental system hardware.

Objectives

- Enable and disable integrated peripherals.
- Configure systems with or without external peripherals such as keyboards.
- Differentiate between the various types of mass storage devices.
- Set the correct hardware ID for different devices, especially the boot device.
- Know the differences between coldplug and hotplug devices.
- Determine hardware resources for devices.
- Tools and utilities to list various hardware information (e.g. lsusb, lspci, etc.)
- Tools and utilities to manipulate USB devices
- Conceptual understanding of sysfs, udev, hald, dbus
- /sys
- /proc
- /dev
- modprobe
- lsmod
- lspci
- lsusb

Find out about the hardware

HAL

HAL is Hardware Abstraction Layer. It abstracts your hardware details from you, say any first network card will be *eth0*. This way Linux will see any hardware as an *standard* hardware and you will be able to replace the hardware easily.

dbus

A line like a bus that connects all parts of the OS to each other. dbus lets different parts of the system to communicate with each other. For example, when you install a USB into your computer, dbus lets GNOME know about it. Using dbus, hardware & software can talk with each other.

udev

Supplies the software with the events and access info of devices and can handle rules.

There are a lot of devices in `/dev/` and if you plugin any device, it will have a file in `/dev` (say /dev/sdb2). **udev** lets you control what will be what in `/dev`. For example, you can use a rule to force your 8GB flash drive with one specific vendor to be `/dev/mybackup` all the time or you can tell it to copy all photos to your home directory as soon as your camera is connected.

sysfs

The `/sys` directory is where **HAL** keeps its database of everything connected to the system.

```
jadi@funlife:~$ ls /sys
block  bus  class  dev    devices  firmware  fs  hypervisor  kernel  module  power
```

All block devices are at the `block` and `bus` directory has all the connected PCI, USB, serial, .. devices. Note that here in `sys` we have the devices based on their technology but `/dev/` is abstracted.

proc directory

This is where kernel keeps its settings and properties. This directory is created on ram and files might have write accessible.