

Journeyman

Devices:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/dev-directory. The page title is "Devices". On the left, a sidebar lists numbered topics: 1. /dev directory, 2. device types, 3. Device Names, 4. sysfs, 5. udev, 6. lsusb, lspci, lsscsi, and 7. dd. The first topic, "1. /dev directory", is selected and expanded. The main content area contains text about the /dev directory, a terminal command box showing "\$ ls /dev", and two interactive boxes: "Ceremonias" and "Examen". The "Ceremonias" box asks about the contents of the /dev directory. The "Examen" box asks where device files are stored, with a correct answer of "/dev" and a "Revisar respuesta" button.

Filesystem Types:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/filesystem-types. The page title is "The Filesystem". On the left, a sidebar lists numbered topics: 1. Filesystem Hierarchy, 2. Filesystem Types, 3. Anatomy of a Disk, 4. Disk Partitioning, 5. Creating Filesystems, 6. mount and umount, 7. /etc/fstab, 8. swap, 9. Disk Usage, 10. Filesystem Repair, 11. Inodes, and 12. symlinks. The second topic, "2. Filesystem Types", is selected and expanded. The main content area contains text about filesystem types, a "Ceremonias" box asking for research on other filesystem types, and an "Examen" box asking for the common Linux filesystem type. The "Examen" box has a correct answer of "EXT4" and a "Revisar respuesta" button.

Anatomy of a Disk:

The screenshot shows a web browser window with the URL <https://linuxjourney.com/lesson/anatomy-of-a-disk>. The page title is "The Filesystem". The left sidebar has a navigation menu with items like "Filesystem Hierarchy", "Filesystem Types", "Anatomy of a Disk" (which is highlighted in green), "Disk Partitioning", "Creating Filesystems", etc. The main content area starts with a section titled "3. Anatomy of a Disk". It contains text about hard disks being subdivided into partitions, with examples like `/dev/sda1` and `/dev/sda2`. Below this is a "Partition Table" section with detailed information. To the right, there are two boxes: "Ceremonias" (with instructions to run `parted -l`) and "Examen" (a question about partition types). A green box indicates a correct answer ("¡Respuesta correcta!"). At the bottom, there are "Continuar" and "Revisar respuesta" buttons.

Disk Partitioning:

The screenshot shows a web browser window with the URL <https://linuxjourney.com/lesson/disk-partitioning>. The page title is "The Filesystem". The left sidebar has a navigation menu with items like "Filesystem Hierarchy", "Filesystem Types", "Anatomy of a Disk", "Disk Partitioning" (highlighted in green), "Creating Filesystems", etc. The main content area starts with a section titled "4. Disk Partitioning". It contains text about practical work with filesystems on a USB drive. Below this is a "Launch parted" section with a terminal input field containing `$ sudo parted`. Further down are sections for "Select the device" (with a terminal input field showing `select /dev/sdb2`) and "View current partition table" (with a terminal input field showing `(parted) print`). To the right, there are two boxes: "Ceremonias" (with instructions to partition a USB drive) and "Examen" (a question about the parted command). A green box indicates a correct answer ("¡Respuesta correcta!"). At the bottom, there are "Continuar" and "Revisar respuesta" buttons.

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/creating-filesystems. The left sidebar is titled 'The Filesystem' and contains a numbered list from 1 to 12. Item 5, 'Creating Filesystems', is highlighted with a teal background. The main content area has a header '5. Creating Filesystems'. It includes a note about partitioning and creating a filesystem, a command box containing '\$ sudo mkfs -t ext4 /dev/sdb2', and a text block explaining the simplicity of the mkfs tool. To the right, there are two boxes: 'Ceremonias' (Make an ext4 filesystem on the USB drive) and 'Examen' (What command is used to create a filesystem?). The 'Examen' box shows a correct answer ('mkfs') and a 'Revisar respuesta' button. The bottom of the screen shows a taskbar with various icons.

Mount and umount:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/mounting-and-unmounting-filesystems. The left sidebar is titled 'The Filesystem' and contains a numbered list from 1 to 12. Item 6, 'mount and umount', is highlighted with a teal background. The main content area has a header '6. mount and umount'. It includes a note about mounting a filesystem, a command box containing '\$ sudo mount -t ext4 /dev/sdb2 /mydrive', and a text block explaining the purpose of mount points. To the right, there are two boxes: 'Ceremonias' (Look at the manpage for mount and umount and see what other options you can use.) and 'Examen' (What command is used to attach a filesystem?). The 'Examen' box shows a correct answer ('mount') and a 'Revisar respuesta' button. The bottom of the screen shows a taskbar with various icons.

/etc/fstab:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/etc-fstab-file-system-table. The page title is "The Filesystem". On the left sidebar, under "The Filesystem", the "7. ./etc/fstab" section is highlighted. The main content area starts with a heading "7. ./etc/fstab" and a paragraph explaining what it is. It includes a code block showing a terminal session:

```
pete@icebox:~$ cat /etc/fstab
UUID=130b882f-7d79-436d-a096-1e594c92bb76 /
UUID=78d203a0-7c18-49bd-9e07-54f44cc5726 /home
UUID=22c3d34b-467e-467c-b44d-f03803c2c526 none
```

Below the code, there's a note about each line representing one filesystem and its fields. A list of fields is provided:

- UUID - Device identifier
- Mount point - Directory the filesystem is mounted to
- Filesystem type
- Options - other mount options, see manpage for more details
- Dump - used by the dump utility to decide when to make a backup, you should just default to 0
- Pass - Used by fsck to decide what order filesystems should be checked, if the value is 0, it will not be checked

There's also a note about adding entries directly to the file.

On the right side, there are two boxes: "Ceremonias" (with a note about adding a USB drive) and "Examen" (with a question about what file defines filesystems). Below the exam box is a green button saying "¡Respuesta correcta!" and a text input field containing "/etc/fstab".

Swap:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/swap-space. The page title is "The Filesystem". On the left sidebar, under "The Filesystem", the "8. swap" section is highlighted. The main content area starts with a heading "8. swap" and a paragraph about swap partitions. It includes a table:

Number	Start	End	Size	Type	File system
5	6861MB	7380MB	519MB	logical	linux-swap(v1)

Below the table, there's a paragraph explaining what swap is and how it works. A section titled "Using a partition for swap space" follows, with a note about setting up /dev/sdb2 for swap. A list of steps for enabling swap is provided:

1. First make sure we don't have anything on the partition
2. Run: mkswap /dev/sdb2 to initialize swap areas
3. Run: swapon /dev/sdb2 this will enable the swap device
4. If you want the swap partition to persist on bootup, you need to add an entry to the /etc/fstab file. sw is the filesystem type that you'll use.
5. To remove swap: swapoff /dev/sdb2

At the bottom, there's a note about allocating swap space relative to memory. On the right side, there are two boxes: "Ceremonias" (with a note about partitioning free space on a USB drive) and "Examen" (with a question about enabling swap). Below the exam box is a green button saying "¡Respuesta correcta!" and a text input field containing "swapon".

Disk Usage:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/disk-usage. The page content is as follows:

The Filesystem

- 1. Filesystem Hierarchy
- 2. Filesystem Types
- 3. Anatomy of a Disk
- 4. Disk Partitioning
- 5. Creating Filesystems
- 6. mount and umount
- 7. ./etc/fstab
- 8. swap
- 9. Disk Usage**
- 10. Filesystem Repair
- 11. Inodes
- 12. symlinks

9. Disk Usage

There are a few tools you can use to see the utilization of your disks:

```
pete@icebox:~$ df -h
Filesystem      1K-blocks   Used Available Use% Mounted
/dev/sda1        6.2G  2.3G    3.6G  40% /
```

The `df` command shows you the utilization of your currently mounted filesystems. The `-h` flag gives you a human readable format. You can see what the device is, and how much capacity is used and available.

Let's say your disk is getting full and you want to know what files or directories are taking up that space, for that you can use the `du` command.

```
$ du -h
```

This shows you the disk usage of the current directory you are in, you can take a peek at the root directory with `du -h` but that can get a little cluttered.

Both of these commands are so similar in syntax it can be hard to remember which one to use, to check how much of your disk is free use `df`. To check **disk usage**, use `du`.

Ceremonias

Look at your disk usage and free space with both `du` and `df`.

Examen

What command is used to show how much space is free on your disk?

¡Respuesta correcta!

df Revisar respuesta

Continuar

ES 06:13 p.m. 06/06/2025

Filesystem Repair:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/filesystem-repair. The page content is as follows:

The Filesystem

- 1. Filesystem Hierarchy
- 2. Filesystem Types
- 3. Anatomy of a Disk
- 4. Disk Partitioning
- 5. Creating Filesystems
- 6. mount and umount
- 7. ./etc/fstab
- 8. swap
- 9. Disk Usage
- 10. Filesystem Repair**
- 11. Inodes
- 12. symlinks

10. Filesystem Repair

Sometimes our filesystem isn't always in the best condition, if we have a sudden shutdown, our data can become corrupt. It's up to the system to try to get us back in a working state (although we sure can try ourselves).

The `fsck` (filesystem check) command is used to check the consistency of a filesystem and can even try to repair it for us. Usually when you boot up a disk, `fsck` will run before your disk is mounted to make sure everything is ok. Sometimes though, your disk is so bad that you'll need to manually do this. However, be sure to do this while you are in a rescue disk or somewhere where you can access your filesystem without it being mounted.

```
$ sudo fsck /dev/sda
```

Ceremonias

Look at the manpage for `fsck` to see what else it can do.

Examen

What command is used to check the integrity of a filesystem?

¡Respuesta correcta!

fsck Revisar respuesta

Continuar

ES 06:14 p.m. 06/06/2025

Filesystem Hierarchy

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/filesystem-hierarchy. The page title is "Filesystem Hierarchy". On the left, a sidebar lists topics: 1. Filesystem Hierarchy (selected), 2. Filesystem Types, 3. Anatomy of a Disk, 4. Disk Partitioning, 5. Creating Filesystems, 6. mount and umount, 7. /etc/fstab, 8. swap, 9. Disk Usage, 10. Filesystem Repair, 11. Inodes, 12. symlinks. The main content area has a heading "1. Filesystem Hierarchy". It contains text about the directory structure and a bulleted list of root directory components:

- / - The root directory of the entire filesystem hierarchy, everything is nested under this directory.
- /bin - Essential ready-to-run programs (binaries), includes the most basic commands such as ls and cp.
- /boot - Contains kernel boot loader files.
- /dev - Device files.
- /etc - Core system configuration directory, should hold only configuration files and not any binaries.
- /home - Personal directories for users, holds your documents, files, settings, etc.
- /lib - Holds library files that binaries can use.
- /media - Used as an attachment point for removable media like USB drives.
- /mnt - Temporarily mounted filesystems.
- /opt - Optional application software packages.
- /proc - Information about currently running processes.
- /root - The root user's home directory.
- /run - Information about the running system since the last boot.
- /sbin - Contains essential system binaries, usually can only be run by root.
- /srv - Site-specific data which are served by the

On the right, there are two sections: "Ceremonias" and "Examen". The "Ceremonias" section asks: "Look inside your /usr directory, what kind of information is located there?". The "Examen" section asks: "What directory is used to store logs?". A response box contains "var" and a "Revisar respuesta" button. A green feedback box says "¡Respuesta correcta!". The bottom status bar shows "ES" and the date "06/06/2025".

Filesystem Types:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/filesystem-types. The page title is "Filesystem Types". On the left, a sidebar lists topics: 1. Filesystem Hierarchy, 2. Filesystem Types (selected), 3. Anatomy of a Disk, 4. Disk Partitioning, 5. Creating Filesystems, 6. mount and umount, 7. /etc/fstab, 8. swap, 9. Disk Usage, 10. Filesystem Repair, 11. Inodes, 12. symlinks. The main content area has a heading "2. Filesystem Types". It contains text about different filesystem implementations and a section on "Journaling". The "Journaling" section explains that journaling comes by default on most filesystem types, but if it doesn't, you should know what it does. It describes how journaling prevents data loss during power outages. The text also mentions that if a journaled filesystem becomes inconsistent, it will perform a filesystem check to fix it. Finally, it notes that journaling writes log entries before copying files.

On the right, there are two sections: "Ceremonias" and "Examen". The "Ceremonias" section asks: "Do a little bit of research online on the other filesystem types: ReiserFS, ZFS, JFS and others you can find.". The "Examen" section asks: "What is the common Linux filesystem type?". A response box contains "Ext4" and a "Revisar respuesta" button. A green feedback box says "¡Respuesta correcta!". The bottom status bar shows "ES" and the date "06/06/2025".

Anatomy of a Disk:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/anatomy-of-a-disk. The page content is as follows:

3. Anatomy of a Disk

Hard disks can be subdivided into partitions, essentially making multiple block devices. Recall such examples as, /dev/sda1 and /dev/sda2, /dev/sda is the whole disk, but /dev/sda1 is the first partition on that disk. Partitions are extremely useful for separating data and if you need a certain filesystem, you can easily create a partition instead of making the entire disk one filesystem type.

Partition Table

Every disk will have a partition table, this table tells the system how the disk is partitioned. This table tells you where partitions begin and end, which partitions are bootable, what sectors of the disk are allocated to what partition, etc. There are two main partition table schemes used, Master Boot Record (MBR) and GUID Partition Table (GPT).

Partition

Disks are comprised of partitions that help us organize our data. You can have multiple partitions on a disk and they can't overlap each other. If there is space that is not allocated to a partition, then it is known as free space. The types of partitions depend on your partition table. Inside a partition, you can have a filesystem or dedicate a partition to other things like swap (we'll get to that soon).

MBR

- Traditional partition table, was used as the standard
- Can have primary, extended, and logical partitions
- MBR has a limit of four primary partitions
- Additional partitions can be made by making a primary

Ceremonias

Run parted -l on your machine and evaluate your results.

Examen

What partition type is used to create more than 4 partitions in the MBR partitioning scheme?

Respuesta correcta!

extended

Continuar

ES 06:34 p.m. 06/06/2025

Disk Partitioning:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/disk-partitioning. The page content is as follows:

4. Disk Partitioning

Let's do some practical stuff with filesystems by working through the process on a USB drive. If you don't have one, no worries, you can still follow along these next couple of lessons.

First we'll need to partition our disk. There are many tools available to do this:

- fdisk - basic command-line partitioning tool, it does not support GPT
- parted - this is a command line tool that supports both MBR and GPT partitioning
- gparted - this is the GUI version of parted
- gdisk - fdisk, but it does not support MBR only GPT

Let's use parted to do our partitioning. Let's say I connect the USB device and we see the device name is /dev/sdb2.

Launch parted

\$ sudo parted

You'll be entered in the parted tool, here you can run commands to partition your device.

Select the device

select /dev/sdb2

To select the device you'll be working with, select it by its device name.

View current partition table

Ceremonias

Partition a USB drive with half of the drive as ext4 and the other half as free space.

Examen

What is the parted command to make a partition?

Respuesta correcta!

mkpart

Continuar

ES 06:34 p.m. 06/06/2025

Creating Filesystems:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/creating-filesystems. The page title is "The Filesystem". On the left sidebar, under "5. Creating Filesystems", the "mkfs" command is demonstrated with the code: `$ sudo mkfs -t ext4 /dev/sdb2`. The main content area shows a section titled "Ceremonias" with the task "Make an ext4 filesystem on the USB drive." Below it is an "Examen" section asking "What command is used to create a filesystem?". A green box indicates the answer "mkfs" is correct. The status bar at the bottom right shows the date and time as 06/06/2025.

mount and umount:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/mounting-and-unmounting-filesystems. The page title is "The Filesystem". On the left sidebar, under "6. mount and umount", the "mount" command is demonstrated with the code: `$ sudo mount -t ext4 /dev/sdb2 /mydrive`. The main content area shows a section titled "Ceremonias" with the task "Look at the manpage for mount and umount and see what other options you can use." Below it is an "Examen" section asking "What command is used to attach a filesystem?". A green box indicates the answer "mount" is correct. The status bar at the bottom right shows the date and time as 06/06/2025.

Inodes:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/inodes. The left sidebar lists 'The Filesystem' topics, with '11. Inodes' highlighted in green. The main content area displays the '11. Inodes' page, which includes sections on 'What is an inode?', 'When are inodes created?', and 'Ceremonias'. A 'Examen' section contains a question about inode counts and a correct answer input field. The bottom status bar shows system icons and the date/time: 06/06/2025, 06:15 p.m.

Symlinks:

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/symlinks. The left sidebar lists 'The Filesystem' topics, with '12. symlinks' highlighted in green. The main content area displays the '12. symlinks' page, which includes sections on 'Symlinks' and 'Ceremonias'. A 'Examen' section contains a question about the command to make a symlink and a correct answer input field. The bottom status bar shows system icons and the date/time: 06/06/2025, 06:17 p.m.

BOOT THE SYSTEM

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/boot-process-overview. The page title is "Boot the System". On the left, a sidebar lists "1. Boot Process Overview", "2. Boot Process: BIOS", "3. Boot Process: Bootloader", "4. Boot Process: Kernel", and "5. Boot Process: Init". The main content area is titled "1. Boot Process Overview". It contains text about the boot process overview, mentioning the BIOS, bootloader, kernel, and init stages. A sidebar on the right contains sections for "Ceremonias" and "Examen". The "Ceremonias" section asks to reboot the system to spot each step. The "Examen" section asks what the last stage in the Linux boot process is, with the answer "init" being correct. A "Continuar" button is at the bottom.

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/boot-process-bios. The page title is "Boot the System". On the left, a sidebar lists "1. Boot Process Overview", "2. Boot Process: BIOS", "3. Boot Process: Bootloader", "4. Boot Process: Kernel", and "5. Boot Process: Init". The main content area is titled "2. Boot Process: BIOS". It contains text about the BIOS booting process, MBR vs GPT, and UEFI. A sidebar on the right contains sections for "Ceremonias" and "Examen". The "Ceremonias" section asks to go into the BIOS menu to see if UEFI booting is enabled. The "Examen" section asks what the BIOS loads, with the answer "bootloader" being correct. A "Continuar" button is at the bottom.

Home | Linux Journey Learning | Linux Journey

linuxjourney.com/lesson/boot-process-bootloader

WhatsApp Web https://discord.com... SKZFLIX Pinterest Notion – The all-in... Nueva pestaña

Otros marcadores

Boot the System

- 1. Boot Process Overview
- 2. Boot Process: BIOS
- 3. Boot Process: Bootloader**
- 4. Boot Process: Kernel
- 5. Boot Process: Init

3. Boot Process: Bootloader

The bootloader's main responsibilities are:

- Booting into an operating system, it can also be used to boot to non-Linux operating systems
- Select a kernel to use
- Specify kernel parameters

The most common bootloader for Linux is GRUB, you are most likely using it on your system. There are many other bootloaders that you can use such as LILO, eflinux, coreboot, SYSLINUX and more. However, we will just be working with GRUB as our bootloader.

So we know that the bootloader's main goal is to load up the kernel, but where does it find the kernel? To find it, we will need to look at our kernel parameters. The parameters can be found by going into the GRUB menu on startup using the 'e' key. If you don't have GRUB no worries, we'll go through the boot parameters that you will see:

- initrd - Specifies the location of initial RAM disk (we'll talk more about this in the next lesson).
- BOOT_IMAGE - This is where the kernel image is located
- root - The location of the root filesystem, the kernel searches inside this location to find init. It is often represented by its UUID or the device name such as /dev/sda1.
- ro - This parameter is pretty standard, it mounts the filesystem as read-only mode.
- quiet - This is added so that you don't see display messages that are going on in the background during boot.

<https://linuxjourney.com/lesson/boot-process-kernel> splash - This lets the splash screen be shown.

Ceremonias

If you have GRUB as your bootloader, go into the GRUB menu with 'e' and take a look at the settings.

Examen

What kernel parameter makes it so you don't see bootup messages?

Respuesta correcta!

quiet

Revisar respuesta

Continuar

Home | Linux Journey Learning | Linux Journey

linuxjourney.com/lesson/boot-process-kernel

WhatsApp Web https://discord.com... SKZFLIX Pinterest Notion – The all-in... Nueva pestaña

Otros marcadores

Boot the System

- 1. Boot Process Overview
- 2. Boot Process: BIOS
- 3. Boot Process: Bootloader
- 4. Boot Process: Kernel**
- 5. Boot Process: Init

4. Boot Process: Kernel

So now that our bootloader has passed on the necessary parameters, let's see how it gets started:

Initrd vs Initramfs

There is a bit of a chicken and egg problem when we talk about the kernel bootup. The kernel manages our system's hardware, however not all drivers are available to the kernel during bootup. So we depend on a temporary root filesystem that contains just the essential modules that the kernel needs to get to the rest of the hardware. In older versions of Linux, this job was given to the initrd (initial ram disk). The kernel would mount the initrd, get the necessary bootup drivers, then when it was done loading everything it needed, it would replace the initrd with the actual root filesystem. These days, we have something called the initramfs, this is a temporary root filesystem that is built into the kernel itself to load all the necessary drivers for the real root filesystem, so no more locating the initrd file.

Mounting the root filesystem

Now the kernel has all the modules it needs to create a root device and mount the root partition. Before you go any further though, the root partition is actually mounted in read-only mode first so that fsck can run safely and check for system integrity. Afterwards it remounts the root filesystem in read-write mode. Then the kernel locates the init program and executes it.

Ceremonias

No exercises for this lesson.

Examen

What is used in modern systems to load up a temporary root filesystem?

Respuesta correcta!

initramfs

Revisar respuesta

Continuar

<https://linuxjourney.com>

5. Boot Process: Init

We've discussed init in previous lessons and know that it is the first process that gets started and it starts all the other essential services on our system. But how?

There are actually three major implementations of init in Linux:

System V init (sysv)

This is the traditional init system. It sequentially starts and stops processes, based on startup scripts. The state of the machine is denoted by runlevels, each runlevel starts or stops a machine in a different way.

Upstart

This is the init you'll find on older Ubuntu installations. Upstart uses the idea of jobs and events and works by starting jobs that performs certain actions in response to events.

Systemd

This is the new standard for init, it is goal oriented. Basically you have a goal that you want to achieve and systemd tries to satisfy the goal's dependencies to complete the goal.

We have an entire course on init systems where we will dive into each of these systems in more detail.

Ceremonias

No exercises for this lesson.

Examen

What is the newest standard for init?

✓ ¡Respuesta correcta!

systemd Revisar respuesta

Return

translation missing: es.courses.messages.return

Kernel

<https://linuxjourney.com>

1. Overview of the Kernel

As you've learned up to this point, the kernel is the core of the operating system. We've talked about the other parts of the operating system but have yet to show how they all work together. The Linux operating system can be organized into three different levels of abstraction.

The most basic level is hardware, this includes our CPU, memory, hard disks, networking ports, etc. The physical layer that actually computes what our machine is doing.

The next level is the kernel, which handles process and memory management, device communication, system calls, sets up our filesystem, etc. The kernel's job is to talk to the hardware to make sure it does what we want our processes to do.

And the level that you are familiar with is the user space, the user space includes the shell, the programs that you run, the graphics, etc.

In this course, we'll be focusing on the kernel and learning its intricacies.

Ceremonias

No exercises for this lesson.

Examen

What level of the operating system manages devices?

✓ ¡Respuesta correcta!

kernel Revisar respuesta

Continuar

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/kernel-privilege-levels. The left sidebar is titled 'Kernel' and contains the following numbered items:

1. Overview of the Kernel
- 2. Privilege Levels**
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules

The main content area is titled '2. Privilege Levels'. It starts with a paragraph about why there are different abstraction layers between user space and kernel. It then discusses privilege levels, comparing them to Britney Spears being protected by security personnel. It explains that there are two main levels or modes in x86 architecture: Ring #3 (user mode) and Ring #0 (kernel mode). A note at the bottom states: 'These different modes are called privilege levels (aptly named for the levels of privilege you get) and are often described as protection rings. To make this picture easier to paint, let's say you find out that Britney Spears is in town at your local klerb, she's protected by her groupies, then her personal bodyguards, then the bouncer outside the klerb. You want to get her autograph (because why not?), but you can't get to her because she is heavily protected. The rings work the same way, the innermost ring corresponds to the highest privilege level. There are two main levels or modes in an x86 computer architecture. Ring #3 is the privilege that user mode applications run in. Ring #0 is the privilege that the kernel'.

To the right, there are two sections: 'Ceremonias' (No exercises for this lesson) and 'Examen' (What ring number has the highest privileges?). The exam section includes a green box saying '¡Respuesta correcta!', a text input field with '0', a 'Revisar respuesta' button, and an orange 'Continuar' button.

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/system-calls. The left sidebar is titled 'Kernel' and contains the following numbered items:

1. Overview of the Kernel
2. Privilege Levels
- 3. System Calls**
4. Kernel Installation
5. Kernel Location
6. Kernel Modules

The main content area is titled '3. System Calls'. It begins with a paragraph about system calls being like VIP passes to a secret side door. It then explains what system calls are and how they provide user space processes with services from the kernel. A note at the bottom states: 'I won't get into specifics of system calls, as that will require you to know a bit of C, but the basics is that when you call a program like ls, the code inside this program contains a system call wrapper (so not the actual system call yet). Inside this wrapper it invokes the system call which will execute a trap, this trap then gets caught by the system call handler and then references the system call in the system call table. Let's say we are trying to call the stat() system call, it's identified by a syscall ID and the purpose of the stat() system call is to query the status of a file. Now remember, you were running the ls program in non-privilege mode. So now it sees you're trying to make a syscall, it then switches you over to kernel mode, there it does lots of things but most importantly it looks up your syscall number, finds it in a table based on the syscall ID and then executes the function you wanted to run. Once it's done, it will return back to user mode and your process will receive a return status if it was successful or if it had an error.'

To the right, there are two sections: 'Ceremonias' (No exercises for this lesson) and 'Examen' (What is used to switch from user mode to kernel mode?). The exam section includes a green box saying '¡Respuesta correcta!', a text input field with 'System call', a 'Revisar respuesta' button, and an orange 'Continuar' button.

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/kernel-installation. The page is titled "4. Kernel Installation". On the left, a sidebar menu lists "Kernel" topics: 1. Overview of the Kernel, 2. Privilege Levels, 3. System Calls, 4. Kernel Installation (highlighted in green), 5. Kernel Location, and 6. Kernel Modules. The main content area contains text and code snippets. A "Ceremonias" section lists tasks: 1. Find out what kernel version you have. 2. Research the different versions of kernels available. An "Examen" section asks, "How do you see the kernel version of your system?" with a correct answer of "uname -r" and a "Revisar respuesta" button.

Kernel

1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules

4. Kernel Installation

Ok, now that we've got all that boring stuff out of the way, let's talk about actually installing and modifying kernels. You can install multiple kernels on your system, remember in our lesson on the boot process? In our GRUB menu we can choose which kernel to boot to.

To see what kernel version you have on your system, use the following command:

```
$ uname -r  
3.19.0-43-generic
```

The uname command prints system information, the -r command will print out all of the kernel release version.

You can install the Linux kernel in different ways, you can download the source package and compile from source or you can install it using package management tools.

```
$ sudo apt install linux-generic-lts-vivid
```

and then just reboot into the kernel you installed. Simple right? Kind of, you'll need to also install other linux packages such as the linux-headers, linux-image-generic, etc). You can also specify the version number, so the above command can look like, `sudo apt install 3.19.0-43-generic`

Alternatively, if you just want the updated kernel version, just use dist-upgrade, it performs upgrades to all package on your system:

```
$ sudo apt dist-upgrade
```

There are many different kernel versions, some are used as LTS (long term support) some are the latest and greatest, the

Ceremonias

1. Find out what kernel version you have.
2. Research the different versions of kernels available

Examen

How do you see the kernel version of your system?

✓ ¡Respuesta correcta!

uname -r Revisar respuesta

Continuar

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/kernel-location. The page is titled "5. Kernel Location". The sidebar menu is identical to the previous screenshot. The main content area contains text and code snippets. A "Ceremonias" section instructs users to go into their /boot directory and see what files are there. An "Examen" section asks, "What is the kernel image called in /boot?" with a correct answer of "vmlinuz" and a "Revisar respuesta" button.

Kernel

1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules

5. Kernel Location

What happens when you install a new kernel? Well it actually adds a couple of files to your system, these files are usually added to the /boot directory.

You will see multiple files for different kernel versions:

- vmlinuz - this is the actual linux kernel
- initrd - as we've discussed before, the initrd is used as a temporary file system, used before loading the kernel
- System.map - symbolic lookup table
- config - kernel configuration settings, if you are compiling your own kernel, you can set which modules can be loaded

If your /boot directory runs out of space, you can always delete old versions of these files or just use a package manager, but be careful when doing maintenance in this directory and don't accidentally delete the kernel you are using.

Ceremonias

Go into your boot directory and see what files are in there.

Examen

What is the kernel image called in /boot?

✓ ¡Respuesta correcta!

vmlinuz Revisar respuesta

Continuar

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/kernel-modules. The page title is "Learning | Linux Journey". The left sidebar has a "Kernel" category with numbered links from 1 to 6. Link 6, "Kernel Modules", is highlighted with a teal background. The main content area starts with a section titled "6. Kernel Modules". It contains text explaining kernel modules and their purpose. Below this are several code snippets and instructions:

- View a list of currently loaded modules**:
\$ lsmod
- Load a module**:
\$ sudo modprobe bluetooth
- Modprobe loads tries the module from /lib/modules/(kernel version)/kernel/drivers**. Kernel modules may also have dependencies, modprobe loads our module dependencies if they are not already loaded.
- Remove a module**:
\$ sudo modprobe -r bluetooth
- Load on bootup**

To the right, there are two boxes: "Ceremonias" and "Examen".

Ceremonias: Unload your bluetooth module with modprobe and see what happens. How will you fix this?

Examen: What command is used to unload a module?

Response: modprobe -r
Result: ✓ ¡Respuesta correcta!

Return

Init

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/sysv-overview. The page title is "Learning | Linux Journey". The left sidebar has an "Init" category with numbered links from 1 to 7. Link 1, "System V Overview", is highlighted with a teal background. The main content area starts with a section titled "1. System V Overview". It contains text about the purpose of init, different implementations in Linux, and the traditional System V init. Below this is a note about /etc/inittab and a detailed explanation of how Sys V starts services sequentially. A note at the bottom discusses power states.

Para saber si estás utilizando la implementación Sys V de init, verifica si tienes el fichero /etc/inittab.

Sys V inicia y detiene procesos de forma secuencial, por lo que si quieres iniciar un servicio llamado foo-b, antes de que foo-b pueda funcionar, tienes que asegurarte de que foo-a ya se encuentra corriendo. Sys V hace esto mediante scripts, los cuales inician y detienen servicios por nosotros, podemos escribir nuestros propios scripts o usar los que ya vienen implementados en el sistema operativo y que son usados para cargar servicios esenciales.

La ventaja de utilizar esta implementación de init, radica en la relativa facilidad para resolver dependencias, ya que sabes que foo-a viene antes de foo-b, sin embargo la performance no es muy buena ya que generalmente una cosa es iniciada o detenida a la vez.

Cuando usas Sys V, el estado de la máquina es definido por una serie de niveles de ejecución que van del 0 al 6. Estos niveles varían dependiendo de la distribución, pero la mayoría de las veces se verán de la siguiente manera:

- 0: Apagar
- 1: Modo monousuario
- 2: Modo multiusuario sin soporte de red
- 3: Modo multiusuario con soporte de red
- 4: No usado

To the right, there are two boxes: "Ceremonias" and "Examen".

Ceremonias: Si estás corriendo System V init, cambia el nivel de ejecución por defecto de tu máquina a otra cosa y ve que sucede.

Examen: ¿Qué nivel de ejecución es usado generalmente para apagar el sistema?

Response: 0
Result: ✓ ¡Respuesta correcta!

Continuar

Learning | Linux Journey

linuxjourney.com/lesson/sysv-services

WhatsApp Web SKZFLIX Pinterest Notion – The all-in... Nueva pestaña Otros marcadores

Init

1. System V Overview

2. System V Service

3. Upstart Overview

4. Upstart Jobs

5. Systemd Overview

6. Systemd Goals

7. Power States

2. System V Service

Hay varias herramientas de línea de comando que puedes utilizar para gestionar los servicios de Sys V.

Listar servicios

```
$ service --status-all
```

Iniciar un servicio

```
$ sudo service networking start
```

Detener un servicio

```
$ sudo service networking stop
```

Reiniciar un servicio

```
$ sudo service networking restart
```

Estos comandos no son solo específicos de sistemas que usan Sys V, puedes utilizarlos también para gestionar servicios de Upstart. Aunque Linux está tratando de alejarse de los scripts más tradicionales de Sys V, todavía hay herramientas para ayudar en esa transición.

Ceremonias

Elige un par de servicios y cambia su estado. ¿Qué observas?

Examen

¿Cuál es el comando para detener un servicio llamado peanut usando Sys V?

✓ ¡Respuesta correcta!

sudo service peanut stop

Revisar respuesta

Continuar

Learning | Linux Journey

linuxjourney.com/lesson/upstart-overview

WhatsApp Web SKZFLIX Pinterest Notion – The all-in... Nueva pestaña Otros marcadores

Init

1. System V Overview

2. System V Service

3. Upstart Overview

4. Upstart Jobs

5. Systemd Overview

6. Systemd Goals

7. Power States

pete@icebox:~\$ ls /etc/init

acpid.conf	mountnfs.sh.conf
alsa-restore.conf	mtab.sh.conf
alsa-state.conf	networking.conf
alsa-store.conf	network-interface.conf
anacron.conf	network-interface-conta

Dentro de estas configuraciones, se incluye información sobre como y cuando iniciar los trabajos.

Por ejemplo, el fichero networking.conf, podría decir algo sencillo como:

```
start on runlevel [235]
stop on runlevel [0]
```

Esto significa que iniciará la configuración de red en el nivel de ejecución 2, 3 o 5 y lo detendrá en el nivel 0. Hay muchas formas de escribir los ficheros de configuración y es algo que descubrirás cuando eches un vistazo a las diferentes configuraciones disponibles.

El modo en el que Upstart trabaja es el siguiente:

1. Primero, carga las configuraciones de los trabajos desde /etc/init.
2. Cuando un evento de inicio ocurra, correrá el trabajo disparado por dicho evento.
3. Estos trabajos generarán nuevos eventos y esos eventos dispararán más trabajos.
4. Upstart continuará haciendo esto hasta que complete todos los trabajos necesarios.

Ceremonias

Si estás utilizando Upstart, trata de entender las configuraciones de los trabajos en /etc/init.

Examen

¿Cuál es la implementación de init que es utilizada en Ubuntu?

✓ ¡Respuesta correcta!

Upstart

Revisar respuesta

Continuar

Learning | Linux Journey

linuxjourney.com/lesson/upstart-jobs

WhatsApp Web https://discord.com... SKZFLIX Pinterest Notion – The all-in-... Nueva pestaña Otros marcadores

Init

1. System V Overview

2. System V Service

3. Upstart Overview

4. Upstart Jobs

5. Systemd Overview

6. Systemd Goals

7. Power States

nombre del trabajo, el segundo (antes de /) es el objetivo del mismo y el tercero (después de /) es el estado actual. Puedes ver que nuestro trabajo shutdown tiene como objetivo detenerse, pero actualmente está en estado de espera. Los estados y objetivos cambiarán a medida que inicies o detengas trabajos.

Ver un trabajo en específico

```
initctl status networking  
networking start/running
```

No nos meteremos en los detalles de como escribir la configuración de un trabajo, sin embargo ya sabes que los mismos pueden ser detenidos, iniciados y reiniciados en estas configuraciones. Los trabajos también emiten eventos, por lo que pueden iniciar otros trabajos. A continuación veremos los comandos para trabajar con Upstart, pero si eres curioso, no estaría mal que te interiorizaras con los ficheros .conf.

Iniciar un trabajo de forma manual

```
$ sudo initctl start networking
```

Detener un trabajo de forma manual

```
$ sudo initctl stop networking
```

Reiniciar un trabajo de forma manual

```
$ sudo initctl restart networking
```

Emitir un evento de forma manual

```
$ sudo initctl emit some_event
```

Ceremonias

Observa tu lista de trabajos y cambia el estado de alguno con los comandos que aprendimos hoy. ¿Qué notas?

Examen

¿Como reiniciarías de forma manual un trabajo llamado peanuts?

¡Respuesta correcta!

```
sudo initctl restart peanuts
```

Revisar respuesta

Continuar

Learning | Linux Journey

linuxjourney.com/lesson/systemd-overview

WhatsApp Web https://discord.com... SKZFLIX Pinterest Notion – The all-in-... Nueva pestaña Otros marcadores

Init

1. System V Overview

2. System V Service

3. Upstart Overview

4. Upstart Jobs

5. Systemd Overview

6. Systemd Goals

7. Power States

página de inicio

5. Systemd Overview

Systemd se está convirtiendo lentamente en el standard de init. Si tienes el directorio /usr/lib/systemd, lo más probable es que lo estés utilizando.

Systemd utiliza objetivos para lograr que el sistema esté listo y en funcionamiento. Basicamente, tienes un objetivo que quieres conseguir y este objetivo a su vez tiene dependencias con las que tenemos que cumplir. Systemd es extremadamente flexible y robusto, no sigue una secuencia estricta para iniciar procesos. Esto es lo que ocurre durante un arranque típico de systemd:

1. Primero, systemd carga sus archivos de configuración, generalmente ubicados en /etc/systemd/system o /usr/lib/systemd/system
2. Luego determina el objetivo de arranque, por lo general es default.target.
3. Systemd determina las dependencias del objetivo de arranque y las activa.

De manera similar a los niveles de ejecución de Sys V, systemd inicia dentro de diferentes objetivos:

- poweroff.target - apagar sistema
- rescue.target - modo monousuario
- multi-user.target - modo multiusuario con soporte de red
- graphical.target - modo multiusuario con soporte de red e interfaz gráfica
- reboot.target - reiniciar sistema

Por defecto el objetivo de arranque de defaults.target apunta generalmente a graphical.target.

El componente básico de systemd es la unidad. Systemd no

Ceremonias

No hay ejercicios para esta lección.

Examen

¿Qué unidad es utilizada para agrupar otras unidades?

¡Respuesta correcta!

```
target
```

Revisar respuesta

Continuar

Learning | Linux Journey

linuxjourney.com/lesson/systemd-goals

WhatsApp Web SKZFLIX Pinterest Notion – The all-in... Nueva pestaña Otros marcadores

Init

- 1. System V Overview
- 2. System V Service
- 3. Upstart Overview
- 4. Upstart Jobs
- 5. Systemd Overview
- 6. Systemd Goals**
- 7. Power States

[Install] es utilizada para las dependencias. Esto es solamente la punta del iceberg sobre como escribir ficheros de systemd, por lo que te imploro que leas sobre este tema si quieras saber mas.

Ahora, metámosnos con algunos comandos que podemos usar con unidades de systemd:

Listar unidades

```
$ systemctl list-units
```

Ver el estado de una unidad

```
$ systemctl status networking.service
```

Iniciar una unidad

```
$ sudo systemctl start networking.service
```

Detener una unidad

```
$ sudo systemctl stop networking.service
```

Reiniciar una unidad

```
$ sudo systemctl restart networking.service
```

Habilitar una unidad

```
$ sudo systemctl enable networking.service
```

Deshabilitar una unidad

```
$ sudo systemctl disable networking.service
```

Aun asi, todavia tienes que ver que tan profundo puede llegar a ser systemd, investiga si quieras aprender mas.

Ceremonias

Observa el estado de las unidades e inicia y deten algunos servicios ¿Que notas?

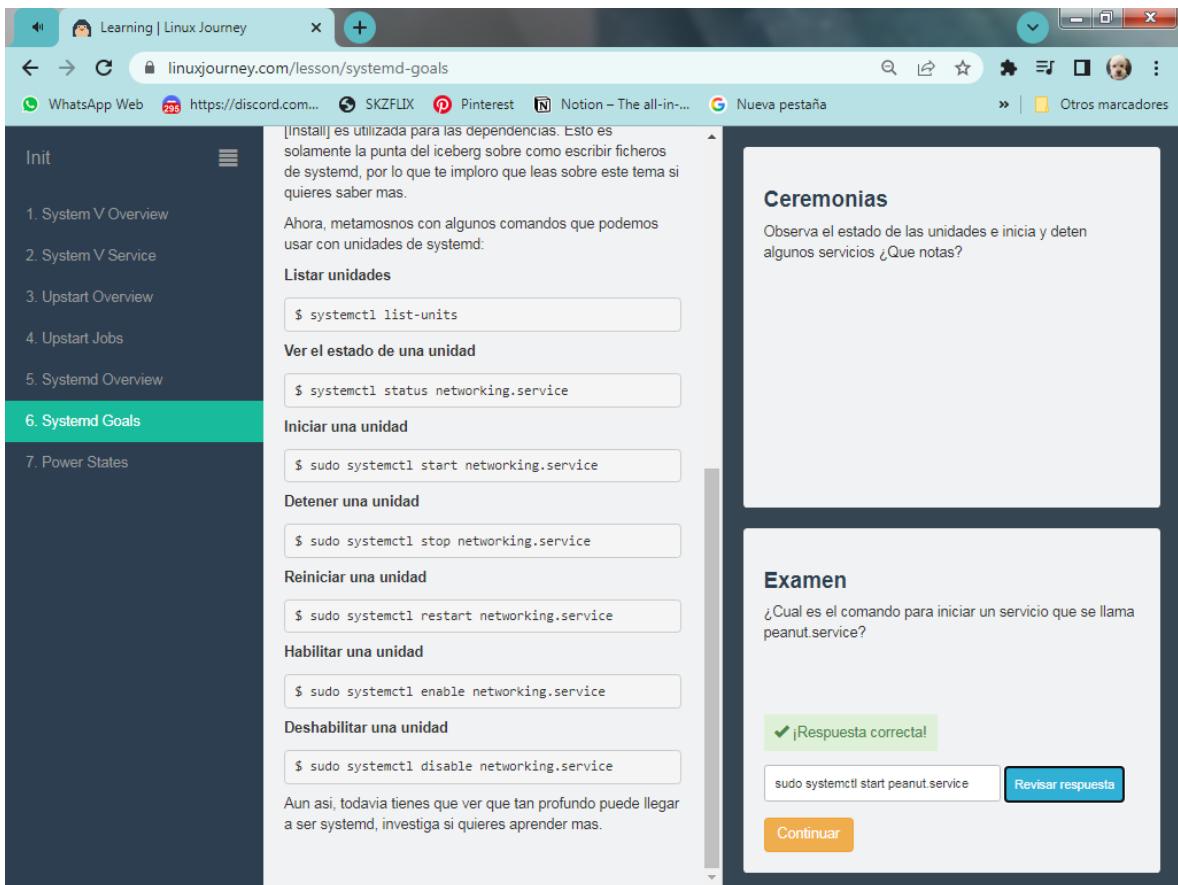
Examen

¿Cuál es el comando para iniciar un servicio que se llama peanut service?

✓ ¡Respuesta correcta!

sudo systemctl start peanut.service Revisar respuesta

Continuar



Learning | Linux Journey

linuxjourney.com/lesson/power-states

WhatsApp Web SKZFLIX Pinterest Notion – The all-in... Nueva pestaña Otros marcadores

Init

- 1. System V Overview
- 2. System V Service
- 3. Upstart Overview
- 4. Upstart Jobs
- 5. Systemd Overview
- 6. Systemd Goals
- 7. Power States**

7. Power States

Es difícil de creer que hasta ahora no hayamos discutido las diferentes maneras de controlar el estado del sistema desde la línea de comando, cuando hablamos de init, no solo hablamos de los modos que nos permiten iniciar nuestro sistema, sino tambien de aquellos que lo detienen.

Detener el sistema

```
$ sudo shutdown -h now
```

Esto detendrá el sistema (apagarlo), además debes especificar el momento en el que quieras que esta acción ocurría. Puedes expresar el tiempo en minutos.

```
$ sudo shutdown -h +2
```

Esto hará que el sistema se apague dentro de dos minutos. También puedes reiniciar el sistema con el comando shutdown:

```
$ sudo shutdown -r now
```

O simplemente usar el comando reboot.

```
$ sudo reboot
```

Ceremonias

¿Qué piensas que ocurre con init cuando apagas tu equipo?

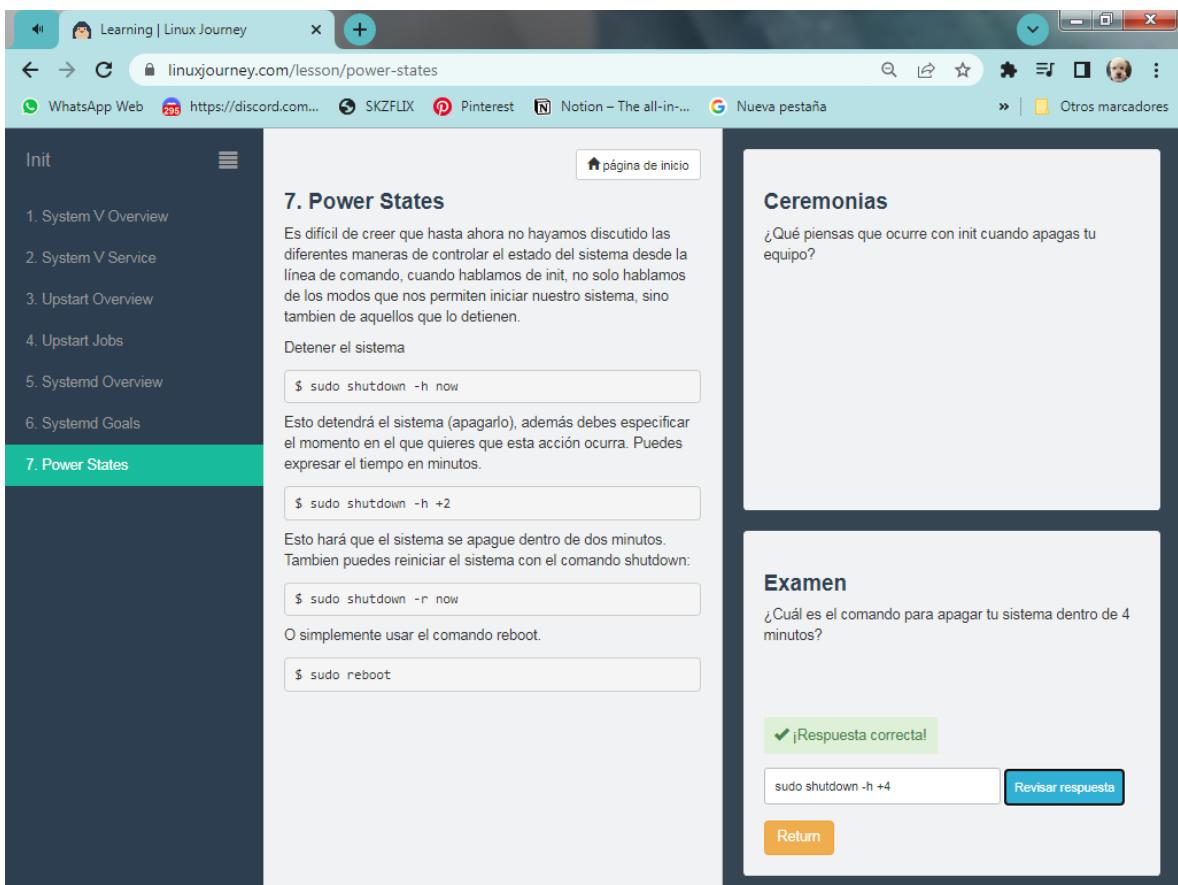
Examen

¿Cuál es el comando para apagar tu sistema dentro de 4 minutos?

✓ ¡Respuesta correcta!

sudo shutdown -h +4 Revisar respuesta

Return



Process Utilization

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/tracking-processes-top. The page title is "Process Utilization". On the left, a sidebar lists topics: 1. Tracking processes: top (highlighted in green), 2. lsof and fuser, 3. Process Threads, 4. CPU Monitoring, 5. I/O Monitoring, 6. Memory Monitoring, 7. Continuous Monitoring, and 8. Cron Jobs.

1. Tracking processes: top

In this course, we'll go over how to read and analyze the resource utilization on your system, this lesson shows some great tools to use when you need to track what a process is doing.

top

We've discussed top before, but we're going to dig into the specifics of what it's actually displaying. Remember top is the tool we used to get a real time view of the system utilization by our processes:

```
top - 18:06:26 up 6 days, 4:07, 2 users, load ave
Tasks: 389 total, 1 running, 387 sleeping, 0 sto
%Cpu(s): 1.8 us, 0.4 sy, 0.0 ni, 97.6 id, 0.1 wa
KiB Mem: 32870888 total, 27467976 used, 5402912 fr
KiB Swap: 33480700 total, 39892 used, 33440808 fr

PID USER      PR  NI    VIRT    RES    SHR S %CPU
6675 patty    20   0 1731472 520960 30876 S  8.3
6926 patty    20   0  935888 163456 25576 S  4.3
```

Let's go over what this output means, you don't have to memorize this, but come back to this when you need a reference.

1st line: This is the same information you would see if you ran the uptime command (more to come)

The fields are from left to right:

Ceremonias

Play around with the top command and see what processes are using the most resources.

Examen

What command displays the same output as the first line in top?

Respuesta correcta!

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/tracking-processes-lsof-fuser. The page title is "Process Utilization". On the left, a sidebar lists topics: 1. Tracking processes: top, 2. lsof and fuser (highlighted in green), 3. Process Threads, 4. CPU Monitoring, 5. I/O Monitoring, 6. Memory Monitoring, 7. Continuous Monitoring, and 8. Cron Jobs.

2. lsof and fuser

Let's say you plugged in a USB drive and starting working on some files, once you were done, you go and umount the USB device and you're getting an error "Device or Resource Busy". How would you find out which files in the USB drive are still in use? There are actually two tools you can use for this:

lsof

Remember files aren't just text files, images, etc, they are everything on the system, disks, pipes, network sockets, devices, etc. To see what is in use by a process, you can use the lsof command (short for "list open files") this will show you a list of all the open files and their associated process.

```
pete@icebox:~$ lsof .
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NOD
1xsession 1491 pete cwd DIR 8,6 4096 131
update-no 1796 pete cwd DIR 8,6 4096 131
nm-applet 1804 pete cwd DIR 8,6 4096 131
indicator 1809 pete cwd DIR 8,6 4096 131
xterm 2205 pete cwd DIR 8,6 4096 131
bash 2207 pete cwd DIR 8,6 4096 131
lsof 5914 pete cwd DIR 8,6 4096 131
lsof 5915 pete cwd DIR 8,6 4096 131
```

Now I can see what processes are currently holding the device/file open. In our USB example, you can also kill these processes so we can umount this pesky drive.

Ceremonias

Read the manpages for lsof and fuser, there is a lot of information that we didn't cover that allows you to have greater flexibility with these tools.

Examen

What command is used to list open files and their process information?

Respuesta correcta!

Learning | Linux Journey x +

linuxjourney.com/lesson/process-threads

WhatsApp Web https://discord.com... SKZFLIX Pinterest Notion – The all-in... Nueva pestaña » Otros marcadores

Process Utilization

1. Tracking processes: top
2. lsof and fuser
- 3. Process Threads**
4. CPU Monitoring
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs

3. Process Threads

You may have heard of the terms single-threaded and multi-threaded processes. Threads are very similar to processes, in that they are used to execute the same program, they are often referred to as lightweight processes. If a process has one thread it is single-threaded and if a process has more than one thread it is multi-threaded. However, all processes have at least one thread.

Processes operate with their own isolated system resources, however threads can share these resources among each other easily, making it easier for them to communicate among each other and at times it is more efficient to have a multi-threaded application than a multi-process application.

Basically, let's say you open up LibreOffice Writer and Chrome, each is its own separate process. Now you go inside Writer and start editing text, when you edit the text it gets automatically saved. These two parallel "lightweight processes" of saving and editing are threads.

To view process threads, you can use:

```
pete@icebox:~$ ps m
 PID TTY      STAT   TIME  COMMAND
 2207 pts/2    - 0:01 bash
      -  Ss    0:01 -
 5252 pts/2    - 0:00 ps m
      -  R+    0:00 -
```

The processes are denoted with each PID and underneath the processes are their threads (denoted by a -). So you can see that the processes above are both single-threaded.

Ceremonias

Run the `ps m` command and see what processes you have running are multi-threaded.

Examen

True or false, all processes start out single-threaded.

¡Respuesta correcta!

true

Revisar respuesta

Continuar

Learning | Linux Journey x +

linuxjourney.com/lesson/cpu-monitoring

WhatsApp Web https://discord.com... SKZFLIX Pinterest Notion – The all-in... Nueva pestaña » Otros marcadores

Process Utilization

1. Tracking processes: top
2. lsof and fuser
3. Process Threads
- 4. CPU Monitoring**
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs

4. CPU Monitoring

Let's go over a useful command, `uptime`.

```
pete@icebox:~$ uptime
17:23:35 up 1 day, 5:59, 2 users, load average:
```

We talked about uptime in the first lesson of this course, but we haven't gone over the load average field. Load averages are good way to see the CPU load on your system. These numbers represent the average CPU load in 1, 5, and 15 minute intervals. What do I mean by CPU load, the CPU load is the average number of processes that are waiting to be executed by the CPU.

Let's say you have a single-core CPU, think of this core as a single lane in traffic. If it's rush hour on the freeway, this lane is gonna be really busy and traffic is gonna be at 100% or a load of 1. Now the traffic has become so bad, it's backing up the freeway and getting the regular roads busy by twice the amount of cars, we can say that your load is 200% or a load of 2. Now let's say it clears up a bit and there are only half as many cars on the freeway lane, we can say the load of the lane is 0.5. When traffic is non-existent and we can get home quicker, the load should ideally be very low, like 2am traffic low. The cars in this case are processes and these processes are just waiting to get off the freeway and get home.

Now just because you have a load average of 1 doesn't mean your computer is slogging around. Most modern machines these days have multiple cores. If you had a quad core processor (4 cores) and your load average is 1, it's really just affecting 25% of your CPU. Think of each core as a lane in

Ceremonias

Check the load average of your system and see what it's doing.

Examen

What command can you use to see the load average?

¡Respuesta correcta!

uptime

Revisar respuesta

Continuar

Learning | Linux Journey

linuxjourney.com/lesson/io-monitoring

WhatsApp Web SKZFLIX Pinterest Notion – The all-in... Nueva pestaña Otros marcadores

Process Utilization

1. Tracking processes: top
2. Isof and fuser
3. Process Threads
4. CPU Monitoring
- 5. I/O Monitoring**
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs

5. I/O Monitoring

We can also monitor CPU usage as well as monitor disk usage with a handy tool known as `iostat`

```
pete@icebox:~$ iostat
Linux 3.13.0-39-lowlatency (icebox)      01/28/2016

avg-cpu: %user   %nice %system %iowait %steal   %i
          0.13    0.03   0.50    0.01   0.00   99

Device:     tps   kB_read/s   kB_wrtn/s
sda        0.17       3.49       1.92
```

The first part is the CPU information:

- %user - Show the percentage of CPU utilization that occurred while executing at the user level (application)
- %nice - Show the percentage of CPU utilization that occurred while executing at the user level with nice priority.user CPU utilization with nice priorities
- %system - Show the percentage of CPU utilization that occurred while executing at the system level (kernel).
- %iowait - Show the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.
- %steal - Show the percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor.
- %idle - Show the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.

Ceremonias

Use `iostat` to view your disk usage.

Examen

What command can be used to view I/O and CPU usage?

✓ ¡Respuesta correcta!

iostat Revisar respuesta Continuar

Learning | Linux Journey

linuxjourney.com/lesson/memory-monitoring

WhatsApp Web SKZFLIX Pinterest Notion – The all-in... Nueva pestaña Otros marcadores

Process Utilization

1. Tracking processes: top
2. Isof and fuser
3. Process Threads
4. CPU Monitoring
5. I/O Monitoring
- 6. Memory Monitoring**
7. Continuous Monitoring
8. Cron Jobs

6. Memory Monitoring

The fields are as follows:

procs	memory	swap	10-
r b swpd	free buff cache	si so bi	
1 0 0	396528 38816 384036	0 0 4	

procs

- r - Number of processes for run time
- b - Number of processes in uninterruptible sleep

memory

- swpd - Amount of virtual memory used
- free - Amount of free memory
- buff - Amount of memory used as buffers
- cache - Amount of memory used as cache

swap

- si - Amount of memory swapped in from disk
- so - Amount of memory swapped out to disk

io

- bi - Amount of blocks received in from a block device
- bo - Amount of blocks sent out to a block device

system

- in - Number of interrupts per second
- cs - Number of context switches per second

cpu

- us - Time spent in user time
- sy - Time spent in kernel time
- id - Time spent idle
- wa - Time spent waiting for IO

Ceremonias

Look at your memory usage with `vmstat`.

Examen

What tool is used to view memory utilization?

✓ ¡Respuesta correcta!

vmstat Revisar respuesta Continuar

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/continuous-monitoring. The page title is "7. Continuous Monitoring". On the left sidebar, under "Process Utilization", the following items are listed: 1. Tracking processes: top, 2. Iosf and fuser, 3. Process Threads, 4. CPU Monitoring, 5. I/O Monitoring, 6. Memory Monitoring, 7. Continuous Monitoring (which is highlighted in green), and 8. Cron Jobs. The main content area starts with a section titled "7. Continuous Monitoring". It discusses the use of sar for historical analysis and provides several command examples:

- Installing sar**: Sar is a tool used for historical analysis. It needs to be installed via `sudo apt install sysstat`.
- Setting up data collection**: Once sysstat is installed, the system will automatically start collecting data.
- Using sar**:
 - `$ sudo sar -q`: This command lists details from the start of the day.
 - `$ sudo sar -r`: This command lists memory usage details from the start of the day.
 - `$ sudo sar -P`: This command lists CPU usage details.
- To view data from a specific day, you can use `/var/log/sysstat/saXX` where XX is the day of the month.

On the right side of the page, there are two boxes: "Ceremonias" and "Examen".

Ceremonias: A box containing the text: "Install sar on your system and start collecting and analyzing your system resource utilization."

Examen: A box asking: "What is a good tool to use for monitoring system resources?". It contains a text input field with "sar" and a button "Revisar respuesta".

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/cron-jobs. The page title is "8. Cron Jobs". On the left sidebar, under "Process Utilization", the following items are listed: 1. Tracking processes: top, 2. Iosf and fuser, 3. Process Threads, 4. CPU Monitoring, 5. I/O Monitoring, 6. Memory Monitoring, 7. Continuous Monitoring, and 8. Cron Jobs (which is highlighted in green). The main content area starts with a section titled "8. Cron Jobs". It discusses the cron service and provides an example of a cron job to change a wallpaper:

```
30 08 * * * /home/pete/scripts/change_wallpaper
```

The fields are as follows from left to right:

- Minute - (0-59)
- Hour - (0-23)
- Day of the month - (1-31)
- Month - (1-12)
- Day of the week - (0-7). 0 and 7 are denoted as Sunday

The asterisk in the field means to match every value. So in my above example, I want this to run every day in every month at 8:30am.

To create a cronjob, just edit the crontab file:

```
crontab -e
```

On the right side of the page, there are two boxes: "Ceremonias" and "Examen".

Ceremonias: A box containing the text: "Create a cronjob that you want to run at a scheduled time."

Examen: A box asking: "What is the command to edit your cronjobs?". It contains a text input field with "crontab -e" and a button "Revisar respuesta".

Logging

The services, kernel, daemons, etc on your system are constantly doing something, this data is actually sent to be saved on your system in the form of logs. This allows us to have a human readable journal of the events that are happening on our system. This data is usually kept in the /var directory, the /var directory is where we keep our variable data, such as logs!

How are these messages even getting received on your system? There is a service called syslog that sends this information to the system logger.

Syslog actually contains many components, one of the important ones is a daemon running called syslogd (newer Linux distributions use rsyslogd), that waits for event messages to occur and filter the ones it wants to know about, and depending on what it's supposed to do with that message, it will send it to a file, your console or do nothing with it.

You would think that this system logger is the centralized place to manage logs, but unfortunately it's not. You'll see many applications that write their own logging rules and generate different log files, however in general the format of logs should include a timestamp and the event details.

Here is an example of a line from syslog:

```
pete@icebox:~$ less /var/log/syslog
Jan 27 07:41:32 icebox anacron[4650]: Job `cron.week'
```

Here we can see that at Jan 27 07:41:32 our cron service ran the cron.weekly job. You can view all the event messages that syslog collects with in the /var/log/syslog file.

Ceremonias
Look at your /var/log/syslog file and see what else is happening on your machine.

Examen
What is the daemon that manages log on newer Linux systems?

✓ ¡Respuesta correcta!
rsyslogd

The syslog service manages and sends logs to the system logger. Rsyslog is an advanced version of syslog, most Linux distributions should be using this new version. The output of all the logs the syslog service collects can be found at /var/log/syslog (every message except auth messages).

To find out what files are maintained by our system logger, look at the configuration files in /etc/rsyslog.d:

```
pete@icebox:~$ less /etc/rsyslog.d/50-default.conf
# First some standard log files. Log by facility.
#
auth,authpriv.*          /var/log/auth.log
*.*;auth,authpriv.none    -/var/log/syslog
#cron.*                   /var/log/cron.log
#daemon.*                 -/var/log/daemon.log
kern.*                    -/var/log/kern.log
#lpr.*                     -/var/log/lpr.log
mail.*                    -/var/log/mail.log
#user.*                   -/var/log/user.log
```

These rules to log files are denoted by the selector on the left column and the action on the right column. The action tells us where to send the log information, in a file, console, etc. Remember not every application and service uses rsyslog to manage their logs, so if you want to know specifically what is logged you'll have to look inside this directory.

Let's actually see logging in action, you can manually send a log with the logger command:

Ceremonias
Look at your /etc/rsyslog.d configuration file and see what else is being logged via the system logger.

Examen
What command can you use to manually log a message?

✓ ¡Respuesta correcta!
logger

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/general-logging. The page is titled "Logging" and has a sidebar with the following menu:

- 1. System Logging
- 2. syslog
- 3. General Logging**
- 4. Kernel Logging
- 5. Authentication Logging
- 6. Managing Log Files

The main content area is titled "3. General Logging". It contains two sections: **/var/log/messages** and **/var/log/syslog**.

/var/log/messages

There are many log files you can view on your system, many important ones can be found under /var/log. We won't go through them all, but we'll discuss a couple of the major ones.

There are two general log files you can view to get a glimpse of what your system is doing:

/var/log/syslog

This log contains all non-critical and non-debug messages, includes messages logged during bootup (dmesg), auth, cron, daemon, etc. Very useful to get a glimpse of how your machine is acting.

These two logs should be more than enough when troubleshooting issues with your system. However, if you just want to view a specific log component, there are also separate logs for those as well.

Ceremonias

Look at your /var/log/messages and /var/log/syslog files and see what the differences are.

Examen

What log file logs everything except auth messages?

Respuesta correcta!

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/kernel-logging. The page is titled "Logging" and has a sidebar with the following menu:

- 1. System Logging
- 2. syslog
- 3. General Logging
- 4. Kernel Logging**
- 5. Authentication Logging
- 6. Managing Log Files

The main content area is titled "4. Kernel Logging". It contains two sections: **/var/log/dmesg** and **/var/log/kern.log**.

/var/log/dmesg

On boot-time your system logs information about the kernel ring buffer. This shows us information about hardware drivers, kernel information and status during bootup and more. This log file can be found at /var/log/dmesg and gets reset on every boot, you may not actually see any use in it now, but if you were to ever have issues with something during bootup or a hardware issue, dmesg is the best place to look. You can also view this log using the dmesg command.

/var/log/kern.log

Another log you can use to view kernel information is the /var/log/kern.log file, this logs the kernel information and events on your system, it also logs dmesg output.

Ceremonias

Look at your dmesg and kern logs, what differences do you notice?

Examen

What command can be used to view kernel bootup messages?

Respuesta correcta!

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/authentication-logging. The page title is "5. Authentication Logging". On the left sidebar, under the "Logging" category, "5. Authentication Logging" is highlighted in green. The main content area contains a section titled "5. Authentication Logging" with the following text:

Authentication logging can be very useful to look at if you are having issues logging in.

/var/log/auth.log

This contains system authorization logs, such as user login and the authentication method used.

Sample snippet:

```
Jan 31 10:37:50 icebox pkexec: pam_unix(polkit-1:session)
```

A sidebar on the right contains a "Ceremonias" section with the text: "Do some failed logins and then a successful one, look at your /var/log/auth.log and see what happened." Below it is an "Examen" section asking "What log is used for user authentication?" with a correct answer of "auth.log".

The screenshot shows a web browser window with the URL linuxjourney.com/lesson/managing-log-files. The page title is "6. Managing Log Files". On the left sidebar, under the "Logging" category, "6. Managing Log Files" is highlighted in green. The main content area contains a section titled "6. Managing Log Files" with the following text:

Log files generate lots of data and they store this data on your hard disks, however there are lots of issues with this, for the most part we just want to be able to see newer logs, we also want to manage our disk space efficiently, so how do we do all of this? The answer is with logrotate.

The logrotate utility does log management for us. It has a configuration file that allows us to specify how many and what logs to keep, how to compress our logs to save space and more. The logrotate tool is usually run out of cron once a day and the configuration files can be found in /etc/logrotate.d.

There are other logrotating tools you can use to manage your logs, but logrotate is the most common one.

A sidebar on the right contains a "Ceremonias" section with the text: "Look at your logrotate configuration file and see how it manages some of your logs." Below it is an "Examen" section asking "What utility is used to manage logs?" with a correct answer of "logrotate".