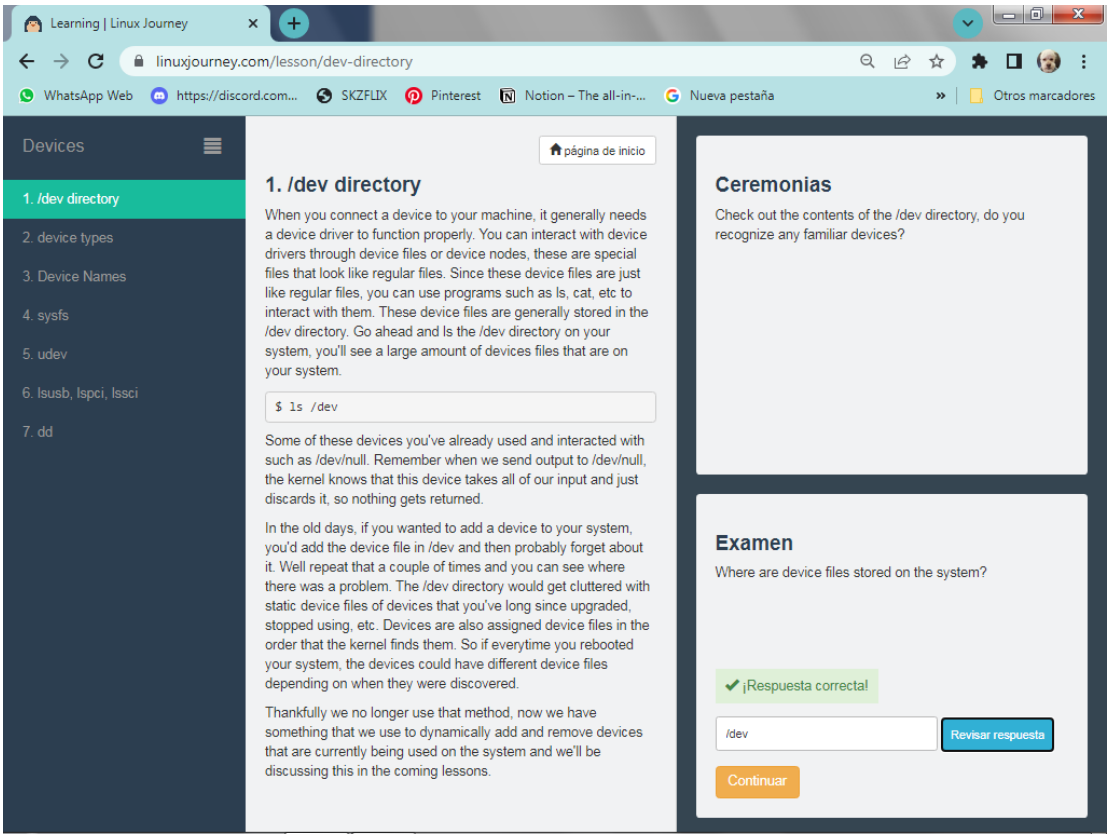
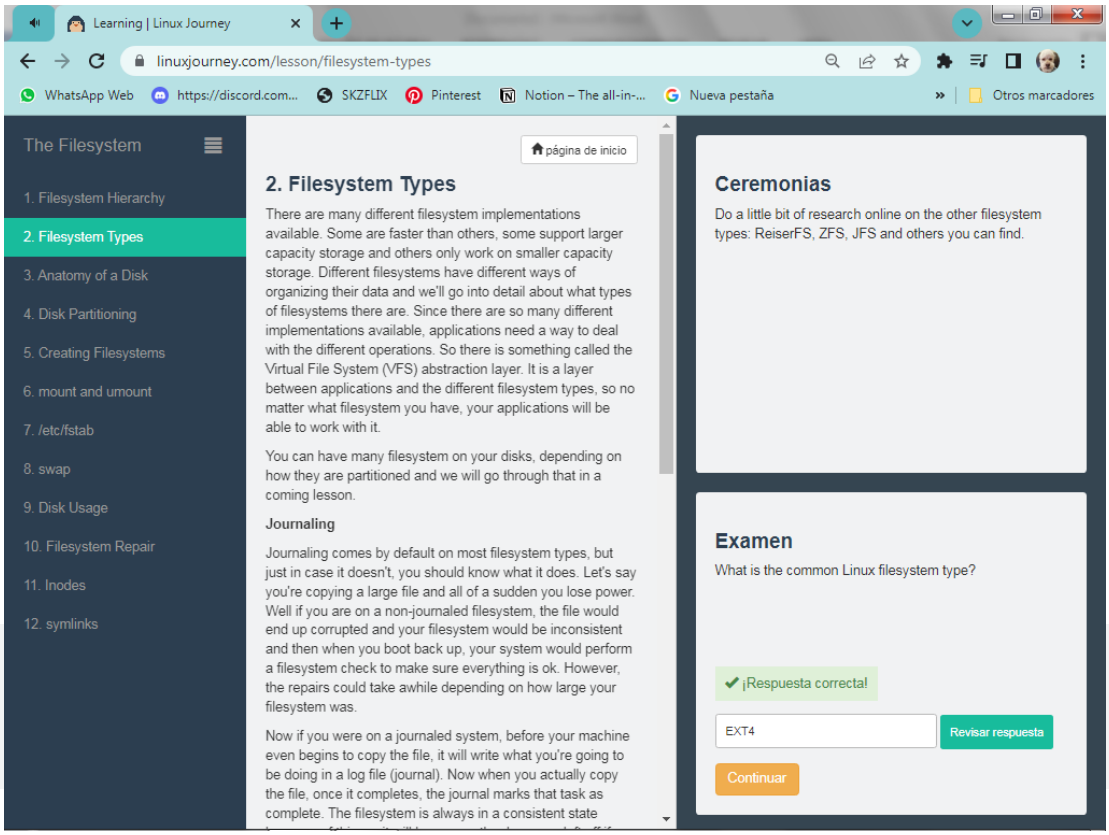
Journeyman

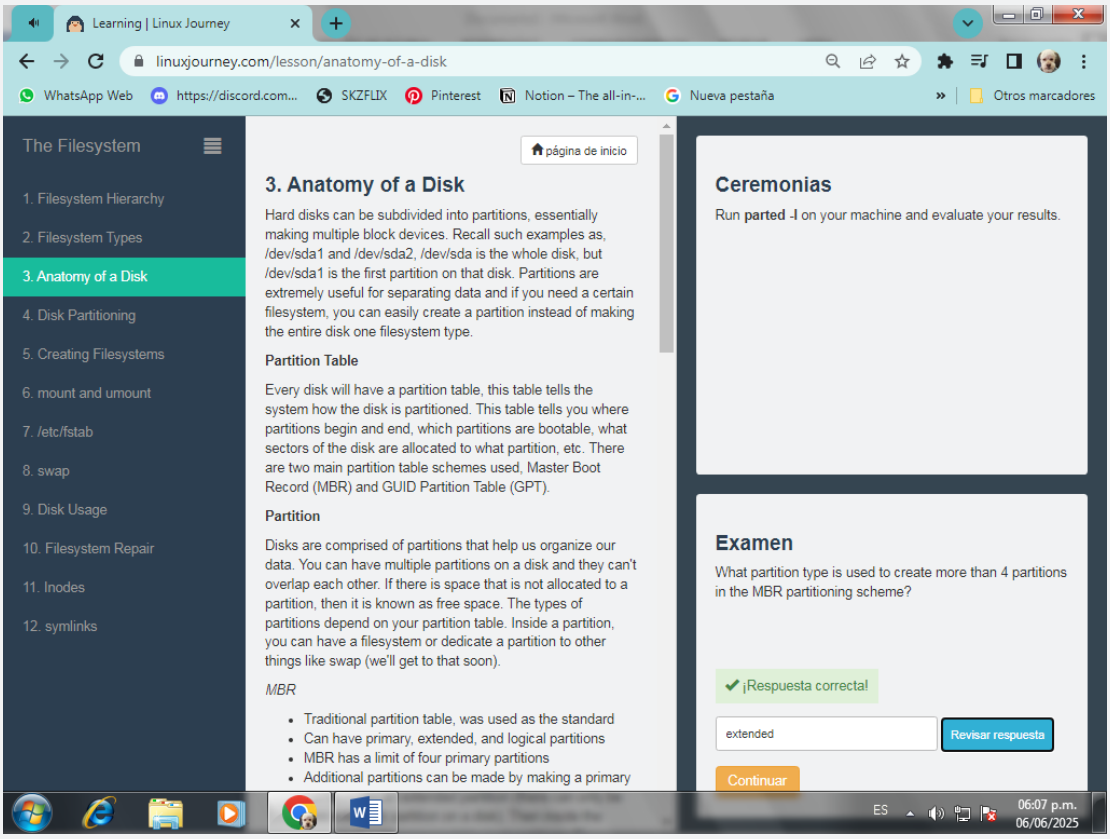
Devices:



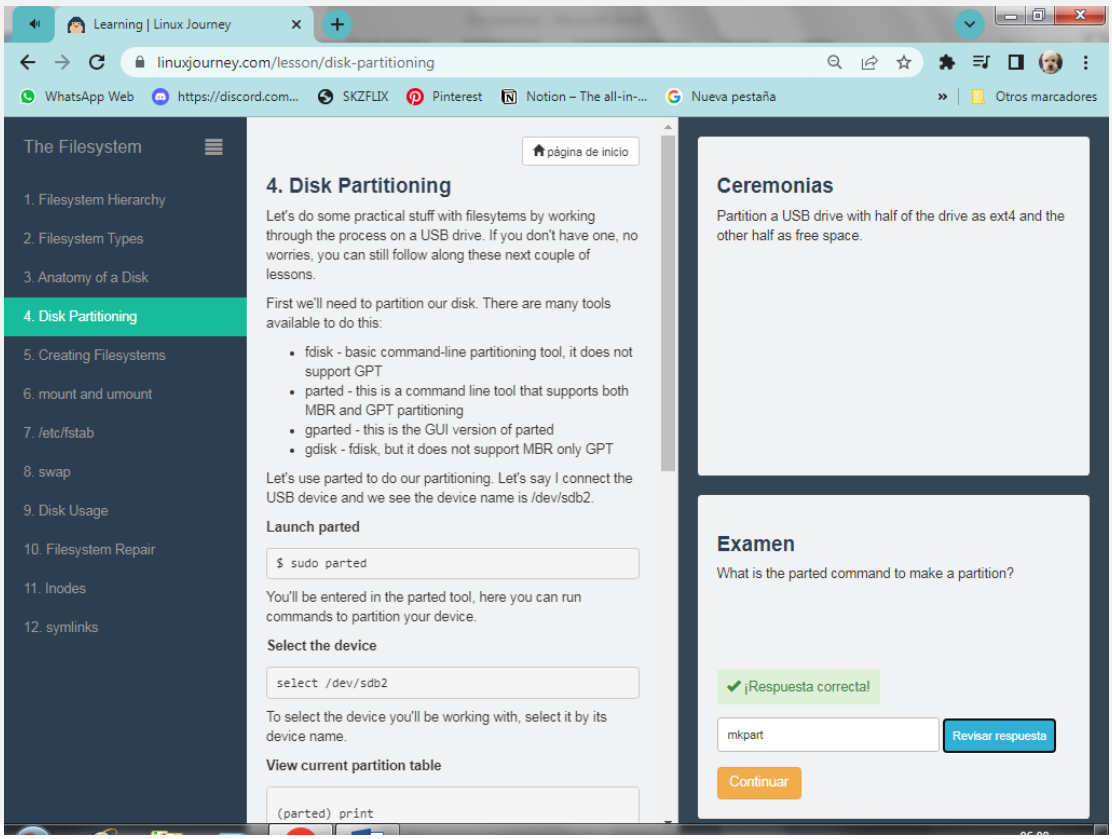
Filesystem Types:

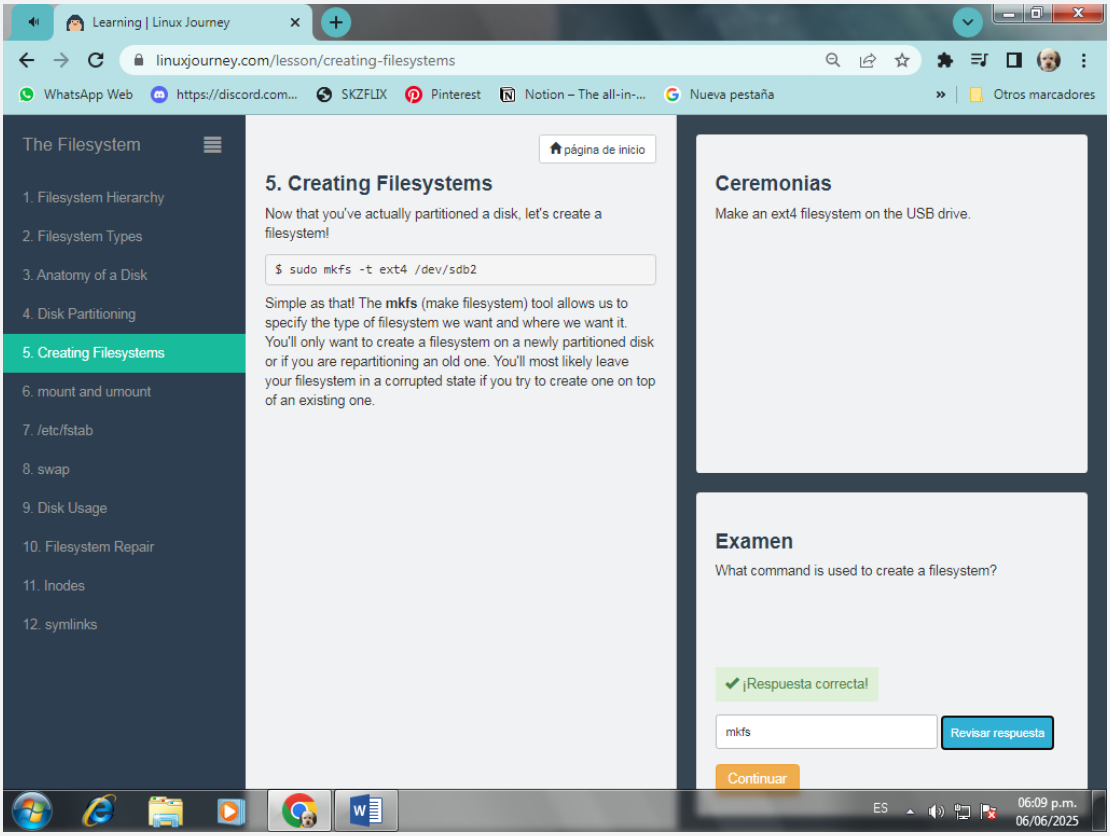


Anatomy of a Disk:

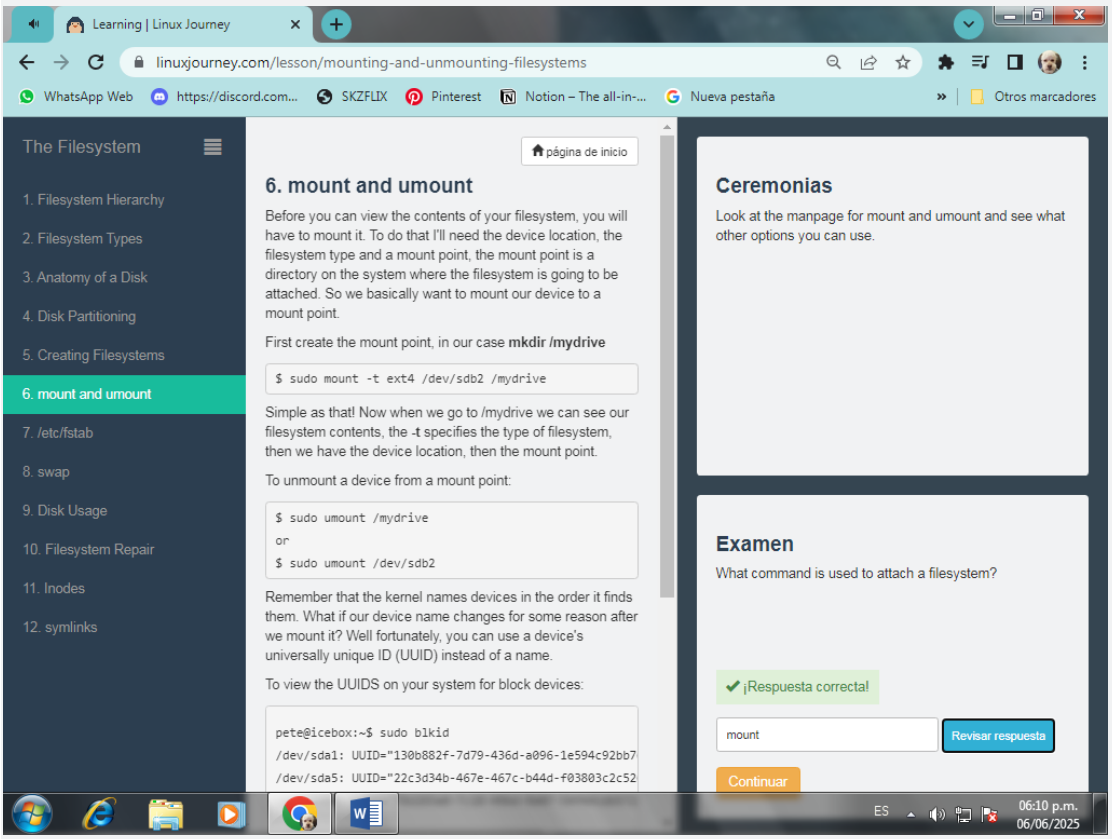


Disk Partitioning:

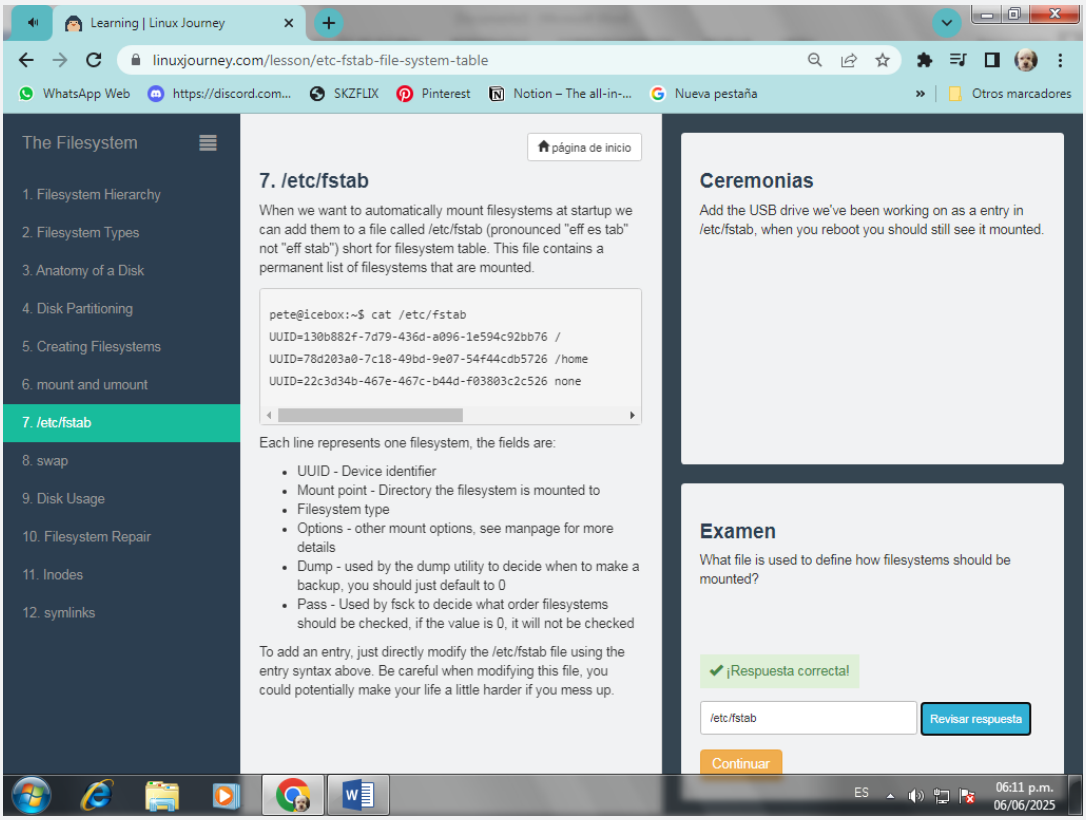




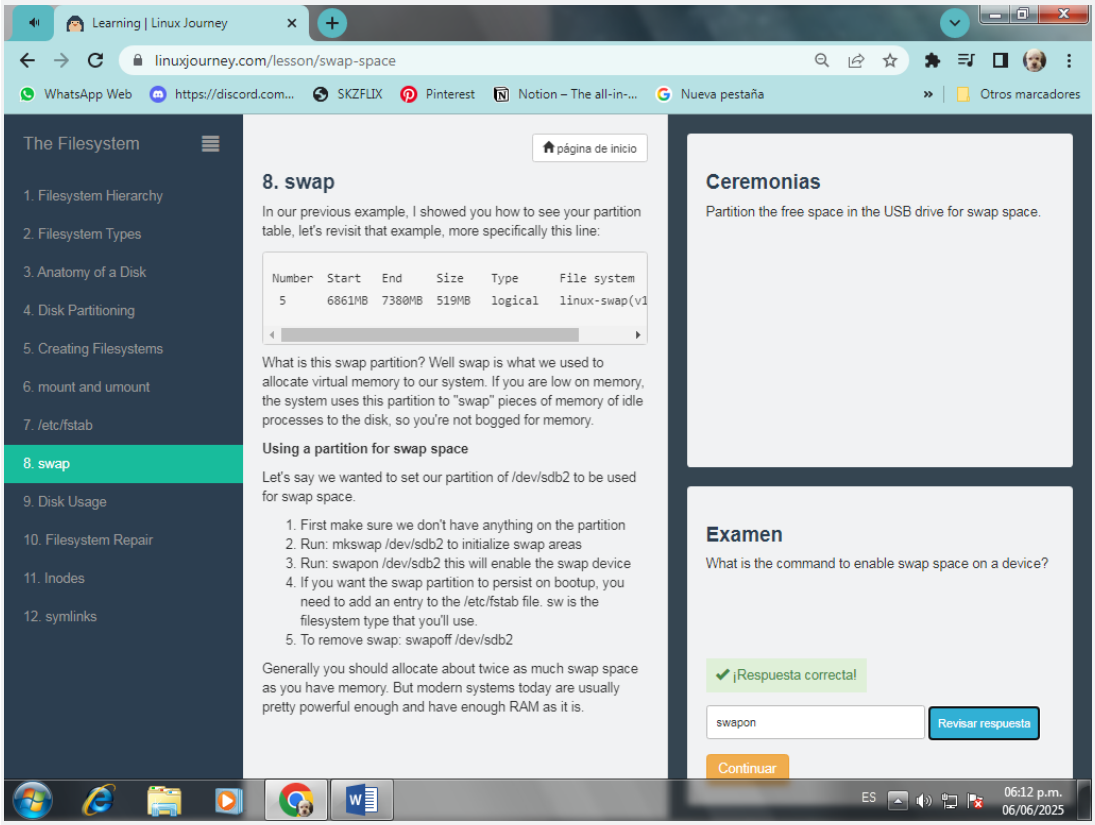
Mount and umount:



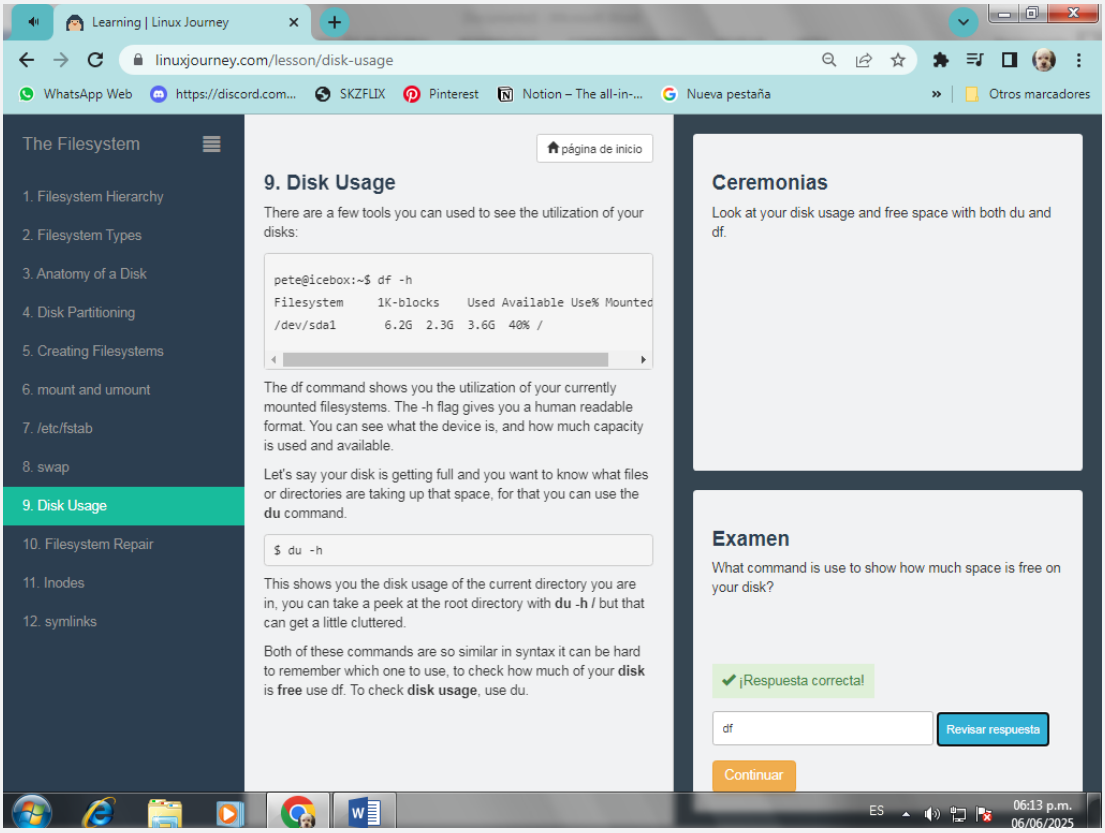
/etc/fstab:



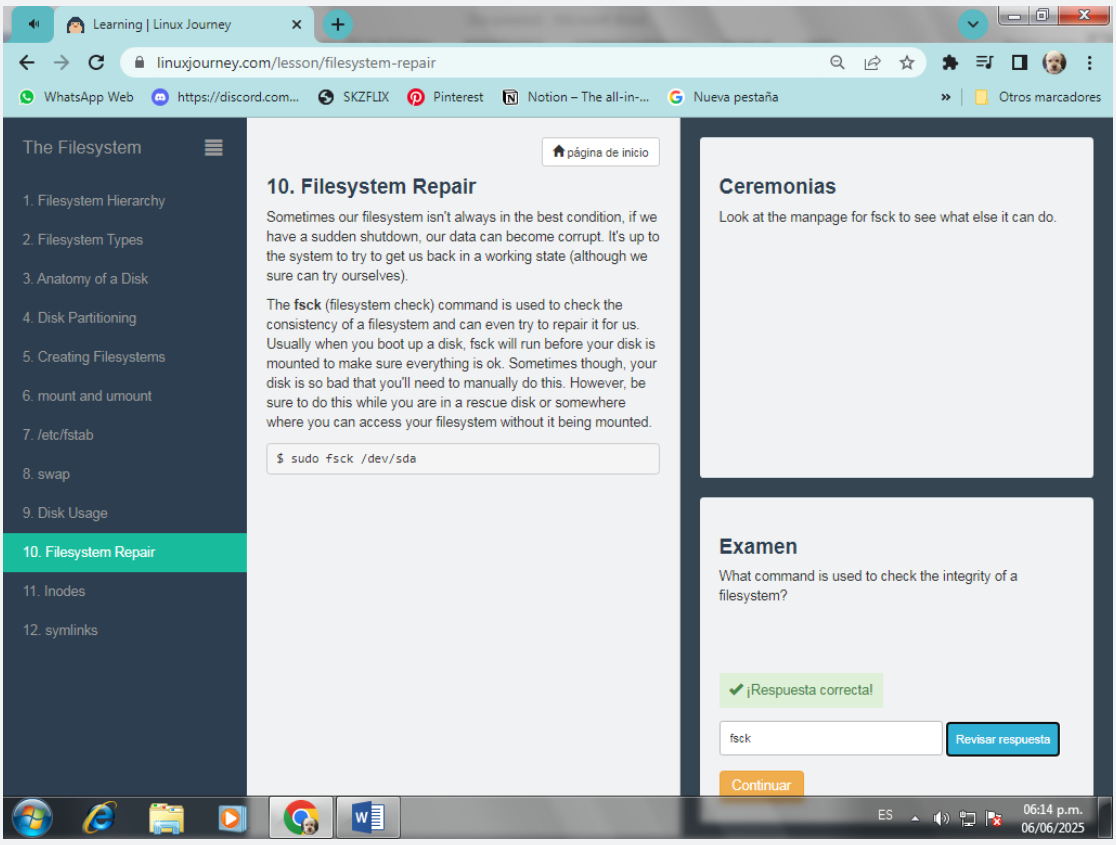
Swap:



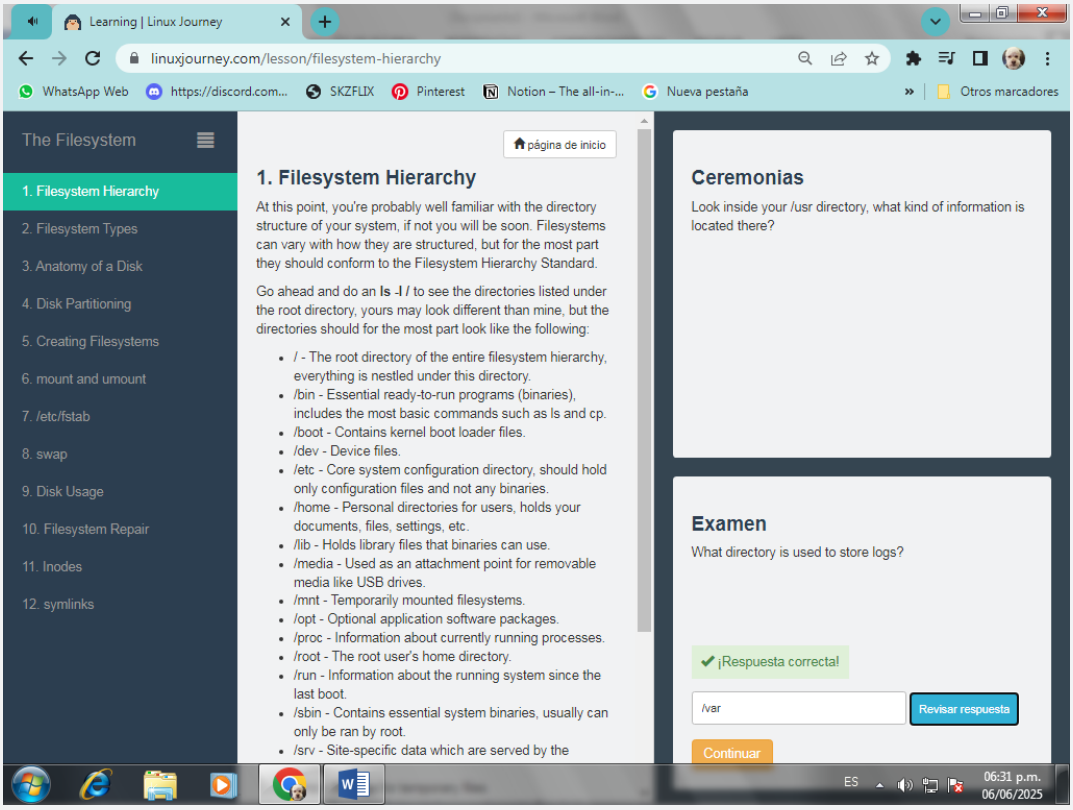
Disk Usage:



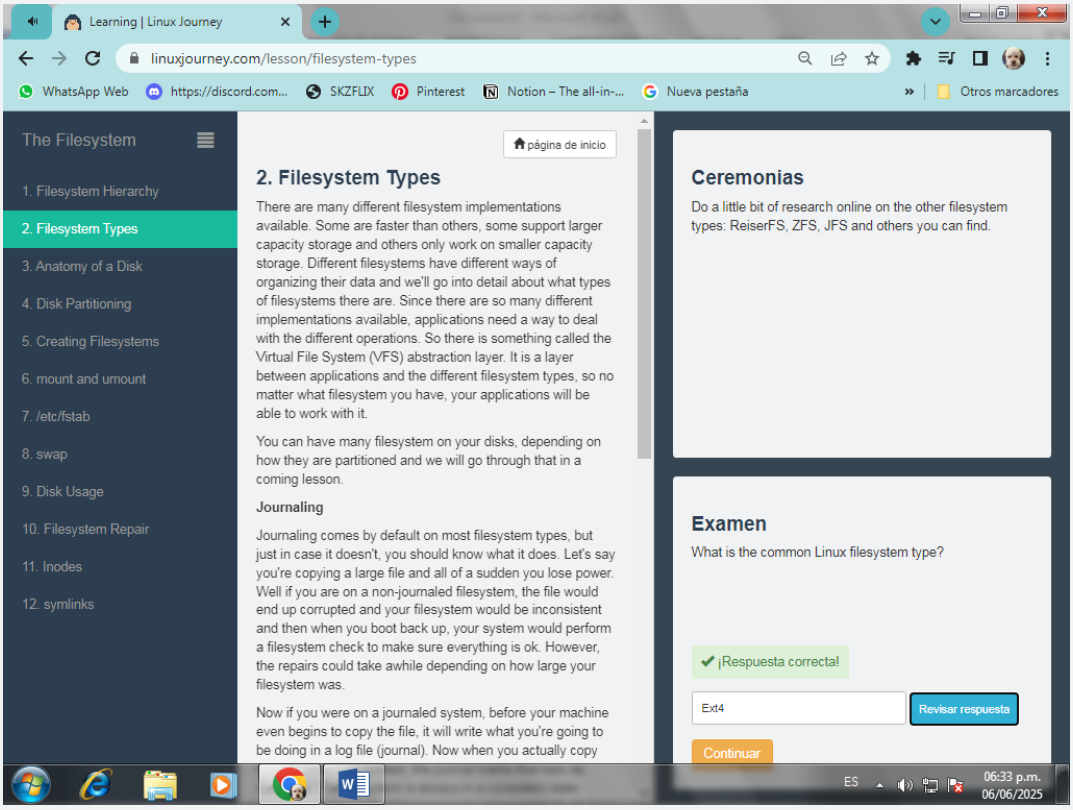
Filesystem Repair:



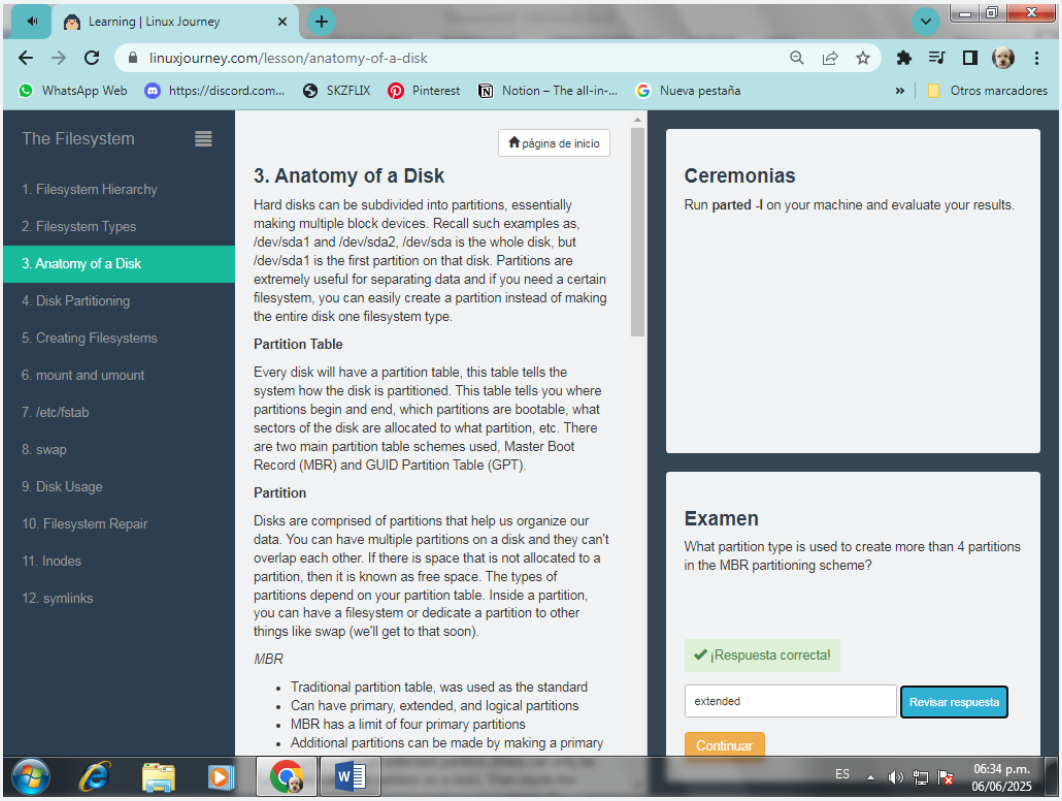
. Filesystem Hierarchy



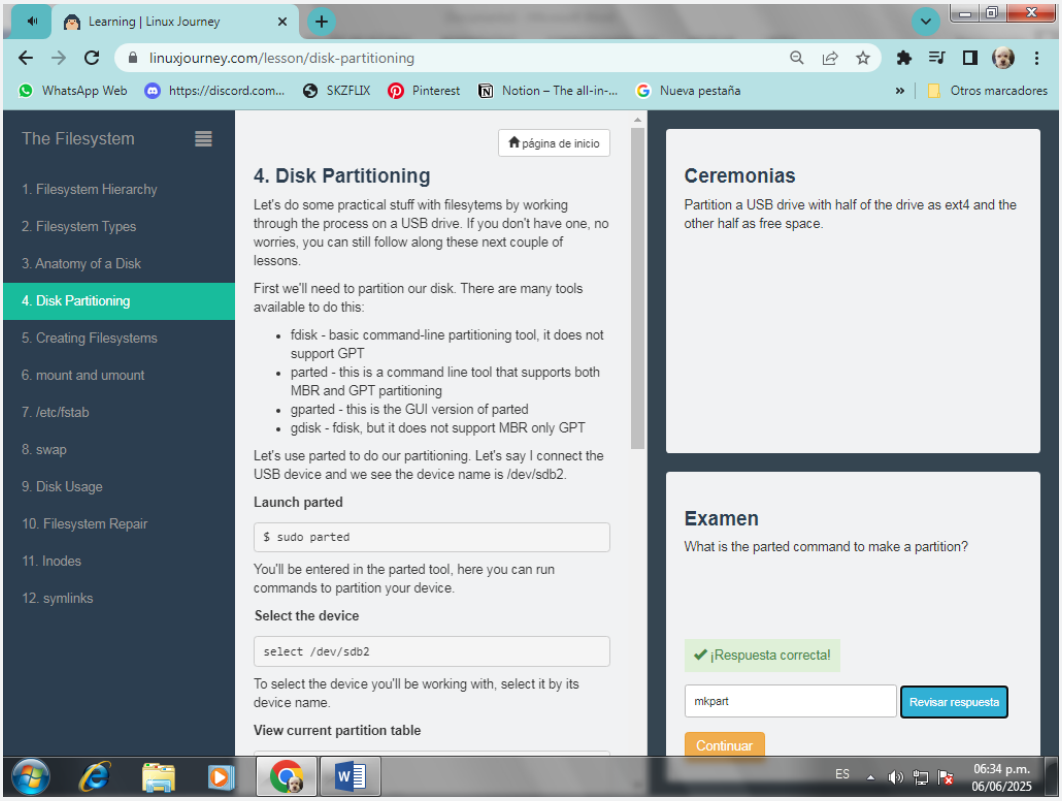
Filesystem Types:



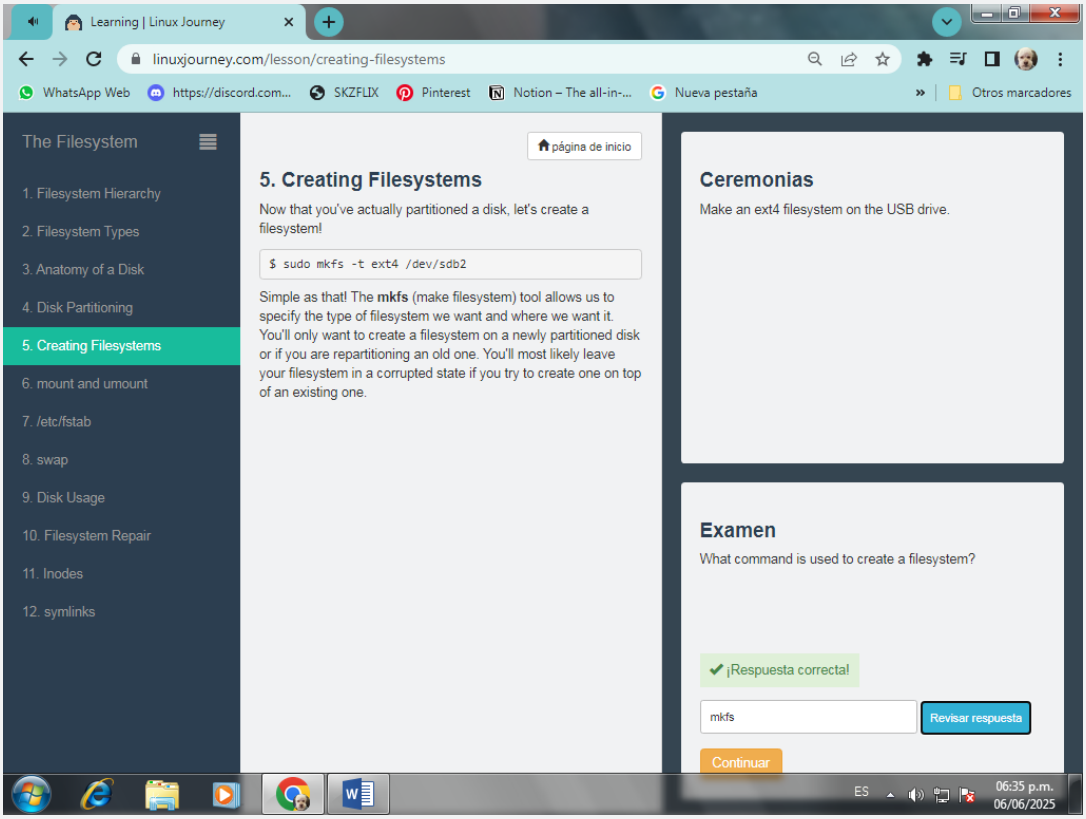
Anatomy of a Disk:



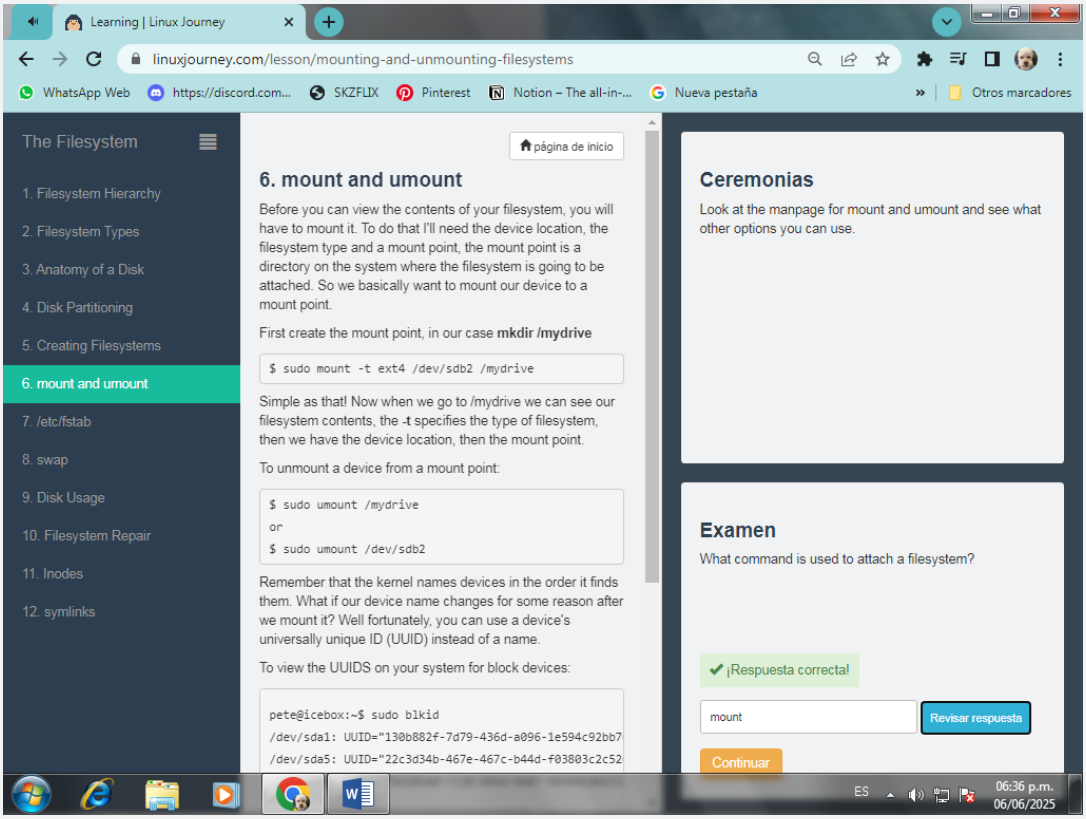
Disk Partitioning:



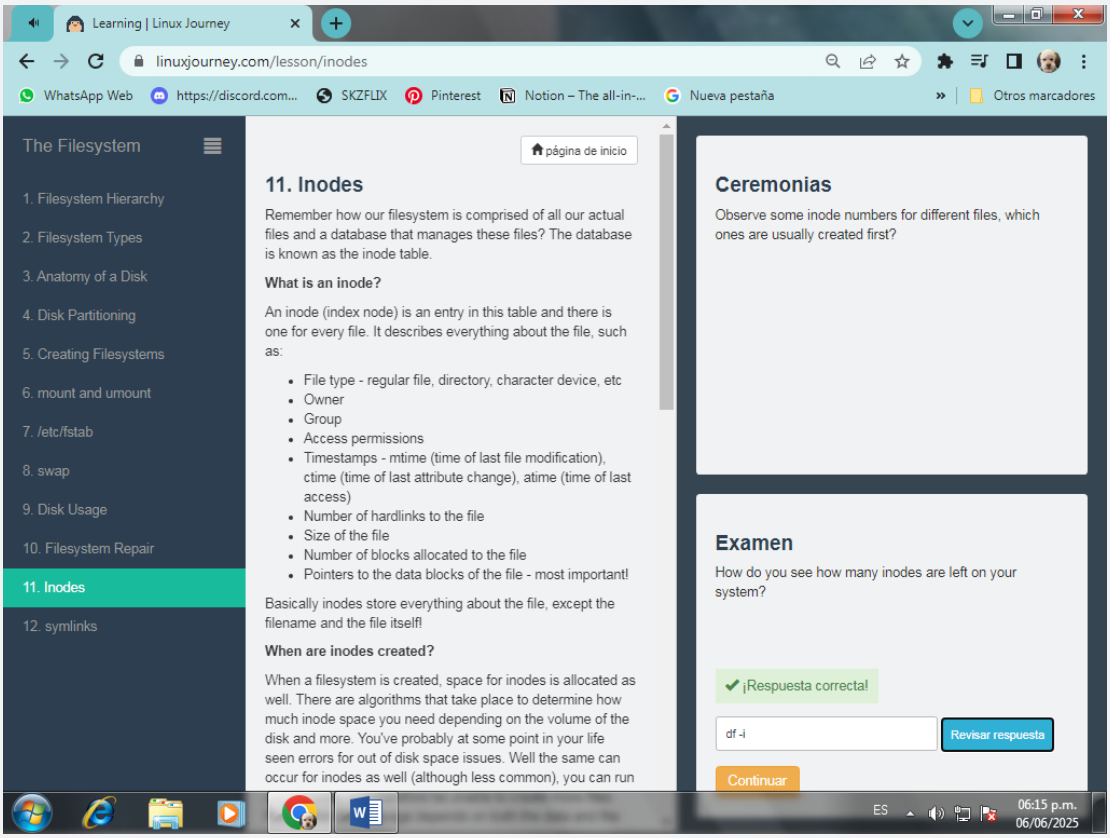
Creating Filesystems:



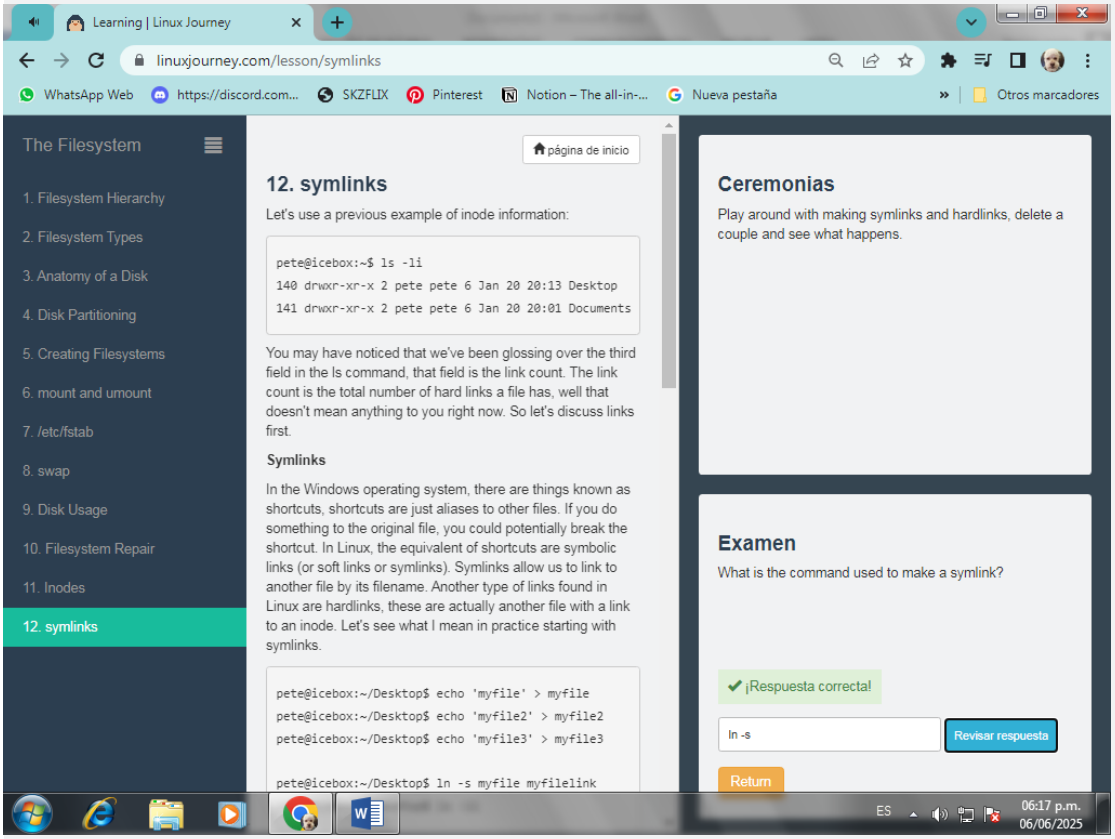
mount and umount:



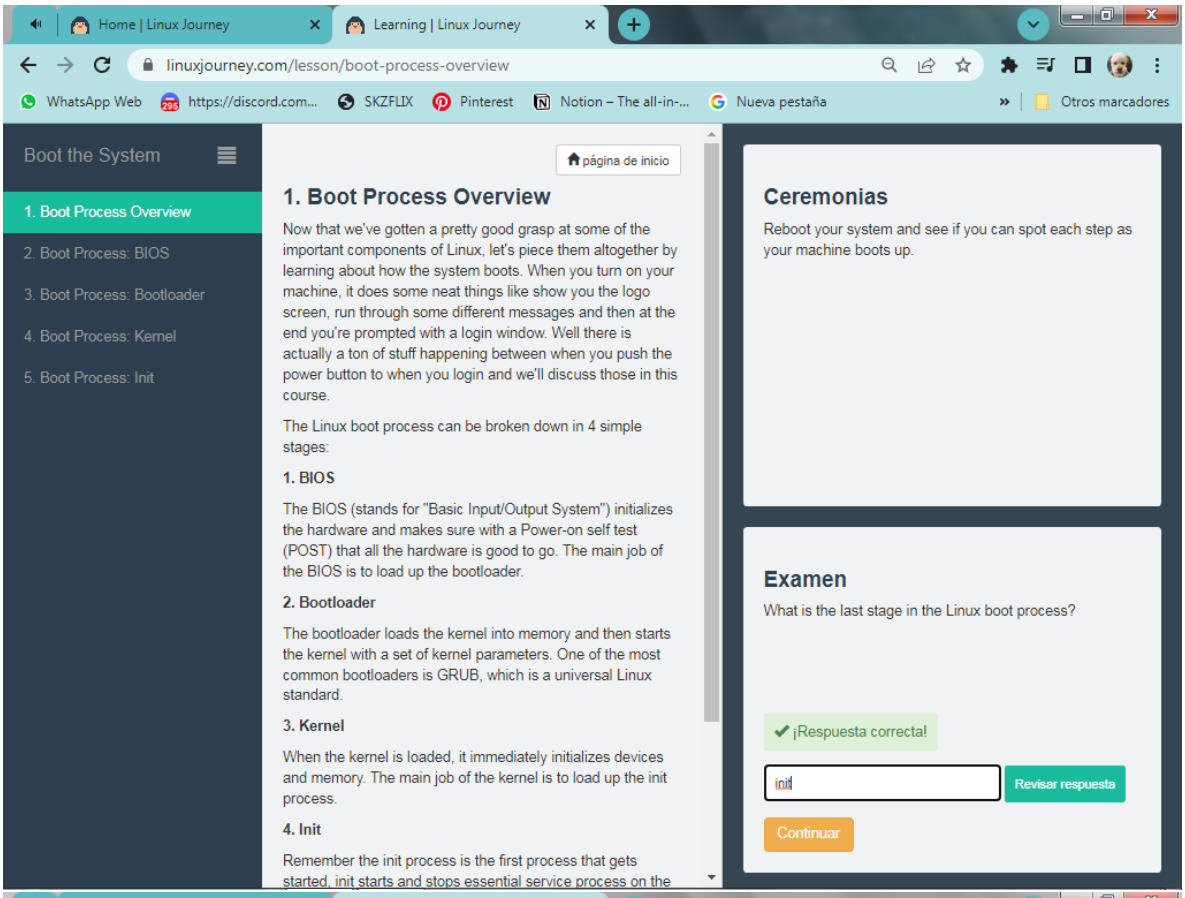
Inodes:

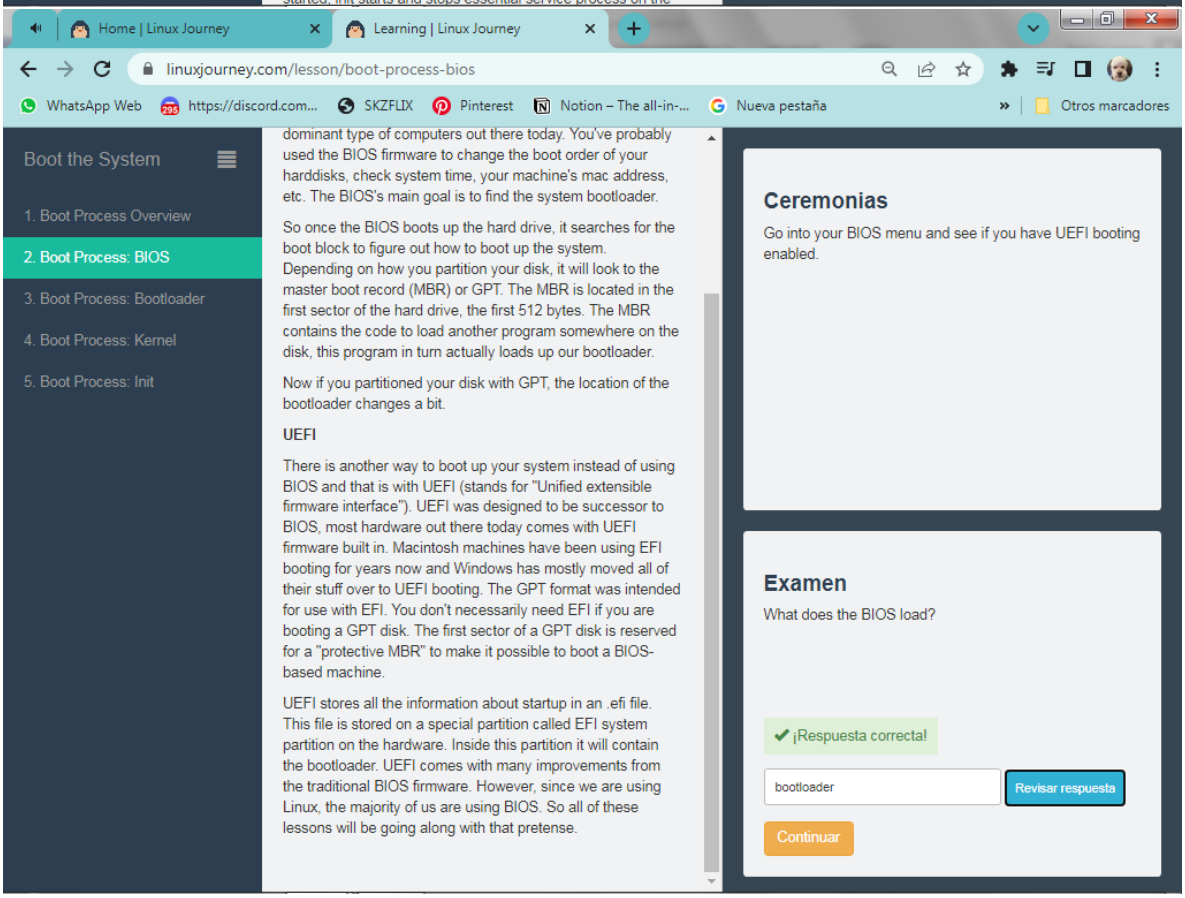


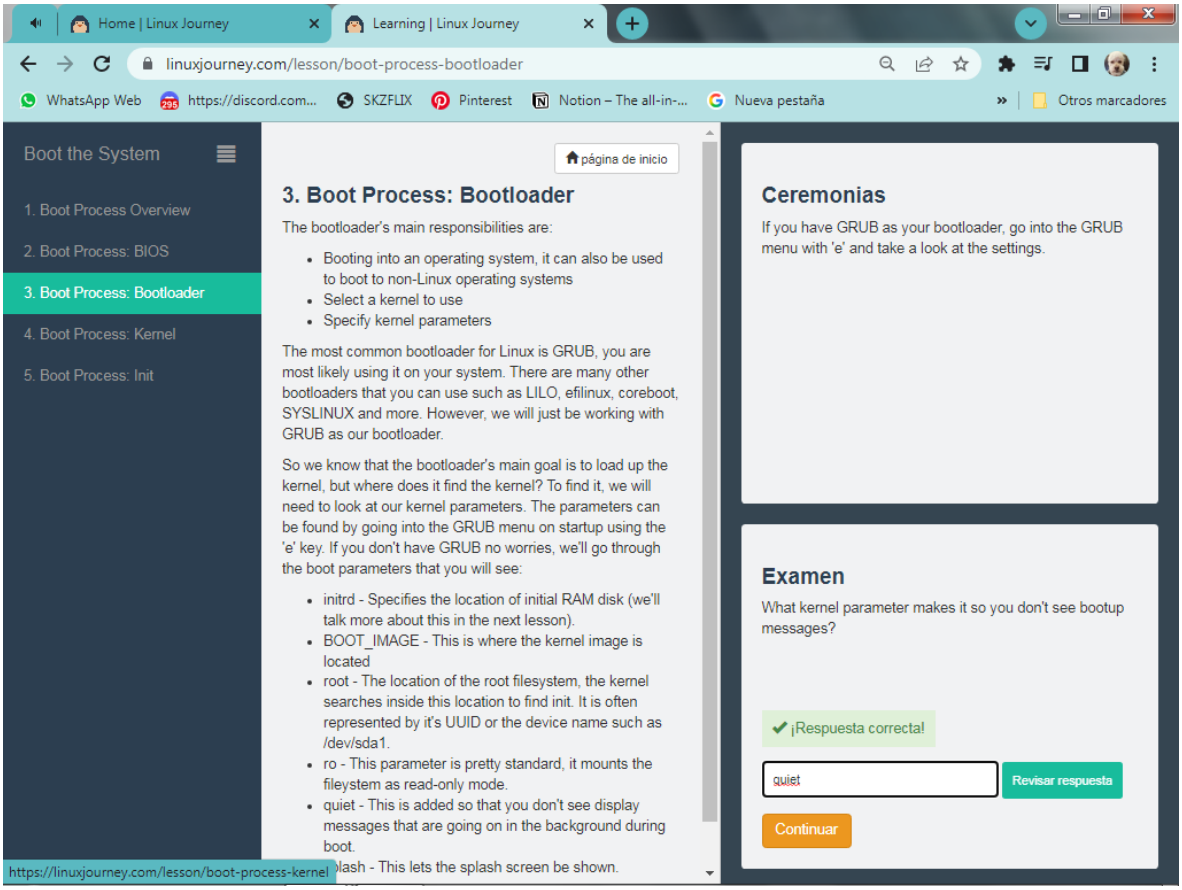
Symlinks:

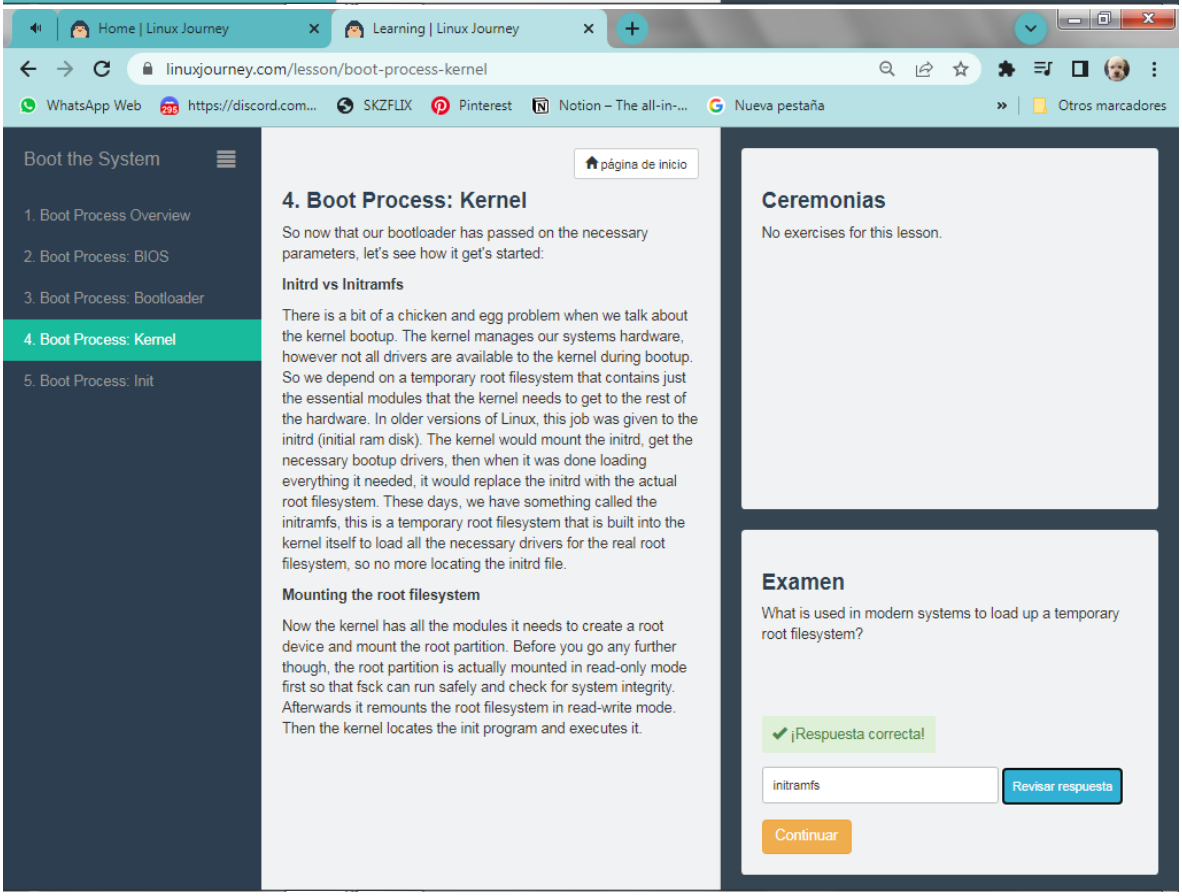


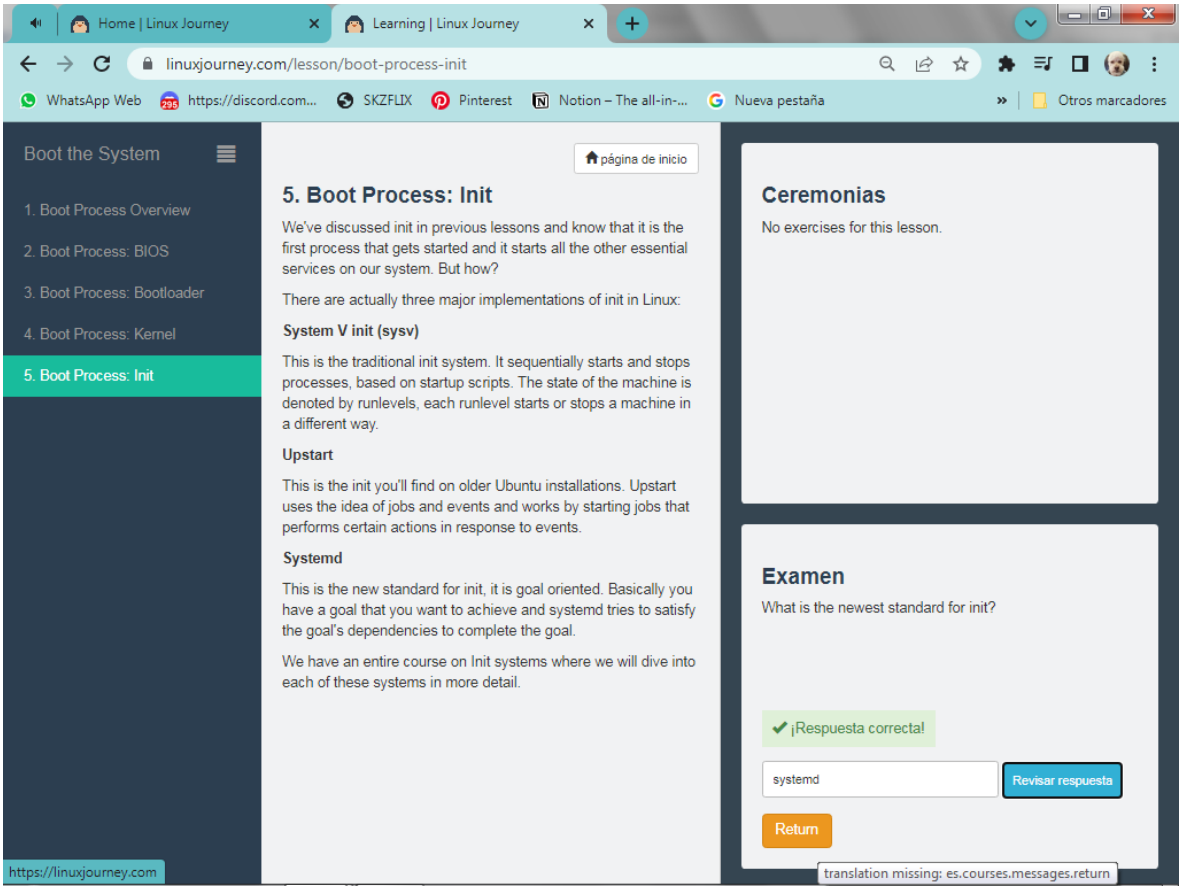
BOOT THE SYTEM



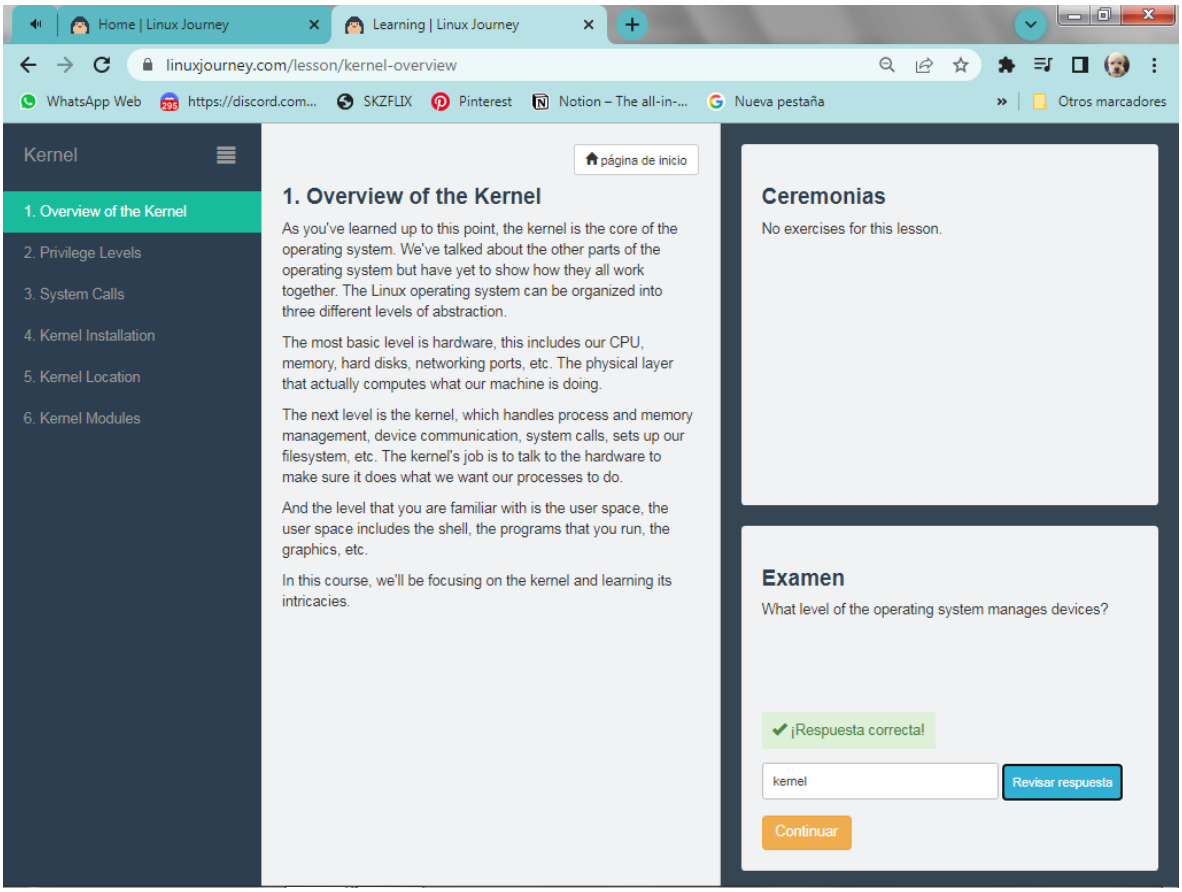


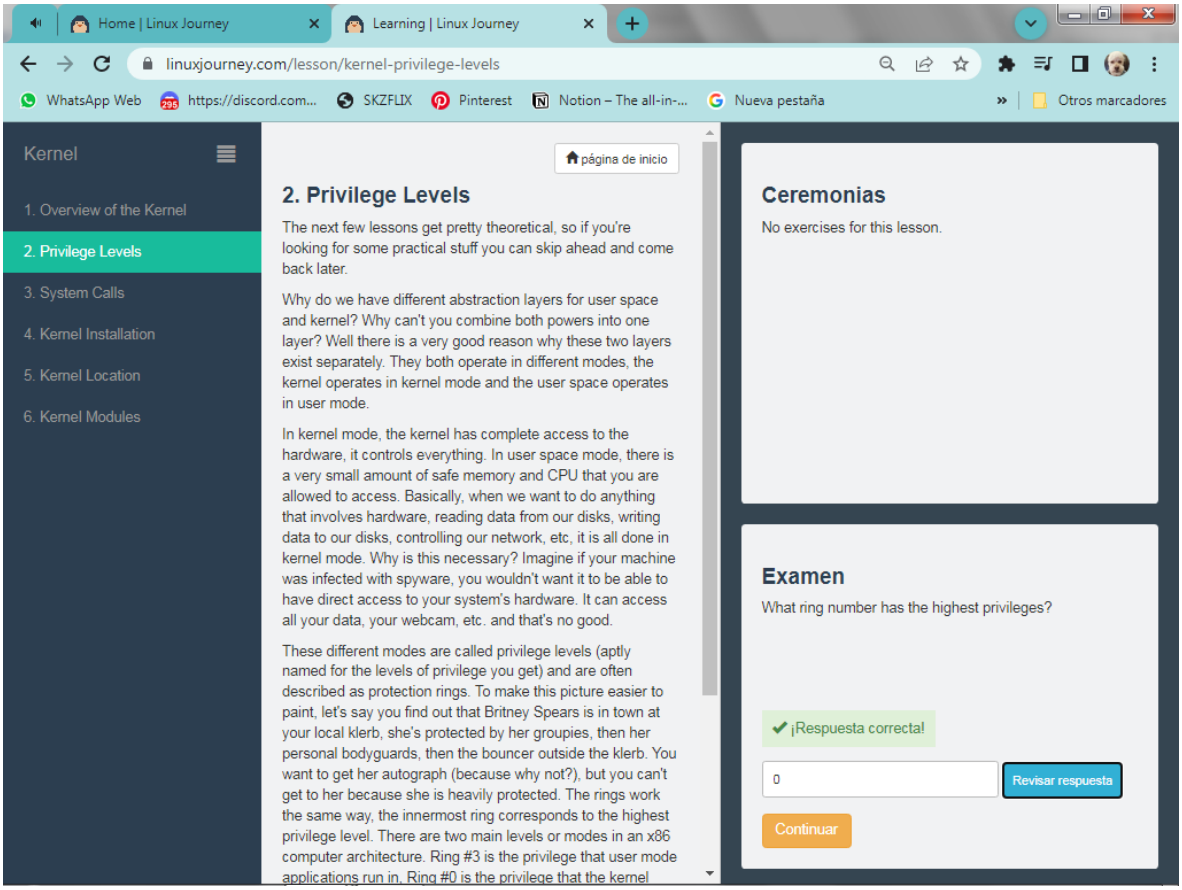

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/boot-process-bootloader
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Boot the System
1. Boot Process Overview
2. Boot Process: BIOS
3. Boot Process: Bootloader
4. Boot Process: Kernel
5. Boot Process: Init
página de inicio
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, efilinux, 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 it's UUID or the device name such as /dev/sda1.
ro - This parameter is pretty standard, it mounts the fileystem 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.
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
https://linuxjourney.com/lesson/boot-process-kernel

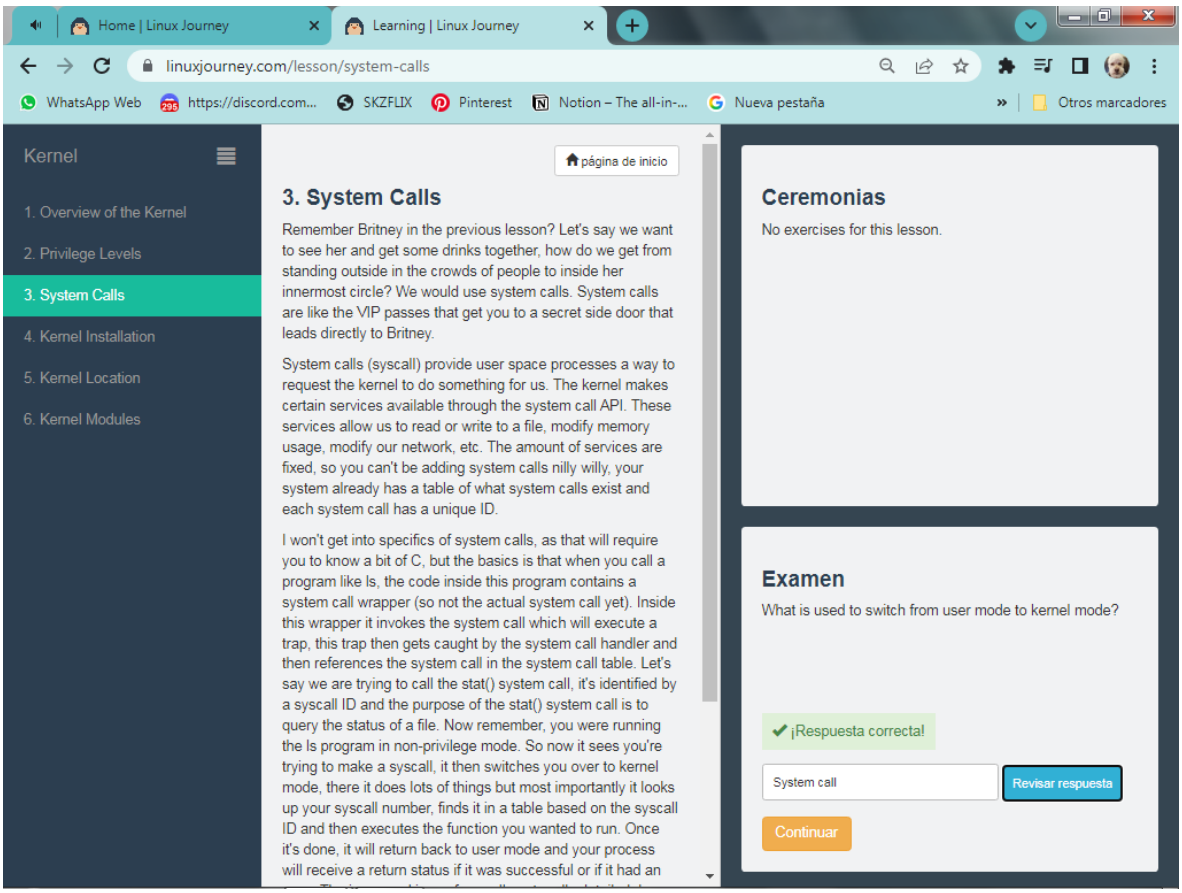

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/boot-process-kernel
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Boot the System
1. Boot Process Overview
2. Boot Process: BIOS
3. Boot Process: Bootloader
4. Boot Process: Kernel
5. Boot Process: Init
página de inicio
4. Boot Process: Kernel
So now that our bootloader has passed on the necessary parameters, let's see how it get's 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 systems 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

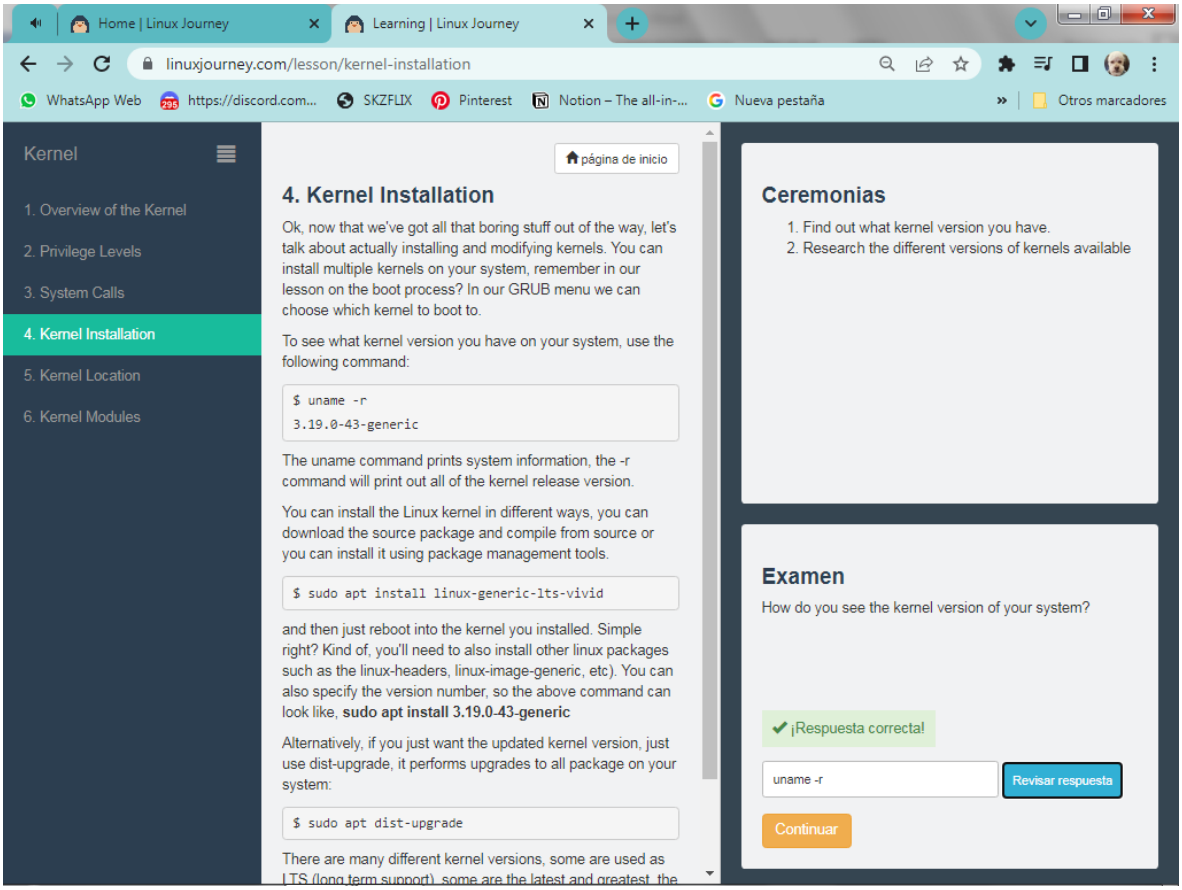

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/boot-process-init
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Boot the System
1. Boot Process Overview
2. Boot Process: BIOS
3. Boot Process: Bootloader
4. Boot Process: Kernel
5. Boot Process: Init
página de inicio
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
https://linuxjourney.com

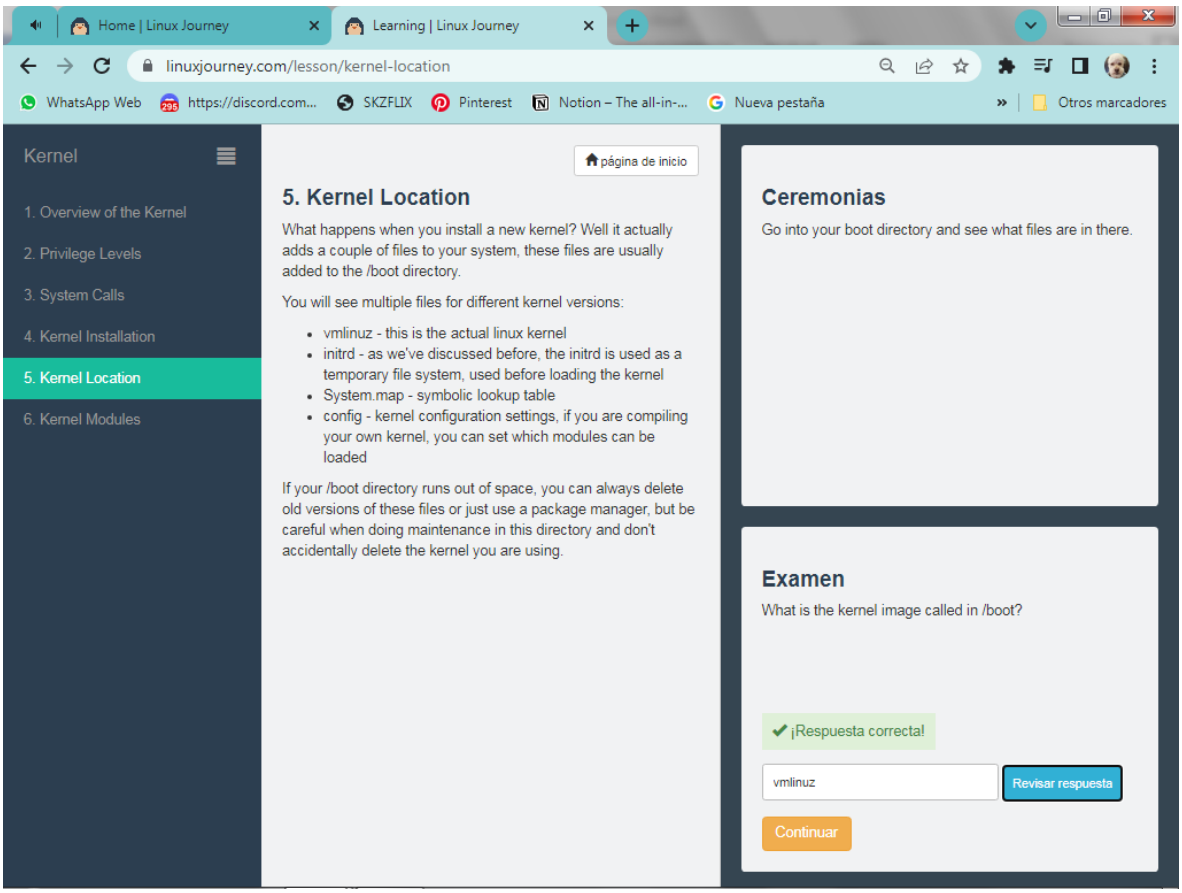
Kernel

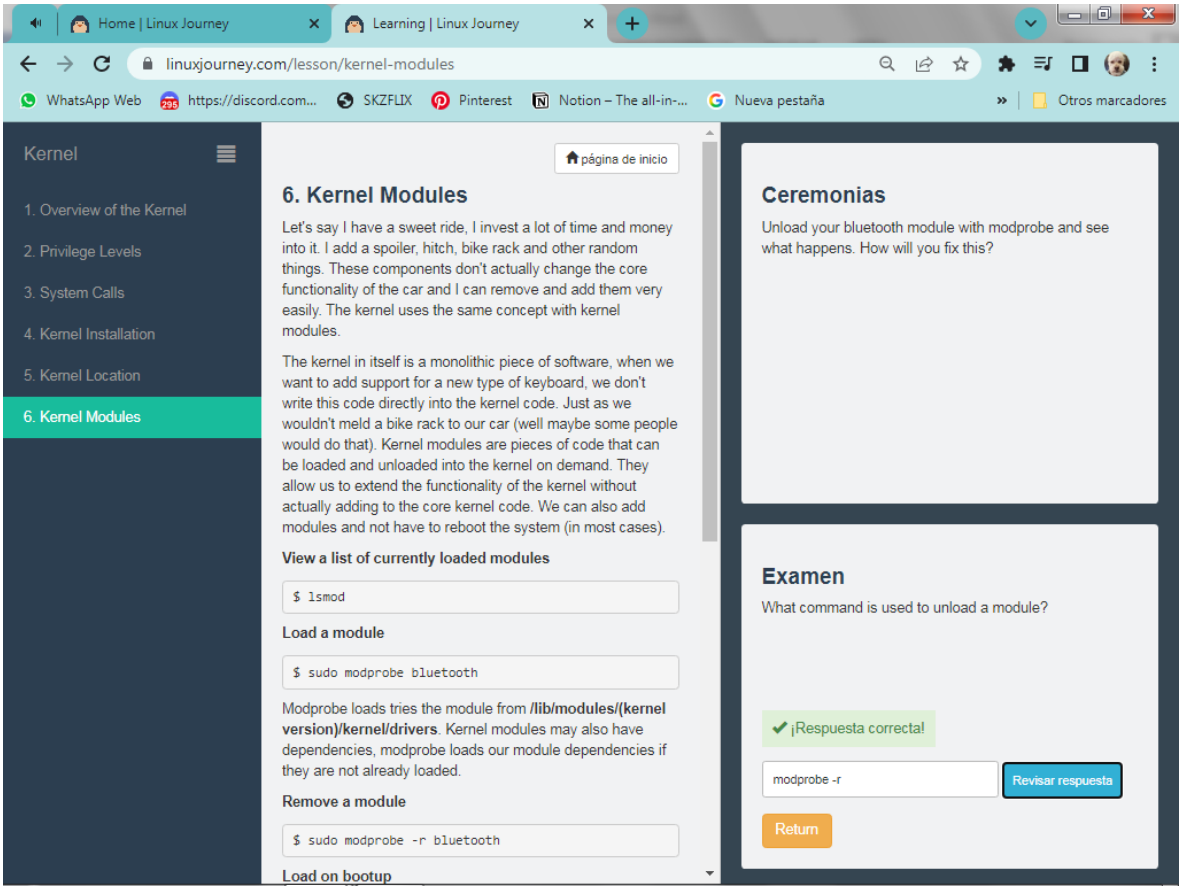

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/kernel-overview
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Kernel
1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules
página de inicio
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


Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/kernel-privilege-levels
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Kernel
página de inicio
1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules
2. Privilege Levels
The next few lessons get pretty theoretical, so if you're looking for some practical stuff you can skip ahead and come back later.
Why do we have different abstraction layers for user space and kernel? Why can't you combine both powers into one layer? Well there is a very good reason why these two layers exist separately. They both operate in different modes, the kernel operates in kernel mode and the user space operates in user mode.
In kernel mode, the kernel has complete access to the hardware, it controls everything. In user space mode, there is a very small amount of safe memory and CPU that you are allowed to access. Basically, when we want to do anything that involves hardware, reading data from our disks, writing data to our disks, controlling our network, etc, it is all done in kernel mode. Why is this necessary? Imagine if your machine was infected with spyware, you wouldn't want it to be able to have direct access to your system's hardware. It can access all your data, your webcam, etc. and that's no good.
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
Ceremonias
No exercises for this lesson.
Examen
What ring number has the highest privileges?
¡Respuesta correcta!
0
Revisar respuesta
Continuar

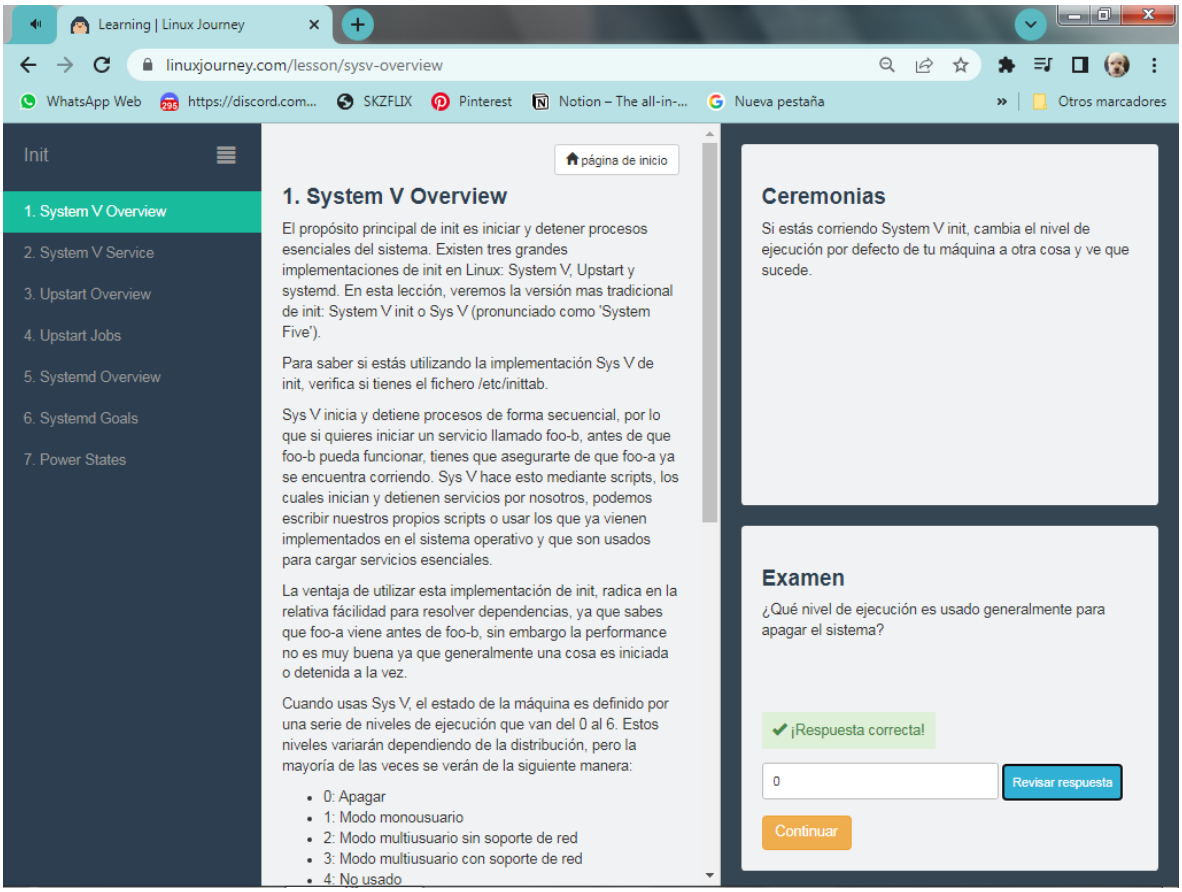

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/system-calls
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Kernel
página de inicio
1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules
3. System Calls
Remember Britney in the previous lesson? Let's say we want to see her and get some drinks together, how do we get from standing outside in the crowds of people to inside her innermost circle? We would use system calls. System calls are like the VIP passes that get you to a secret side door that leads directly to Britney.
System calls (syscall) provide user space processes a way to request the kernel to do something for us. The kernel makes certain services available through the system call API. These services allow us to read or write to a file, modify memory usage, modify our network, etc. The amount of services are fixed, so you can't be adding system calls nilly willy, your system already has a table of what system calls exist and each system call has a unique ID.
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
Ceremonias
No exercises for this lesson.
Examen
What is used to switch from user mode to kernel mode?
¡Respuesta correcta!
System call
Revisar respuesta
Continuar

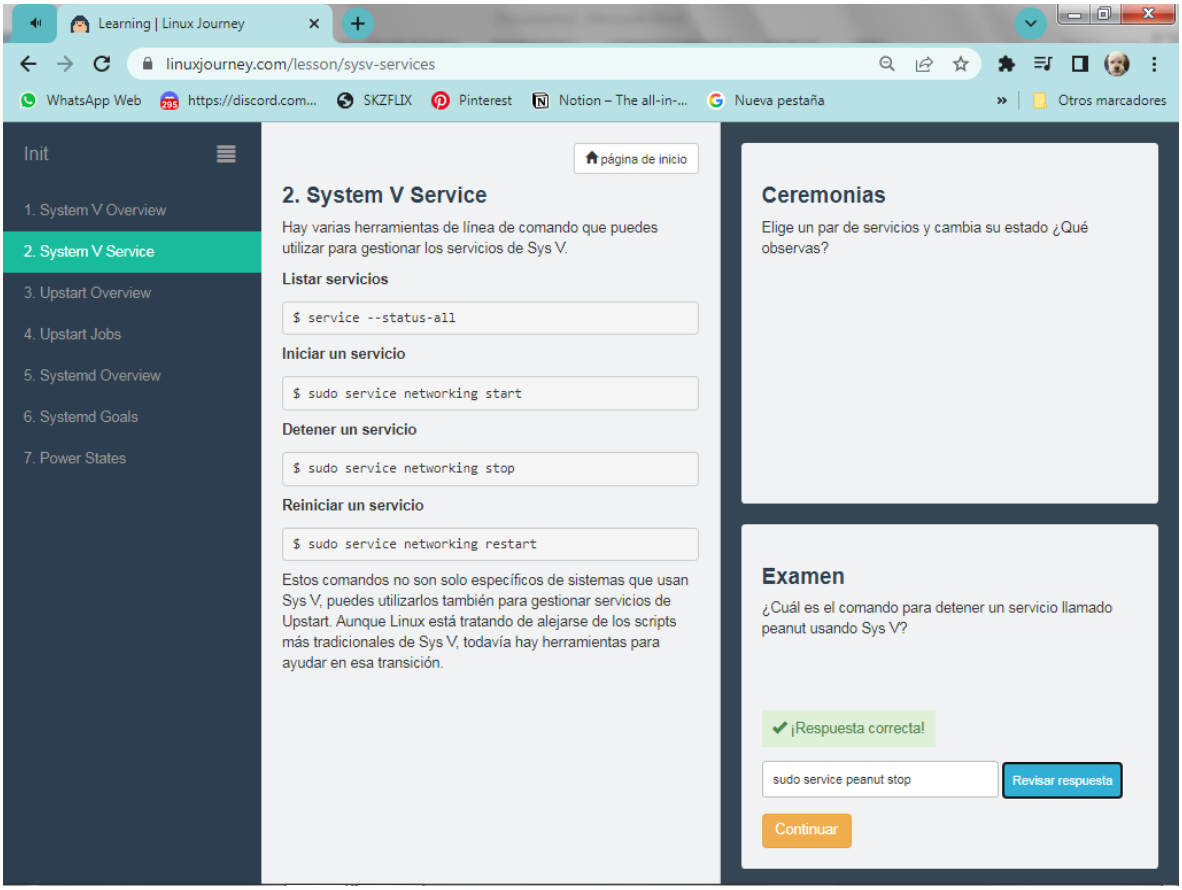

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/kernel-installation
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Kernel
1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules
página de inicio
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
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

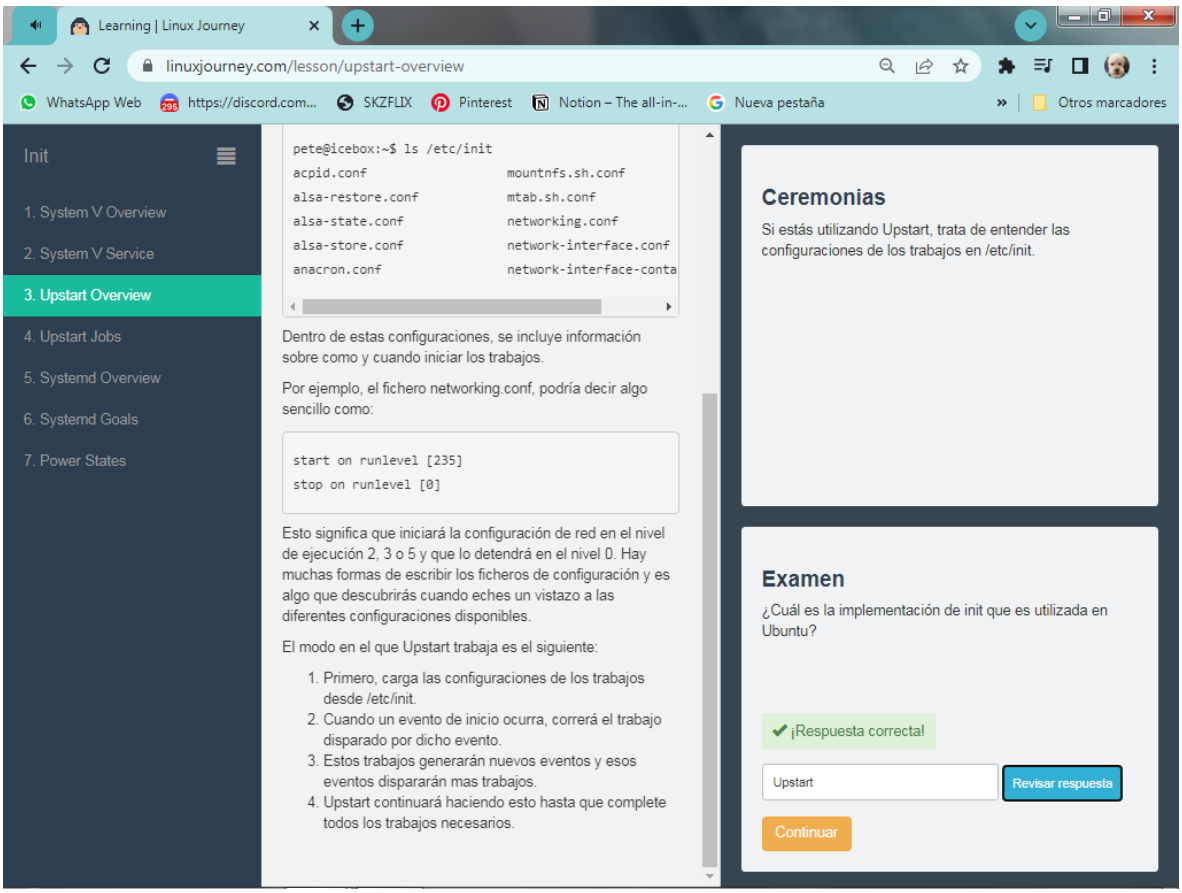

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/kernel-location
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Kernel
1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules
página de inicio
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

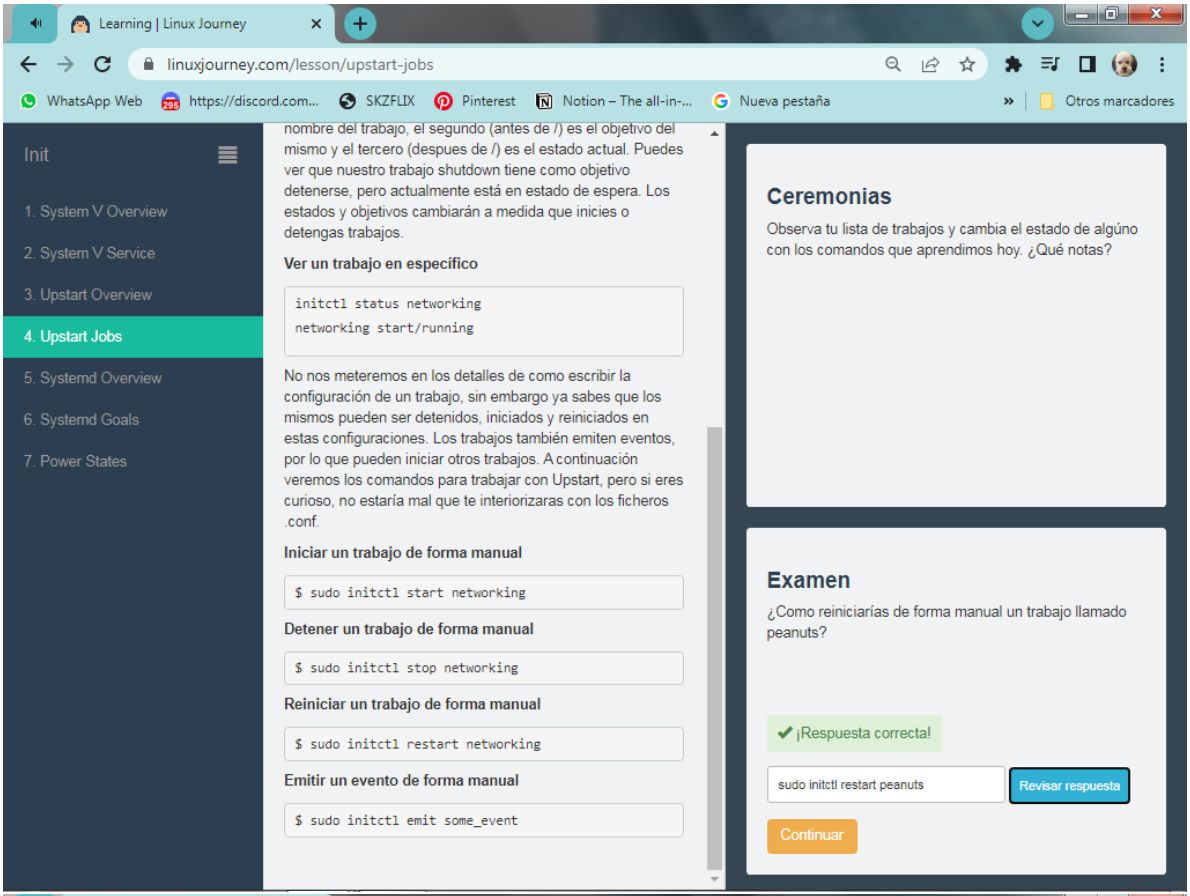

Home | Linux Journey
Learning | Linux Journey
linuxjourney.com/lesson/kernel-modules
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Kernel
1. Overview of the Kernel
2. Privilege Levels
3. System Calls
4. Kernel Installation
5. Kernel Location
6. Kernel Modules
página de inicio
6. Kernel Modules
Let's say I have a sweet ride, I invest a lot of time and money into it. I add a spoiler, hitch, bike rack and other random things. These components don't actually change the core functionality of the car and I can remove and add them very easily. The kernel uses the same concept with kernel modules.
The kernel in itself is a monolithic piece of software, when we want to add support for a new type of keyboard, we don't write this code directly into the kernel code. Just as we wouldn't meld a bike rack to our car (well maybe some people would do that). Kernel modules are pieces of code that can be loaded and unloaded into the kernel on demand. They allow us to extend the functionality of the kernel without actually adding to the core kernel code. We can also add modules and not have to reboot the system (in most cases).
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
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?
¡Respuesta correcta!
modprobe -r
Revisar respuesta
Return

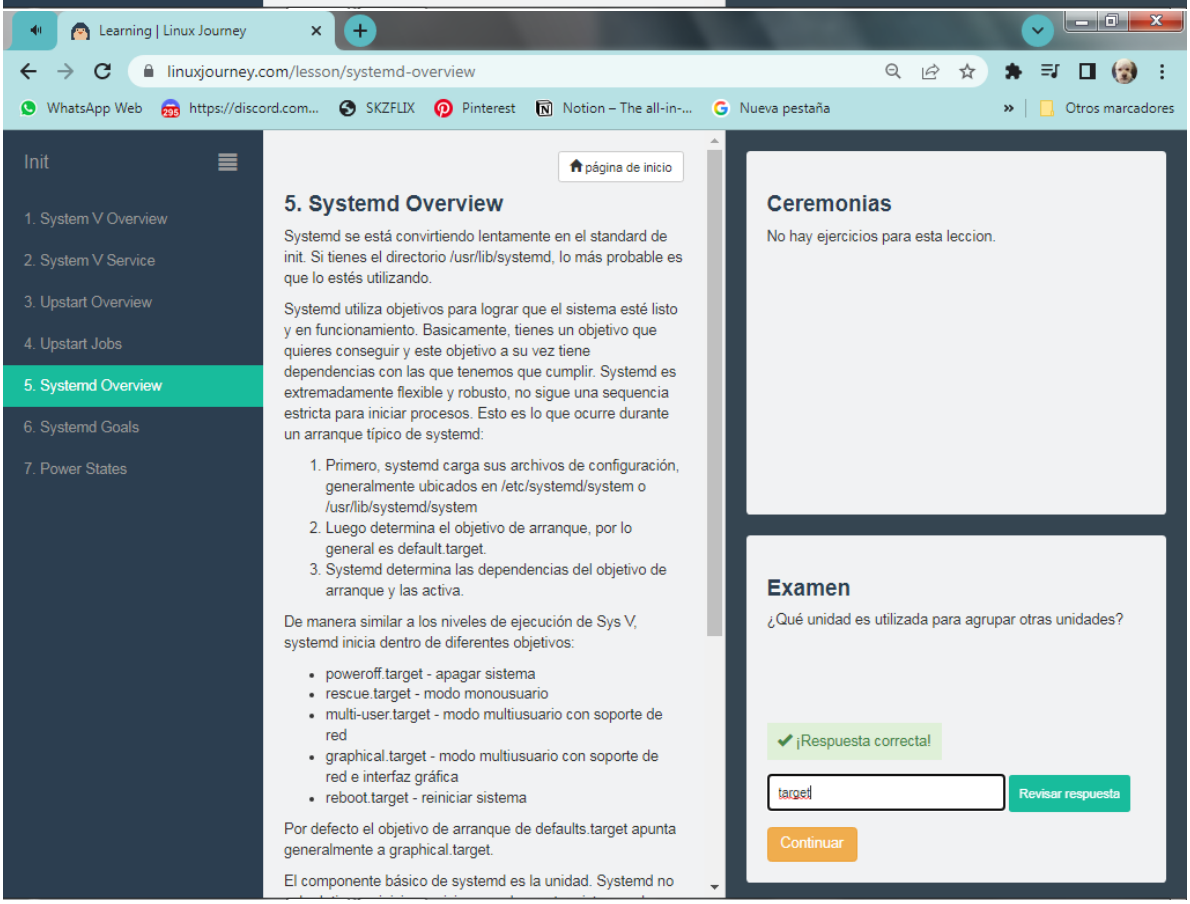
Init

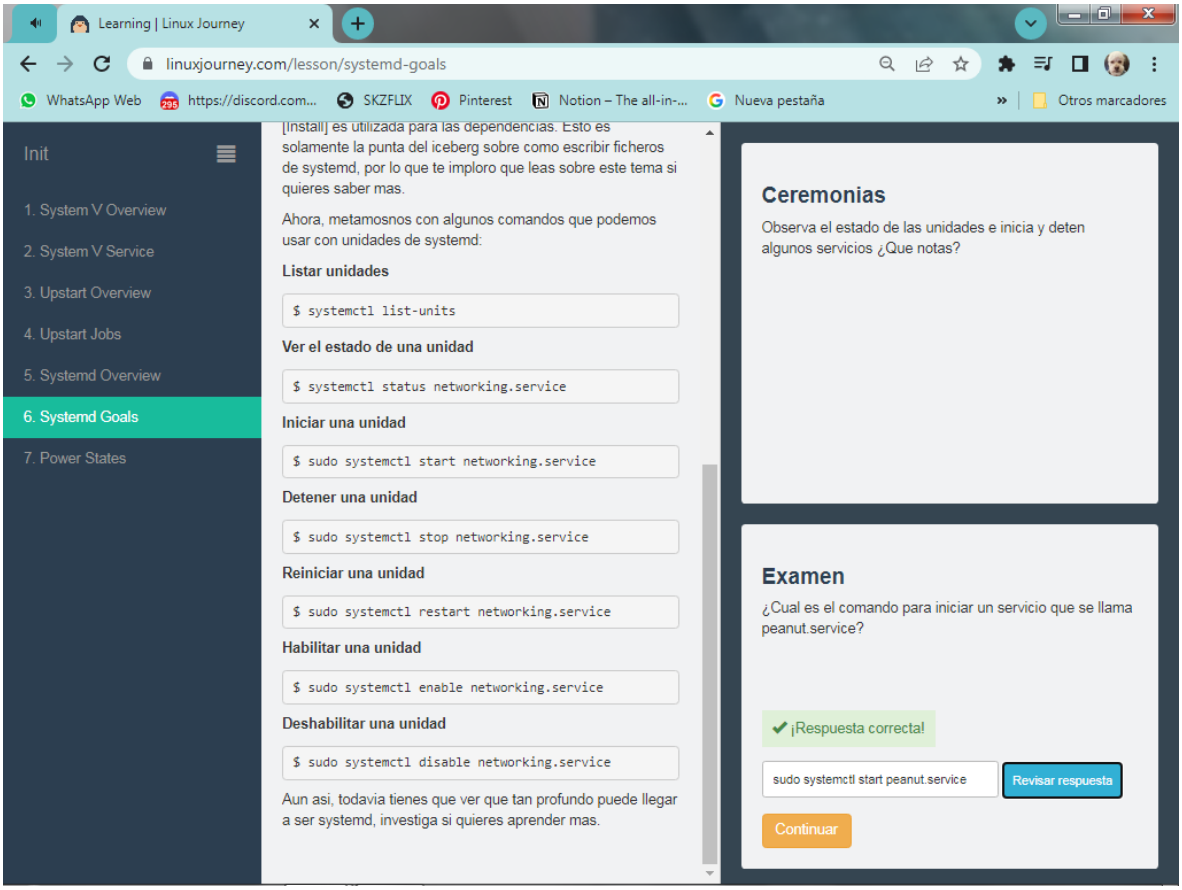

Learning | Linux Journey
linuxjourney.com/lesson/sysv-overview
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Init
1. System V Overview
2. System V Service
3. Upstart Overview
4. Upstart Jobs
5. Systemd Overview
6. Systemd Goals
7. Power States
página de inicio
1. System V Overview
El propósito principal de init es iniciar y detener procesos esenciales del sistema. Existen tres grandes implementaciones de init en Linux: System V, Upstart y systemd. En esta lección, veremos la versión mas tradicional de init: System V init o Sys V (pronunciado como 'System Five').
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 fácilidad 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 variarán 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
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?
¡Respuesta correcta!
0
Revisar respuesta
Continuar

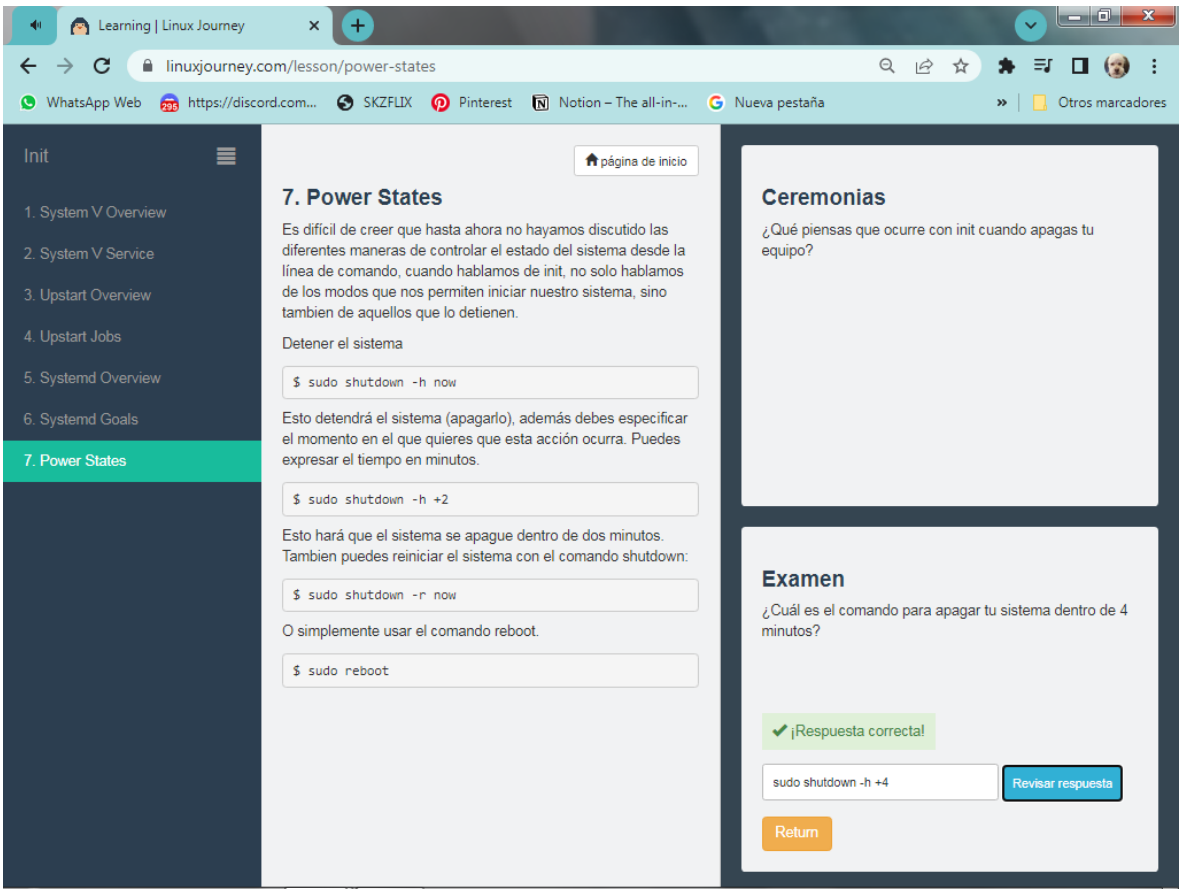

Learning | Linux Journey
linuxjourney.com/lesson/sysv-services
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Init
1. System V Overview
2. System V Service
3. Upstart Overview
4. Upstart Jobs
5. Systemd Overview
6. Systemd Goals
7. Power States
página de inicio
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
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
pete@icebox:~$ ls /etc/init
acpid.conf
alsa-restore.conf
alsa-state.conf
alsa-store.conf
anacron.conf
mountnfs.sh.conf
mtab.sh.conf
networking.conf
network-interface.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 que 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 mas 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

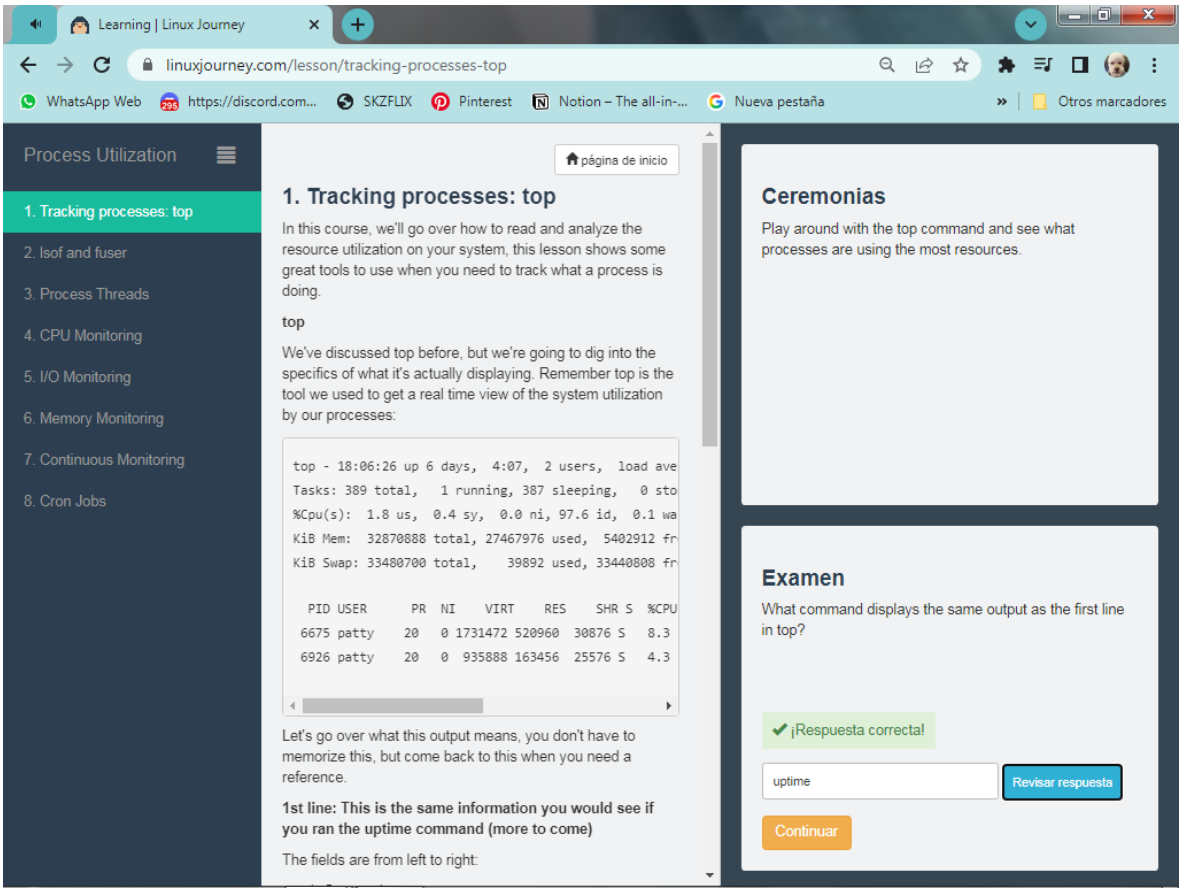

Learning | Linux Journey
linuxjourney.com/lesson/upstart-jobs
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Init
1. System V Overview
2. System V Service
3. Upstart Overview
4. Upstart Jobs
5. Systemd Overview
6. Systemd Goals
7. Power States
nombre del trabajo, el segundo (antes de /) es el objetivo del mismo y el tercero (despues 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 algúno 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

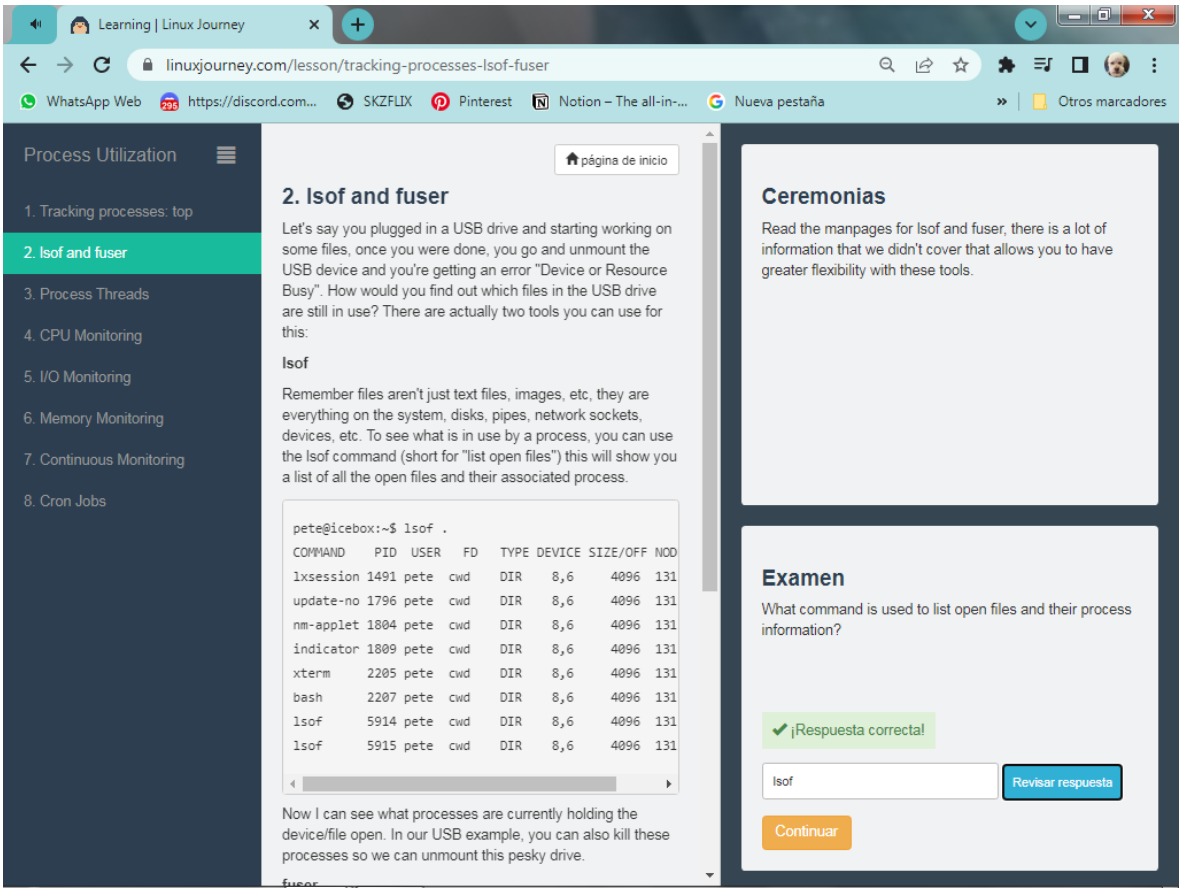

Learning | Linux Journey
linuxjourney.com/lesson/systemd-overview
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Init
1. System V Overview
2. System V Service
3. Upstart Overview
4. Upstart Jobs
5. Systemd Overview
6. Systemd Goals
7. Power 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 sequencia 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 leccion.
Examen
¿Qué unidad es utilizada para agrupar otras unidades?
¡Respuesta correcta!
target
Revisar respuesta
Continuar

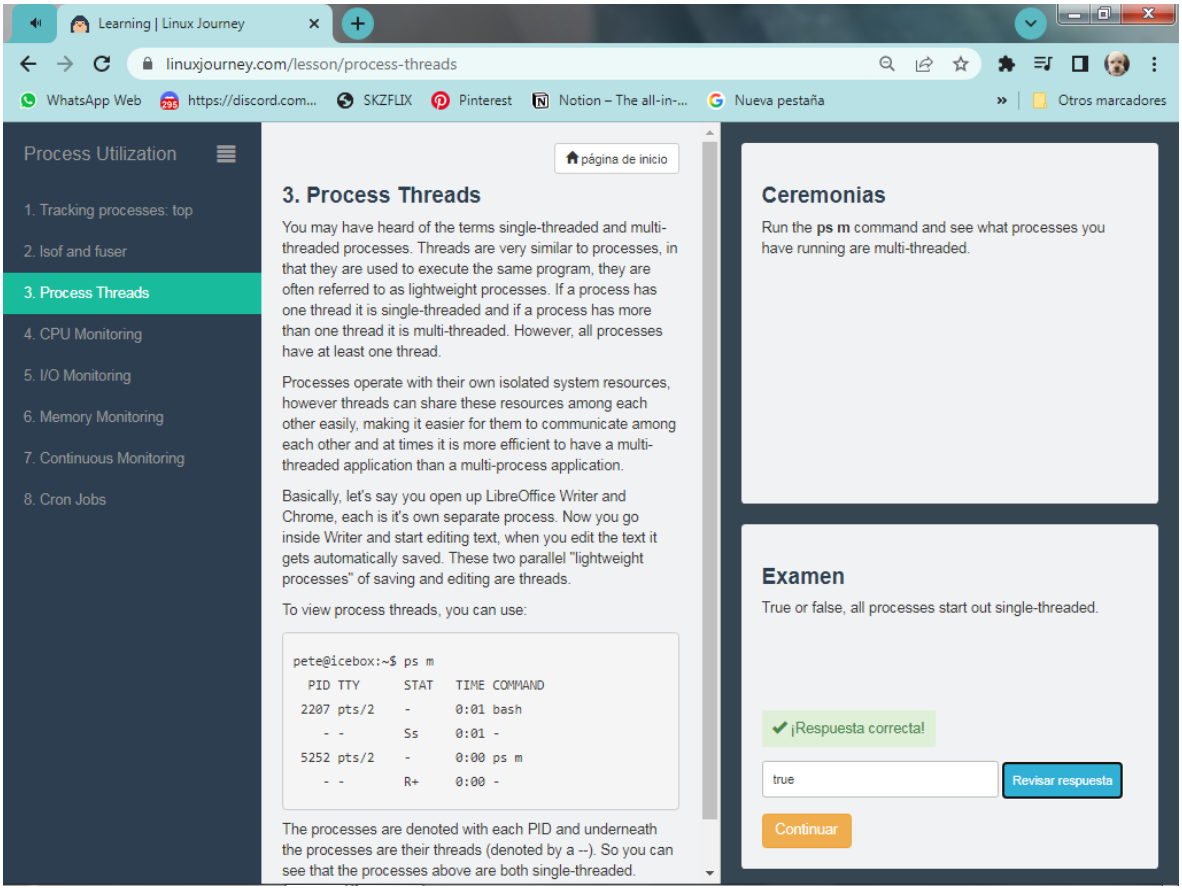

Learning | Linux Journey
linuxjourney.com/lesson/systemd-goals
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Init
1. System V Overview
2. System V Service
3. Upstart Overview
4. Upstart Jobs
5. Systemd Overview
6. Systemd Goals
7. Power States
[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 quieres saber mas.
Ahora, metamosnos 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 quieres aprender mas.
Ceremonias
Observa el estado de las unidades e inicia y deten algunos servicios ¿Que notas?
Examen
¿Cual es el comando para iniciar un servicio que se llama peanut.service?
¡Respuesta correcta!
sudo systemctl start peanut.service
Revisar respuesta
Continuar

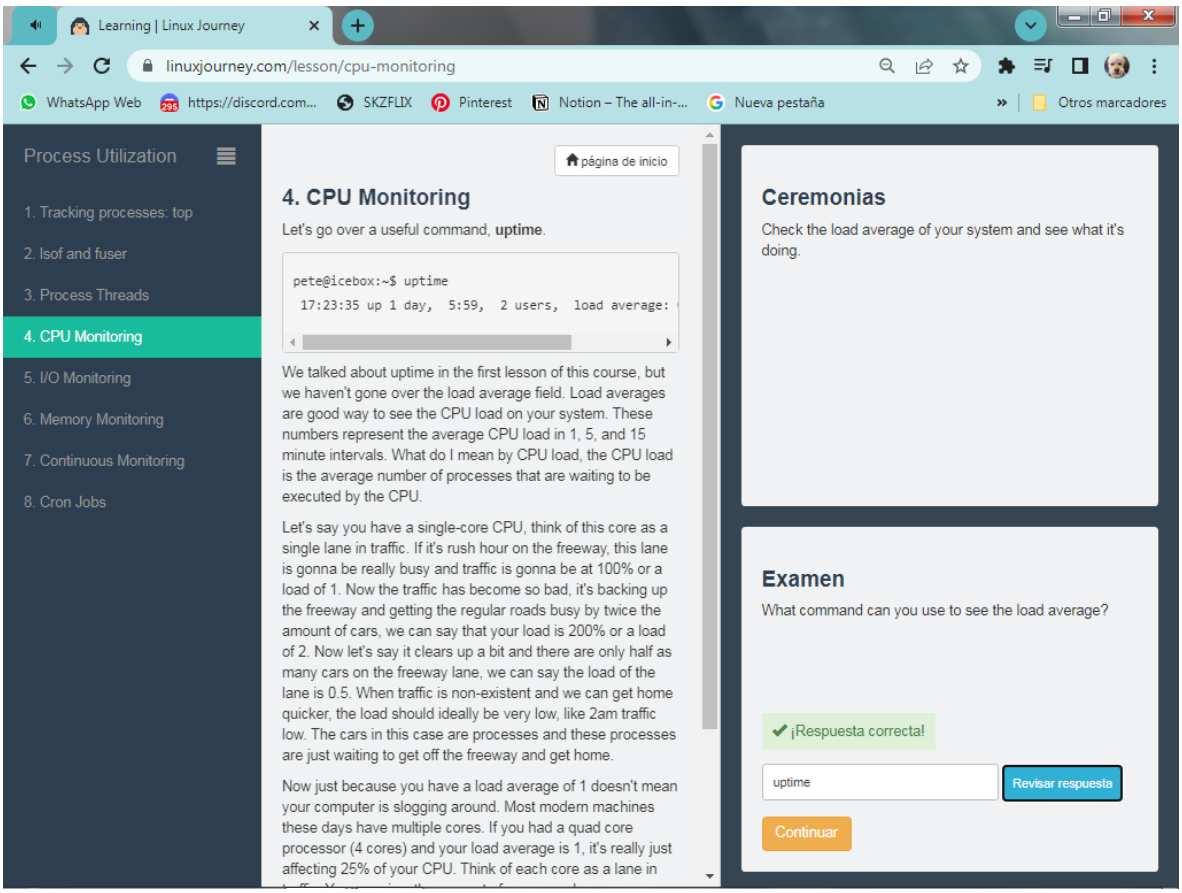

Learning | Linux Journey
linuxjourney.com/lesson/power-states
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Init
1. System V Overview
2. System V Service
3. Upstart Overview
4. Upstart Jobs
5. Systemd Overview
6. Systemd Goals
7. Power States
página de inicio
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 quieres que esta acción ocurra. Puedes expresar el tiempo en minutos.
$ sudo shutdown -h +2
Esto hará que el sistema se apague dentro de dos minutos. Tambien 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

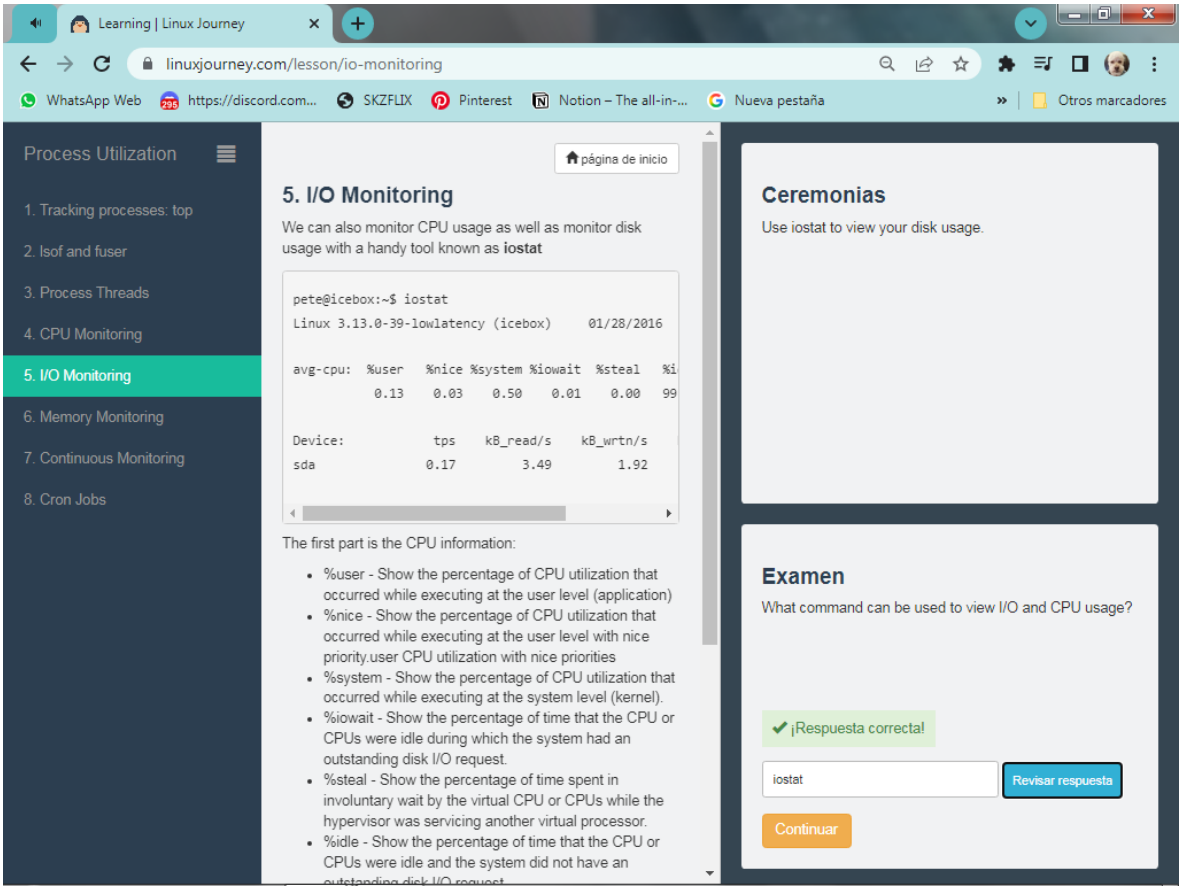
Process Utilization

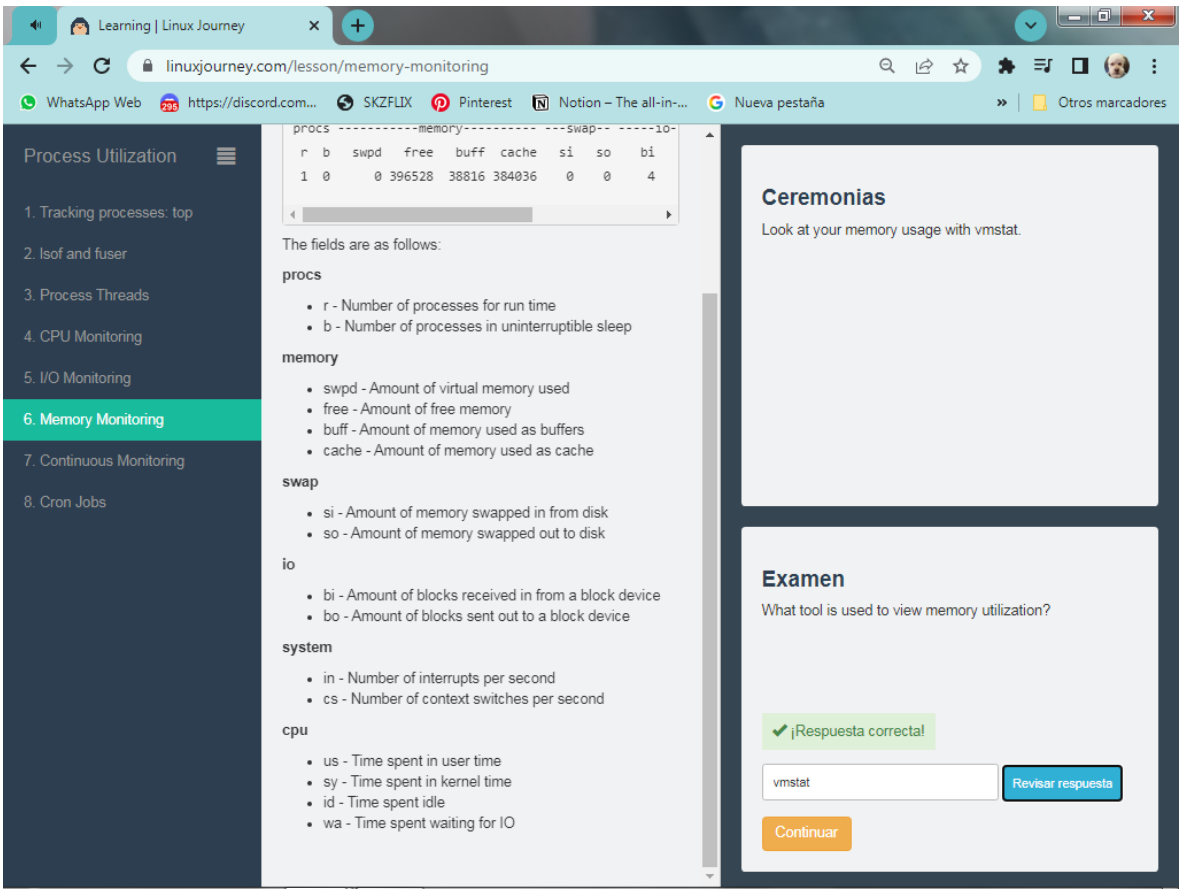


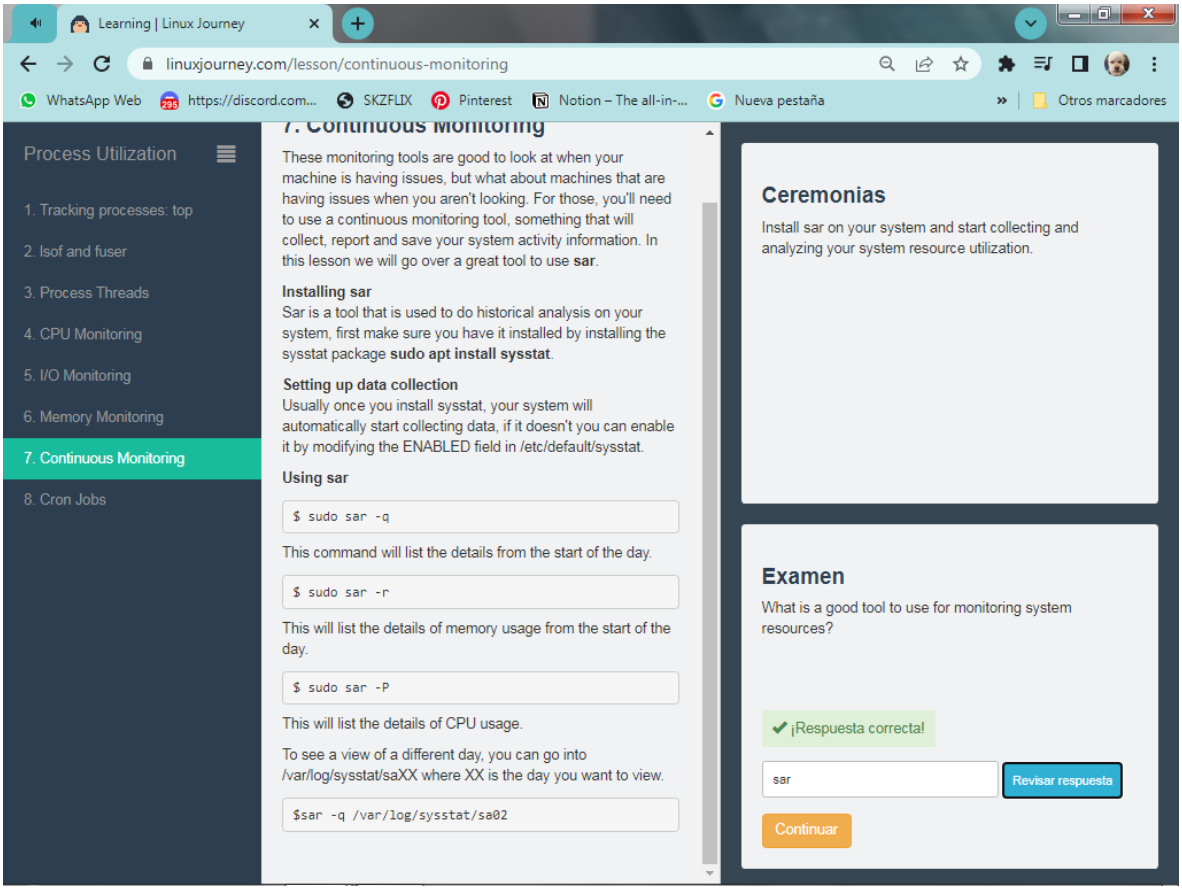


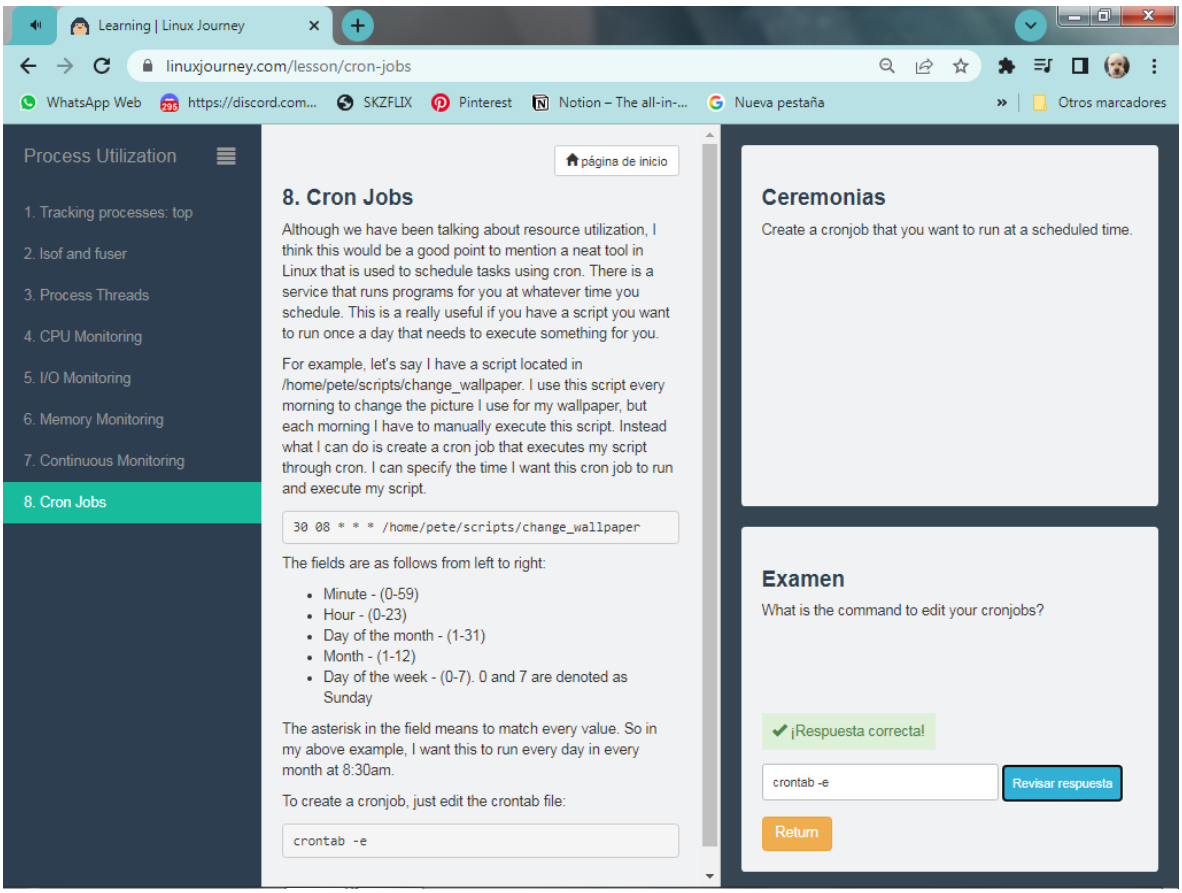

Learning | Linux Journey
linuxjourney.com/lesson/process-threads
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Process Utilization
1. Tracking processes: top
2. lsof and fuser
3. Process Threads
4. CPU Monitoring
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs
página de inicio
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 it's 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
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
página de inicio
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

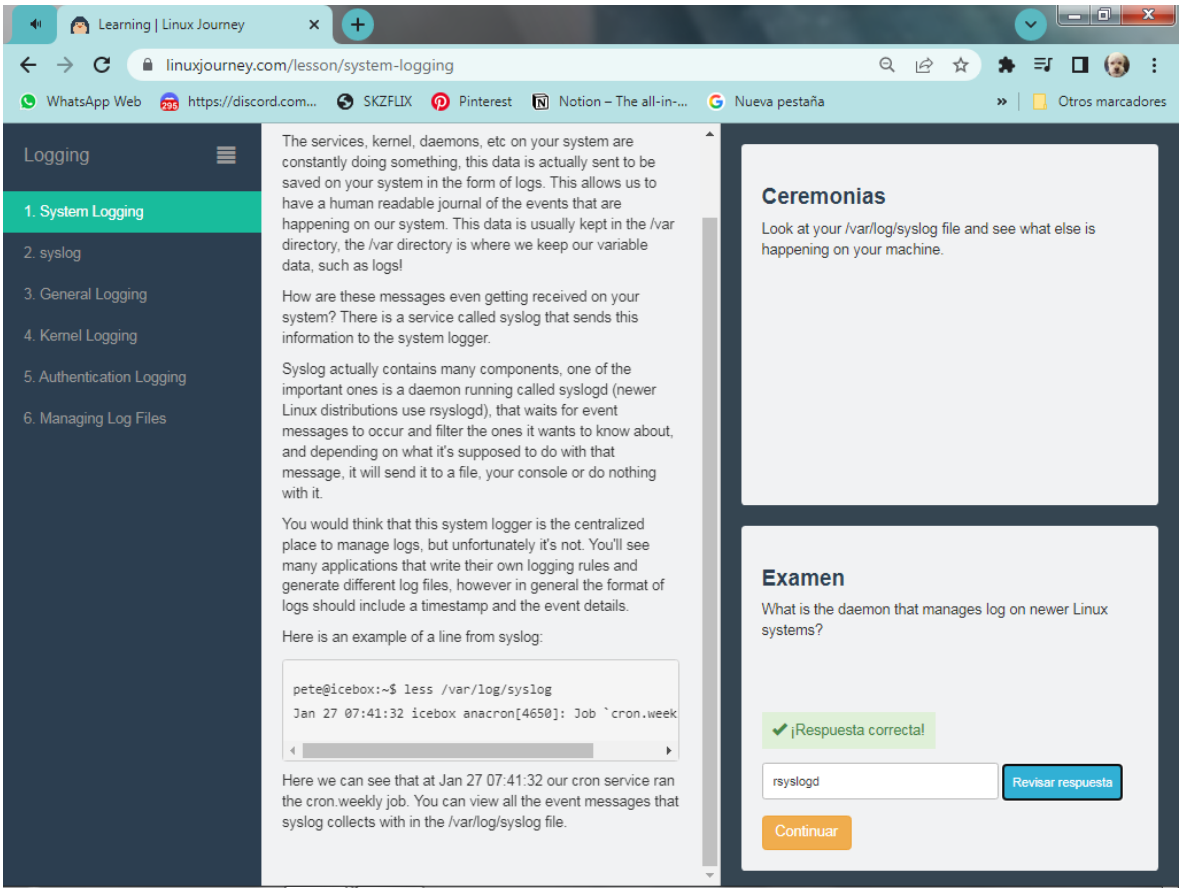

Learning | Linux Journey
linuxjourney.com/lesson/io-monitoring
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Process Utilization
1. Tracking processes: top
2. lsof and fuser
3. Process Threads
4. CPU Monitoring
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs
página de inicio
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
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

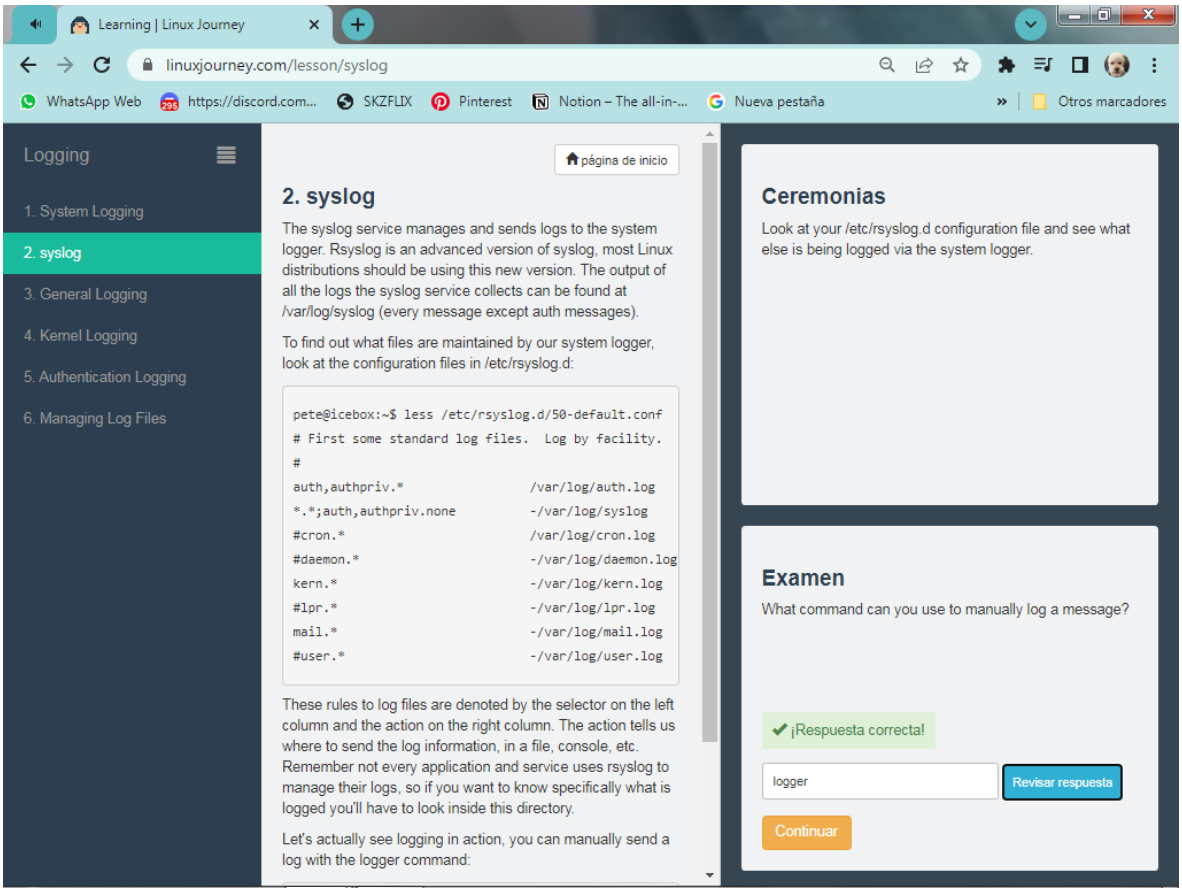

Learning | Linux Journey
linuxjourney.com/lesson/memory-monitoring
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Process Utilization
1. Tracking processes: top
2. lsof and fuser
3. Process Threads
4. CPU Monitoring
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs
procs ----------memory---------- ---swap-- -----io-
r b swpd free buff cache si so bi
1 0 0 396528 38816 384036 0 0 4
The fields are as follows:
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

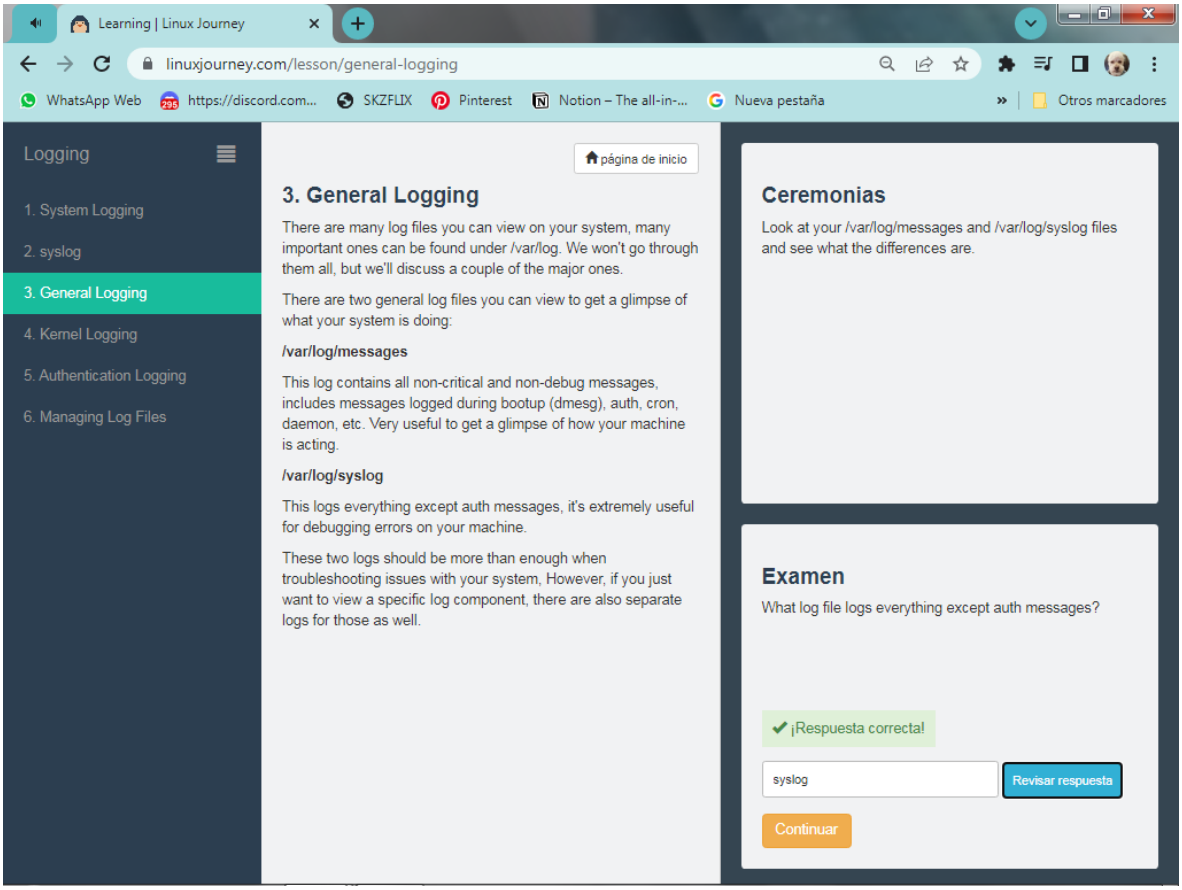

Learning | Linux Journey
linuxjourney.com/lesson/continuous-monitoring
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Process Utilization
1. Tracking processes: top
2. lsof and fuser
3. Process Threads
4. CPU Monitoring
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs
7. Continuous Monitoring
These monitoring tools are good to look at when your machine is having issues, but what about machines that are having issues when you aren't looking. For those, you'll need to use a continuous monitoring tool, something that will collect, report and save your system activity information. In this lesson we will go over a great tool to use sar.
Installing sar
Sar is a tool that is used to do historical analysis on your system, first make sure you have it installed by installing the sysstat package sudo apt install sysstat.
Setting up data collection
Usually once you install sysstat, your system will automatically start collecting data, if it doesn't you can enable it by modifying the ENABLED field in /etc/default/sysstat.
Using sar
$ sudo sar -q
This command will list the details from the start of the day.
$ sudo sar -r
This will list the details of memory usage from the start of the day.
$ sudo sar -P
This will list the details of CPU usage.
To see a view of a different day, you can go into /var/log/sysstat/saXX where XX is the day you want to view.
$sar -q /var/log/sysstat/sa02
Ceremonias
Install sar on your system and start collecting and analyzing your system resource utilization.
Examen
What is a good tool to use for monitoring system resources?
¡Respuesta correcta!
sar
Revisar respuesta
Continuar

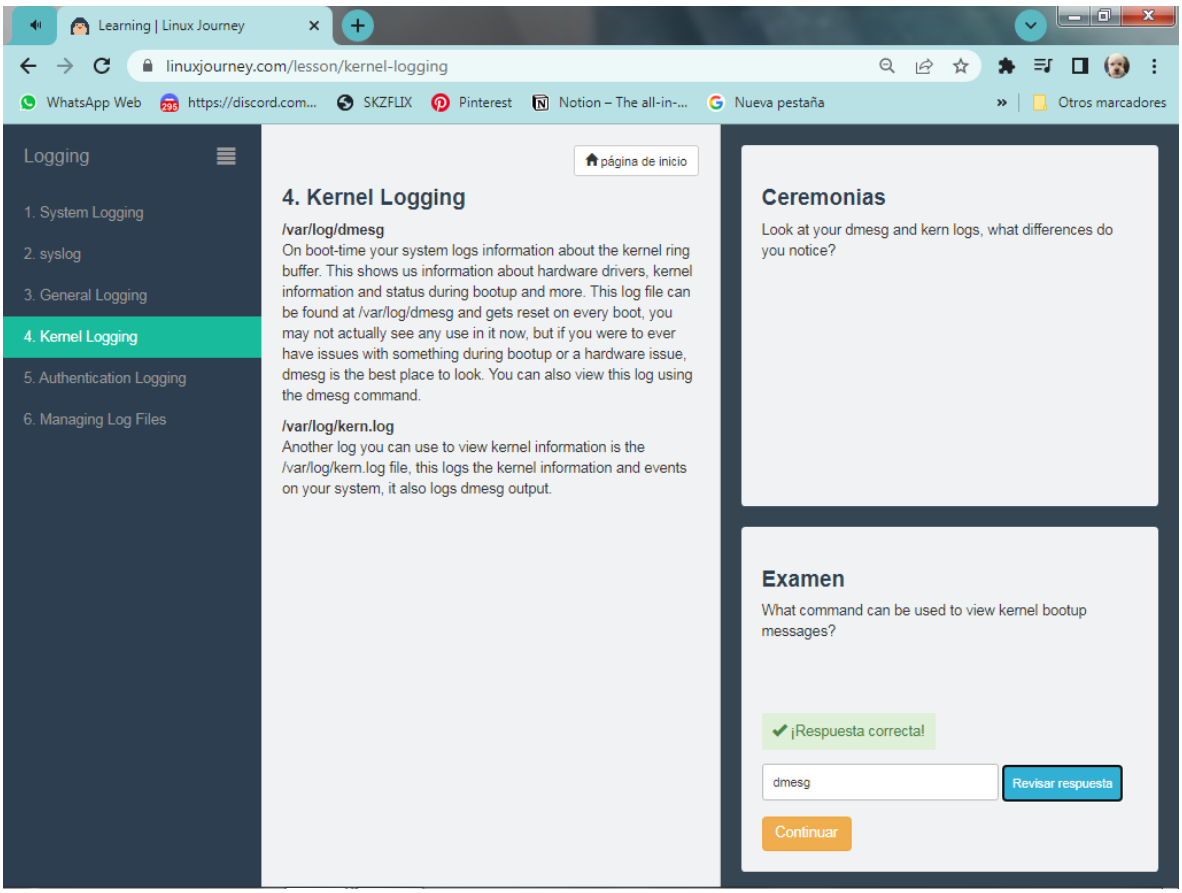

Learning | Linux Journey
linuxjourney.com/lesson/cron-jobs
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Process Utilization
1. Tracking processes: top
2. lsof and fuser
3. Process Threads
4. CPU Monitoring
5. I/O Monitoring
6. Memory Monitoring
7. Continuous Monitoring
8. Cron Jobs
página de inicio
8. Cron Jobs
Although we have been talking about resource utilization, I think this would be a good point to mention a neat tool in Linux that is used to schedule tasks using cron. There is a service that runs programs for you at whatever time you schedule. This is a really useful if you have a script you want to run once a day that needs to execute something for you.
For example, let's say I have a script located in /home/pete/scripts/change_wallpaper. I use this script every morning to change the picture I use for my wallpaper, but each morning I have to manually execute this script. Instead what I can do is create a cron job that executes my script through cron. I can specify the time I want this cron job to run and execute my script.
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
Ceremonias
Create a cronjob that you want to run at a scheduled time.
Examen
What is the command to edit your cronjobs?
¡Respuesta correcta!
crontab -e
Revisar respuesta
Return

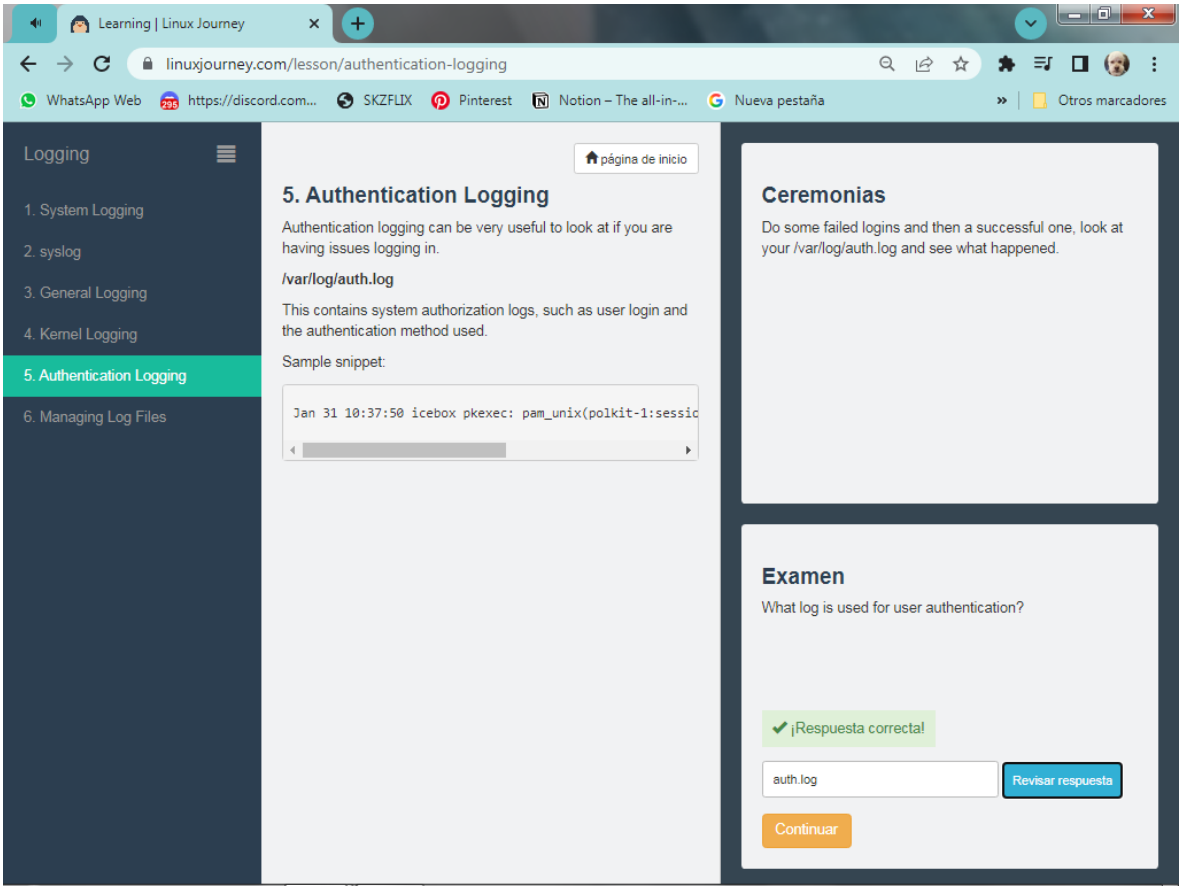
Logging

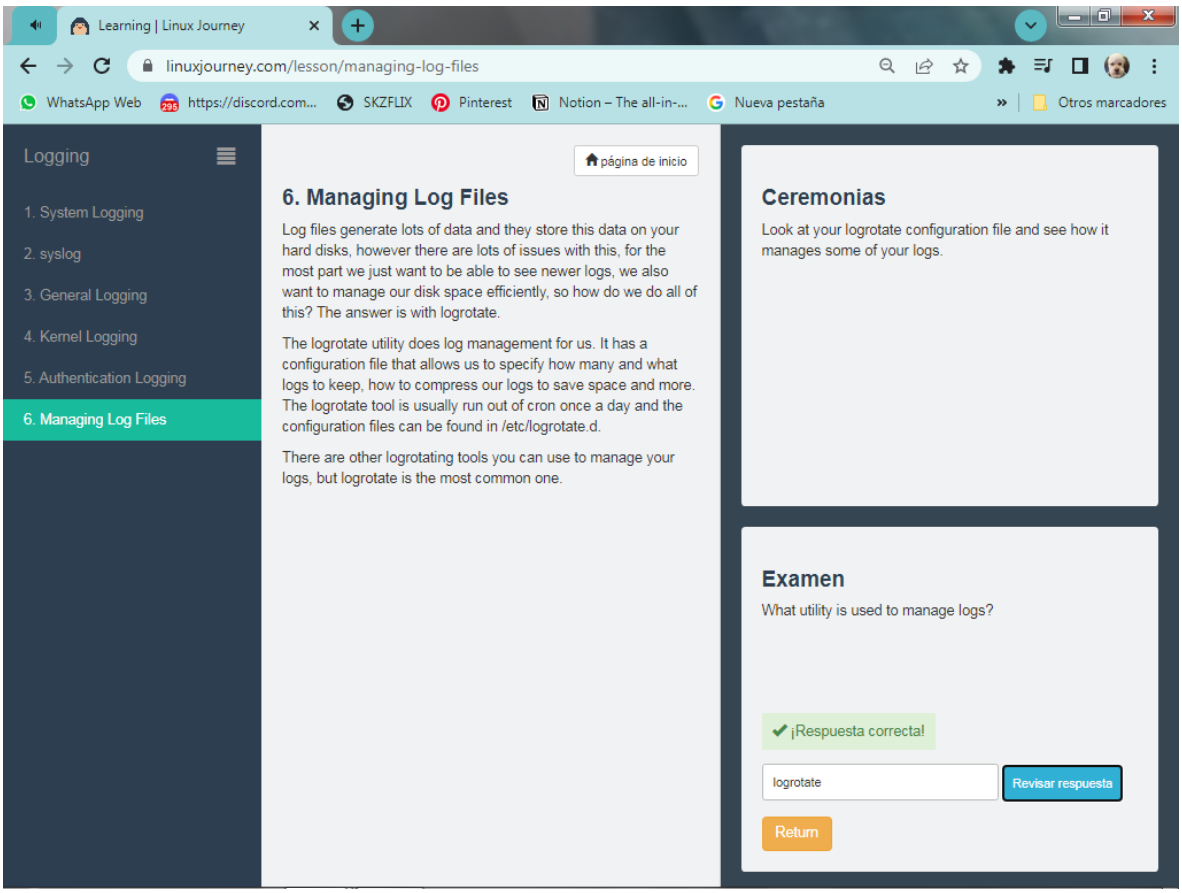

Learning | Linux Journey
linuxjourney.com/lesson/system-logging
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Logging
1. System Logging
2. syslog
3. General Logging
4. Kernel Logging
5. Authentication Logging
6. Managing Log Files
The services, 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
Revisar respuesta
Continuar


Learning | Linux Journey
linuxjourney.com/lesson/syslog
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Logging
1. System Logging
2. syslog
3. General Logging
4. Kernel Logging
5. Authentication Logging
6. Managing Log Files
página de inicio
2. syslog
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
Revisar respuesta
Continuar


Learning | Linux Journey
linuxjourney.com/lesson/general-logging
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Logging
1. System Logging
2. syslog
3. General Logging
4. Kernel Logging
5. Authentication Logging
6. Managing Log Files
página de inicio
3. General Logging
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/messages
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.
/var/log/syslog
This logs everything except auth messages, it's extremely useful for debugging errors on your machine.
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!
syslog
Revisar respuesta
Continuar


Learning | Linux Journey
linuxjourney.com/lesson/kernel-logging
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Logging
1. System Logging
2. syslog
3. General Logging
4. Kernel Logging
5. Authentication Logging
6. Managing Log Files
página de inicio
4. Kernel Logging
/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!
dmesg
Revisar respuesta
Continuar


Learning | Linux Journey
linuxjourney.com/lesson/authentication-logging
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Logging
1. System Logging
2. syslog
3. General Logging
4. Kernel Logging
5. Authentication Logging
6. Managing Log Files
página de inicio
5. Authentication Logging
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:sessic
Ceremonias
Do some failed logins and then a successful one, look at your /var/log/auth.log and see what happened.
Examen
What log is used for user authentication?
¡Respuesta correcta!
auth.log
Revisar respuesta
Continuar


Learning | Linux Journey
linuxjourney.com/lesson/managing-log-files
WhatsApp Web
https://discord.com...
SKZFLIX
Pinterest
Notion – The all-in-...
Nueva pestaña
Otros marcadores
Logging
1. System Logging
2. syslog
3. General Logging
4. Kernel Logging
5. Authentication Logging
6. Managing Log Files
página de inicio
6. Managing Log Files
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.
Ceremonias
Look at your logrotate configuration file and see how it manages some of your logs.
Examen
What utility is used to manage logs?
¡Respuesta correcta!
logrotate
Revisar respuesta
Return