

Trabajaremos con todos estos archivos en el modulo 1. Tomate tu tiempo para hacer todo como debes, como sabes y para la facilidad de Hector.

=====

Archivo 1: A1_Conceptos_importantes.pdf

Tema: Introducción a Linux - Conceptos Fundamentales

Premisa del Curso

Explicación técnica:

El curso se basa en la premisa "Cómo funcionan las cosas", enfocándose en la comprensión profunda de herramientas, entornos, diferencias entre sistemas y utilidades. El objetivo es desarrollar el perfil técnico necesario para llevar adelante proyectos, idear soluciones y participar activamente en el proceso evolutivo de la informática.^[1]

Analogía simple:

Es como aprender a manejar: no solo memorizas las reglas de tránsito, sino que entendés por qué existen, cómo funciona el motor, y así podés arreglar problemas o manejar cualquier auto que te den.

Los Tres Pilares de GNU/Linux

Explicación técnica:

Las distribuciones GNU/Linux se construyen sobre tres pilares fundamentales:

- **Red Hat y sus derivados** (ejemplo: CentOS)
- **Debian y sus derivados** (ejemplo: Ubuntu)
- **Distribuciones espartanas** (orientadas a usuarios avanzados)^[1]

Analogía simple:

Son como tres familias de autos: Ford y sus derivados, Toyota y sus derivados, y los autos de carrera (más complicados pero súper personalizables).

Componentes de una Distribución

Explicación técnica:

Toda distribución GNU/Linux está compuesta por tres elementos perfectamente definidos:

- **El kernel Linux:** núcleo del sistema operativo
- **Las herramientas del sistema operativo GNU:** conjunto de utilidades básicas
- **Herramientas y características propias de la distribución:** personalizaciones específicas^[1]

Analogía simple:

Es como una casa: el kernel es la estructura (cimientos, paredes), GNU son los servicios básicos (electricidad, agua), y las características propias son la decoración y muebles que cada familia elige.

=====

Archivo 2: A2_Que_es_Unix.pdf

Tema: ¿Qué es UNIX?

Definición e Historia

Explicación técnica:

UNIX es una familia de sistemas operativos cuya primera versión fue creada entre 1969 y 1970 en el centro de investigación de Bell Labs. Los principales desarrolladores fueron Ken Thompson, Dennis Ritchie, Joe Ossanna y Douglas McIlroy.^[2]

Analogía simple:

UNIX es como el "abuelo" de muchos sistemas operativos modernos. Imaginate que es el primer auto que se inventó, y todos los autos de hoy tienen algo de ese diseño original.

Importancia Histórica

Explicación técnica:

UNIX fue revolucionario porque introdujo funcionalidades novedosas para la época: multitarea (ejecutar varios programas simultáneamente) y multiusuario (varios usuarios trabajando al mismo tiempo). Su evolución fue paralela al lenguaje C, lo que permitió su portabilidad a múltiples arquitecturas.^[2]

Analogía simple:

Era como pasar de un teléfono que solo podía hacer una llamada por vez, a uno que podía manejar varias llamadas y que varias personas pudieran usarlo sin problemas.

Tipos de Sistemas UNIX

Explicación técnica:

Según Eric Raymond, existen tres tipos de UNIX:

Tipo	Característica	Ejemplos
Genéticos	Derivados del UNIX original de Bell Labs	FreeBSD, NetBSD, OpenBSD, AIX, Oracle Solaris, macOS, HP-UX ^[2]
Registrados	UNIX marca registrada certificada por The Open Group	Oracle Solaris, macOS, HP-UX ^[2]
Funcionales	Funcionalidades UNIX sin heredar su código	Linux, Minix ^[2]

Analogía simple:

- **Genéticos:** Son como los hijos directos de una familia, heredan los genes.
- **Registrados:** Tienen el "apellido oficial" de la familia.
- **Funcionales:** Actúan como la familia pero no comparten sangre.

Innovaciones Introducidas por UNIX

Explicación técnica:

UNIX introdujo conceptos fundamentales que siguen vigentes:

- Sistema de archivos jerárquico
- Shell para comandos y programación de scripts
- Archivos de texto para configuración
- Sintaxis de expresiones regulares
- Herramientas especializadas y eficientes
- Desarrollo modular de software
- Suite de protocolos TCP/IP
- Software documentado con código fuente accesible ^[2]

Analogía simple:

Es como si UNIX hubiera inventado las "reglas básicas" que todos los sistemas modernos usan: cómo organizar archivos (como cajones), cómo hablar con la computadora (como un idioma), y cómo conectarse a internet.

=====

Archivo 3: A3_Manerasde_InstalarLinux.pdf

Tema: Maneras de instalar Linux

Métodos de Instalación Tradicionales

Explicación técnica:

Existen varios métodos para instalar Linux:

- **CD/DVD:** Disco que contiene un instalador directo
- **LiveCD:** Sistema completo que se ejecuta desde RAM sin modificar discos duros, permitiendo probar el sistema antes de instalarlo
- **USB removable:** LiveCD en formato USB para equipos sin lectora de CD^[3]

Analogía simple:

- **CD/DVD:** Como ir a una casa nueva con las herramientas para construirla desde cero.
- **LiveCD:** Como alquilar una casa amueblada para probarla antes de comprarla.
- **USB:** Lo mismo que el LiveCD, pero en una "llave" más moderna.

Virtualización

Explicación técnica:

Linux puede implementarse como máquina virtual usando tecnologías como:

- VirtualBox, VMWare (virtualizadores de escritorio)
- KVM (virtualización a nivel kernel)
- lxc, docker (contenedores)^[3]

Analogía simple:

Es como tener una "computadora dentro de tu computadora". Podés probar Linux sin tocar tu sistema principal, como tener un laboratorio seguro para experimentar.

Instalación en la Nube

Explicación técnica:

Linux puede desplegarse en proveedores de infraestructura como servicio (IaaS):

- Amazon EC2
- Rackspace
- Digital Ocean
- OpenStack (nube privada)^[3]

Analogía simple:

En vez de comprar una computadora física, "alquilás" una computadora en internet que alguien más mantiene, como usar Netflix en vez de comprar DVDs.

Reemplazo de Software de Windows

Explicación técnica:

Existen tres estrategias para reemplazar aplicaciones privativas:

1. **Funcionalidades similares:** Programas que hacen lo mismo pero se ven diferentes (ej: Calligra Words, AbiWord)
2. **Similitud visual y funcional:** Programas que replican la experiencia de Windows (ej: LibreOffice, OnlyOffice)
3. **Alternativas completamente diferentes:** Nuevos paradigmas de trabajo (ej: Markdown con Mark Text)^[3]

Analogía simple:

1. Cambiar de auto manual a automático: hace lo mismo, pero se maneja distinto.
2. Cambiar de un Ford a un Chevrolet: muy parecidos en todo.
3. Cambiar de auto a bicicleta eléctrica: completamente diferente pero te lleva igual.

=====

Archivo 4: A4_OrigenesdeLinux.pdf

Tema: Orígenes de Linux

El Nacimiento de Linux

Explicación técnica:

Linus Torvalds creó el núcleo Linux en 1991 en la Universidad de Helsinki, Finlandia. El núcleo (kernel) es un programa fundamental que determina el uso de memoria RAM, conexiones de red, control de hardware e interacción con el resto del software. Inicialmente se desarrolló sobre el sistema Minix, luego reemplazó las herramientas Minix por las del proyecto GNU.^[4]

Analogía simple:

Linus era como un estudiante que no le gustaba el auto de la universidad (Minix), entonces decidió construir su propio motor (kernel). Al principio usó algunas partes del auto viejo, pero después las fue reemplazando por partes mejores (GNU).

Evolución de las Versiones

Explicación técnica:

- **Versión 0.02** (5 octubre 1991): Primera versión pública, podía ejecutar bash y gcc
- **Versión 0.10** (finales 1991): Primera versión considerada oficial
- **Versión 0.11** (diciembre 1991): Linus invitó a otros a colaborar
- **Versión 0.12** (enero 1992): Primera versión con licencia GNU GPL^[4]

Analogía simple:

Como un proyecto escolar que empezó siendo solo para Linus, luego lo mostró a algunos compañeros, después invitó a todos a ayudar, y finalmente lo hizo "código libre" para que cualquiera pudiera usarlo y mejorarlo.

Características Principales de Linux

Explicación técnica:

Característica	Descripción técnica
Multitarea total	Ejecución simultánea de múltiples tareas y acceso a varios dispositivos ^[4]
Memoria virtual	Simulación de mayor cantidad de memoria mediante técnicas de virtualización ^[4]
Soporte multiusuario	Acceso simultáneo de múltiples usuarios sin conflictos, cada uno con su espacio ^[4]
Código fuente libre	Miles de programas y librerías con licencias de software libre ^[4]
Adaptabilidad	Amplia variedad de software disponible y compatibilidad ^[4]
Estabilidad	Gestión robusta de procesos internos, resistente a fallos ^[4]

Analogía simple:

- **Multitarea:** Como un chef que puede cocinar varios platos al mismo tiempo.
- **Memoria virtual:** Como tener un armario mágico que parece más grande de lo que es.
- **Multiusuario:** Como un edificio donde cada persona tiene su departamento privado.
- **Código libre:** Como recetas de cocina que todos pueden usar y mejorar.
- **Estabilidad:** Como un auto que nunca se rompe.

Seguridad en Linux

Explicación técnica:

Linux implementa un sistema de seguridad basado en el usuario **root** que posee control absoluto, mientras que el resto de usuarios tienen permisos limitados. Los virus tienen dificultades porque: 1) La mayoría no fueron desarrollados para Linux, 2) Los permisos de ejecución están limitados por defecto, 3) Para afectar todo el sistema necesitan privilegios de root. ^[4]

Analogía simple:

Es como un edificio con un administrador (root) que tiene llaves de todo, pero cada inquilino solo puede entrar a su departamento. Si alguien malo quiere hacer daño, necesita robarle las llaves al administrador, lo cual es muy difícil.

Software Libre vs Gratis

Explicación técnica:

Según la Free Software Foundation, software libre significa que la licencia permite:

1. Utilizar el programa para cualquier propósito
2. Estudiar el funcionamiento y modificarlo para necesidades propias
3. Redistribuir copias del programa
4. Mejorar el programa y compartir las mejoras públicamente^[4]

Analogía simple:

"Libre" es como una receta de comida: podés usarla para cocinar lo que quieras, leer cómo se hace, dársela a tus amigos, y si la mejorás, compartir tu versión mejorada. "Gratis" solo significa que no pagás por la receta.

=====

Archivo 5: A5_DistribucionesdeLinux.pdf

Tema: Distribuciones de Linux

Distribuciones Fundamentales

Explicación técnica:

Las distribuciones GNU/Linux fundamentales son: Debian (enfocada en estabilidad y calidad), Fedora (tecnología de vanguardia e innovación), Ubuntu (facilidad de uso y accesibilidad), CentOS (estabilidad y confiabilidad para servidores), y Red Hat Enterprise Linux (estabilidad para entornos empresariales).^[5]

Analogía simple:

Son como diferentes marcas de autos: Debian es como Toyota (confiable y duradero), Fedora es como Tesla (tecnología nueva), Ubuntu es como Honda (fácil de manejar), CentOS es como un camión de trabajo, y RHEL es como un auto ejecutivo.

La Familia Debian

Explicación técnica:

Debian se caracteriza por su filosofía de software libre, enfoque en estabilidad y calidad del software. Su importancia radica en: estabilidad del sistema, amplia comunidad de desarrolladores, compatibilidad con gran variedad de software, ser base de otras distribuciones populares, provisión de actualizaciones de seguridad regulares, y proceso controlado de actualizaciones.^[5]

Sus principales derivados incluyen Ubuntu, Linux Mint, Kali Linux, Raspbian (para Raspberry Pi) y PureOS.^[5]

Analogía simple:

Debian es como una familia muy organizada y estable. Ubuntu es el hijo que se hizo famoso y fácil de tratar, Mint es el que se volvió elegante, Kali es el experto en seguridad, y Raspbian es el que se especializó en proyectos pequeños.

La Familia Red Hat**Explicación técnica:**

Red Hat Enterprise Linux (RHEL) se enfoca en estabilidad, confiabilidad y escalabilidad para entornos empresariales. Sus derivados incluyen CentOS (clon gratuito de RHEL), Scientific Linux (enfocado en investigación científica), Oracle Linux, Amazon Linux, y ClearOS (enfocado en seguridad de red).^[5]

Explicación del proceso de desarrollo:

Fedora → CentOS Stream → Red Hat Enterprise Linux. CentOS Stream actúa como plataforma de desarrollo upstream donde la comunidad puede participar en el desarrollo del código fuente que luego se convierte en RHEL.^[5]

Analogía simple:

Es como una empresa automotriz: Fedora es el laboratorio donde prueban ideas nuevas, CentOS Stream es donde arman los prototipos, y RHEL es el auto final que venden a las empresas con garantía y soporte.

Ubuntu y la Nube**Explicación técnica:**

Ubuntu se utiliza ampliamente en la nube para: instancias de máquinas virtuales en AWS, Azure y GCP; contenedores como sistema operativo base; servidores de aplicaciones eficientes; y servidores de bases de datos para procesar grandes cantidades de datos.^[5]

La familia Ubuntu incluye: Ubuntu Desktop (sistema de escritorio), Ubuntu Server (optimizado para servidores), Ubuntu Cloud (optimizado para nube), y Ubuntu Core (para IoT).^[5]

Analogía simple:

Ubuntu en la nube es como tener versiones especializadas de tu auto favorito: uno para la ciudad (Desktop), uno para trabajo pesado (Server), uno para servicios de entrega (Cloud), y uno pequeño para repartos (Core/IoT).

Distribuciones Espartanas**Explicación técnica:**

Las distribuciones espartanas (Arch Linux, Gentoo, Slackware) se enfocan en ofrecer una experiencia minimalista y eficiente. No incluyen gran cantidad de aplicaciones preinstaladas, sino que ofrecen un sistema liviano donde los usuarios instalan y configuran por su cuenta. Requieren más conocimientos técnicos pero ofrecen mayor flexibilidad y control.^[5]

Analogía simple:

Son como comprar un auto "kit" que viene desarmado: es más difícil de armar, pero podés personalizarlo exactamente como querés. Solo recomendado para mecánicos expertos.

=====

Archivo 6: A6_GNU_LinuxyLanube.pdf

Tema: GNU/Linux y la nube

¿Qué es Cloud Computing?

Explicación técnica:

El cloud computing se refiere a la ejecución de cargas de trabajo en nubes, que son entornos de TI que extraen, agrupan y comparten recursos flexibles en una red. No son tecnologías en sí mismas:

- **Cloud computing:** es la acción de ejecutar cargas en la nube
- **Nubes:** son los entornos donde se ejecutan las aplicaciones
- **Tecnologías:** son los elementos (software/hardware) para diseñar y usar nubes^[6]

Analogía simple:

Cloud computing es como usar el transporte público: vos no tenés que comprar el colectivo (hardware), solo pagás por el viaje (uso de recursos). La "nube" es toda la infraestructura de transporte, y las tecnologías son los colectivos, rutas y paradas.

Ventajas de Linux en la Nube

Explicación técnica:

Costo de licencias atractivo: Linux es open-source, la mayoría de distribuciones son gratuitas. Incluso las distribuciones pagas (como RHEL) cuestan mucho menos que otros sistemas operativos e incluyen soporte técnico de calidad.^[6]

Variedad especializada: Existen distribuciones especializadas para cada necesidad: Oracle Linux para bases de datos, RHEL para grandes empresas, Kali Linux para ciberseguridad, sistemas IoT minúsculos para dispositivos conectados.^[6]

Analogía simple:

- **Costo:** Es como tener herramientas gratuitas de alta calidad vs. herramientas caras.
- **Variedad:** Como tener el destornillador perfecto para cada tipo de tornillo.

Paradigma Unix

Explicación técnica:

No importa qué distribución uses o si estás en una supercomputadora o dispositivo IoT, la operación de comandos y navegación es muy similar entre distribuciones. Las herramientas de diagnóstico y monitoreo (Dig, NsLookUp, IfConfig, Top, Netstat, VMstat, Free, DF) son las mismas en la mayoría de distribuciones, facilitando la portabilidad de aplicaciones.^[6]

Analogía simple:

Es como saber manejar: sin importar si manejas un auto, camión o moto, los conceptos básicos (volante, frenos, acelerador) son similares. Si sabés Linux, podés adaptarte fácil a cualquier distribución.

Distribuciones Populares en la Nube

Explicación técnica:

Distribución	Características	Uso en Nube
Ubuntu Server	Descendiente de Debian, muy usado desde 2013, GUI amigable en versión desktop ^[6]	Una de las más usadas a todo nivel
OpenSUSE	Excelente estabilidad, interfaz elegante, usa RPM + herramientas YAST/ZYPPER ^[6]	Ganando popularidad por estabilidad
RHEL	Distribuido por Red Hat (ahora IBM), sumamente estable ^[6]	Preferido por empresas para BD y plataformas complejas

Analogía simple:

- **Ubuntu:** Como un auto popular, confiable y fácil de conseguir repuestos.
- **OpenSUSE:** Como un auto alemán, elegante y bien diseñado.
- **RHEL:** Como un auto ejecutivo con servicio premium incluido.

=====

Archivo 7: A7_TiposdeParticiones.pdf

Tema: Tipos de Particiones

Concepto de Particionado

Explicación técnica:

Particionar significa dividir el disco rígido en varias partes, cada una usada para instalar determinadas secciones del sistema. Las particiones deben tener algún tipo de filesystem (sistema de archivos): ext2, ext3, ext4, btrfs, xfs, etc.^[7]

Analogía simple:

Es como dividir tu casa en habitaciones: cada cuarto tiene una función específica (cocina, dormitorio, baño). El particionado hace lo mismo con tu disco: cada "habitación" tiene un propósito específico.

Particiones MBR

Explicación técnica:

Tipo	Descripción	Limitaciones
Primarias	Primera partición creada, usada para arrancar el sistema	Máximo 4 por disco ^[7]
Extendidas	Contenedor para particiones lógicas cuando se necesitan más de 4	Solo 1 por disco, no arrancable ^[7]
Lógicas	Particiones dentro de la extendida	Solo existen dentro de extendida ^[7]

Analogía simple:

- **Primarias:** Como las habitaciones principales de una casa (máximo 4).
- **Extendida:** Como un ala de la casa que podés subdividir.
- **Lógicas:** Como los cuartitos dentro de esa ala.

Particiones GPT

Explicación técnica:

GPT (GUID Partition Table) es un estándar más moderno que reemplaza MBR. Ventajas principales:

1. Usa GUID (UUID) para identificar tipos de particiones
2. GUID único para disco y cada partición
3. Número arbitrario de particiones (hasta 128 por defecto)
4. Soporte para discos de hasta 2 ZiB (vs 2 TB de MBR)
5. Copia de seguridad automática al final del disco
6. Checksum CRC32 para detectar errores ^[7]

Analogía simple:

MBR es como una libreta vieja con pocas hojas y letra pequeña. GPT es como una tablet moderna: más espacio, mejor organización, copia de seguridad automática, y detecta si algo se daña.

=====

Archivo 8: A8_FormatosyNomenclaturas.pdf

Tema: Formatos y Nomenclaturas

Particiones Mínimas y Puntos de Montaje

Explicación técnica:

Los puntos de montaje son directorios que sirven para acceder a una partición determinada. El particionado mínimo consiste en:

- **Partición raíz (/):** Contiene todo el sistema
- **Partición swap:** Memoria virtual de reserva, tamaño recomendado 1x a 2x la RAM ^[8]

Analogía simple:

Los puntos de montaje son como "direcciones postales" para cada parte del disco. La partición raíz (/) es como la dirección principal de tu casa, y swap es como un garaje extra para guardar cosas cuando la casa está llena.

Particiones Especiales

Explicación técnica:

Partición	Función	Tamaño recomendado
/boot	Contiene el kernel, configurar como solo lectura	500 MB ^[8]
/boot/efi	Obligatoria con UEFI, debe ser FAT32/vfat	256 MB ^[8]
/home	Documentos y configuraciones de usuarios	Variable según usuarios ^[8]
/usr	Programas y aplicaciones del sistema	Variable ^[8]
/var	Logs, bases de datos, archivos temporales	Variable ^[8]

Analogía simple:

- **/boot:** Como el motor del auto, fundamental pero no lo tocás seguido.
- **/boot/efi:** Como la llave inteligente de autos modernos.
- **/home:** Como las habitaciones personales de cada familia.
- **/var:** Como un sótano donde se guardan logs y archivos temporales.

Partición SWAP

Explicación técnica:

La partición de intercambio es memoria virtual que el sistema usa cuando la RAM está limitada. En escritorios se usa más que en servidores. Si se detecta uso continuo de SWAP en servidores, es recomendable agregar más RAM. Para hibernación en notebooks, el SWAP debe ser mayor que la RAM. ^[8]

Analogía simple:

SWAP es como el baúl de tu auto: cuando el asiento trasero (RAM) está lleno, guardás cosas en el baúl. Pero es más lento acceder al baúl, entonces preferís tener asientos más grandes (más RAM).

Sistemas de Archivos

Explicación técnica:

Los sistemas de archivos nativos recomendados para Linux son:

- **xfs**: Sistema de archivos moderno y eficiente
- **ext4**: Evolución de ext2/ext3, más recomendado que versiones anteriores
- **btrfs**: Sistema avanzado con funciones como snapshots
- **jfs**: Sistema de archivos de IBM^[8]

Analogía simple:

Son como diferentes tipos de organización para tu biblioteca: algunos son mejores para libros grandes (xfs), otros para muchos libros pequeños (ext4), algunos permiten hacer "fotos" de cómo estaba organizada en cierto momento (btrfs).

Nomenclatura de Discos

Explicación técnica:

En Linux, discos y particiones se identifican como archivos, no letras:

Elemento	Nomenclatura	Ejemplo
Primer disco	/dev/sda	-
Segundo disco	/dev/sdb	-
Primera partición primaria	/dev/sda1	Primer disco, primera partición
Primera partición lógica	/dev/sda5	Primer disco, primera lógica (siempre ≥5)

Analogía simple:

En vez de llamar a los discos "C:", "D:" como Windows, Linux usa nombres como "sda" (primer disco), "sdb" (segundo disco). Las particiones son como "sda1" (habitación 1 del primer piso), "sda5" (primer cuarto del sótano).

=====

Archivo 9: A9_LVM.pdf

Tema: LVM (Logical Volume Manager)

¿Qué es LVM?

Explicación técnica:

LVM (Logical Volume Manager) es un gestor de volúmenes lógicos para Linux que permite asignar espacio de medios de almacenamiento en volúmenes lógicos que pueden ser fácilmente redimensionados. Ofrece flexibilidad, disponibilidad y capacidad de crear instantáneas para backup.^[9]

Analogía simple:

LVM es como tener bloques de LEGO modulares en vez de piezas fijas. Podés armar estructuras, desarmarlas, hacerlas más grandes o chicas según necesites, y hasta hacer "fotos" de cómo estaban armadas en cierto momento.

Ventajas de LVM

Explicación técnica:

- **Flexibilidad:** Redimensionamiento fácil de volúmenes
- **Disponibilidad:** Agregar discos y migrar datos sin parar el sistema
- **Instantáneas (Snapshots):** Crear copias de respaldo sin detener servicios^[9]

Analogía simple:

- **Flexibilidad:** Como poder agrandar o achicar las habitaciones de tu casa sin demoler.
- **Disponibilidad:** Como poder agregar un anexo mientras seguís viviendo en la casa.
- **Snapshots:** Como tomar una foto de cómo está tu casa para poder volver a ese estado después.

Componentes de LVM

Explicación técnica:

- **PV (Physical Volume):** Una partición física
- **VG (Volume Group):** Uno o más PV agrupados como si fuera un único disco
- **LV (Logical Volume):** Subdivisiones del VG, similares a particiones comunes^[9]

Un VG puede tener PVs de diferentes discos físicos. Solo los LV se formatean, no los PV ni VG.^[9]

Analogía simple:

- **PV:** Como ladrillos individuales.
- **VG:** Como una pared hecha con esos ladrillos.
- **LV:** Como dividir esa pared en secciones para diferentes usos.

Comandos Básicos de LVM

Comando	Función	Caso de uso
<code>pvs</code>	Mostrar volúmenes físicos	Ver qué discos/particiones están en LVM ^[9]
<code>vgs</code>	Mostrar grupos de volúmenes	Ver información de grupos creados ^[9]
<code>lvs</code>	Mostrar volúmenes lógicos	Ver las "particiones" lógicas activas ^[9]

Analogía simple:

- `pvs`: Ver qué ladrillos tenés disponibles.

- vgs: Ver qué paredes armaste con esos ladrillos.
- lvs: Ver cómo dividiste cada pared para usar.

=====

Archivo 10: B1_ComotrabajarenLinux.pdf

Tema: Cómo trabajar en Linux

Inicio de Sesión

Explicación técnica:

Una vez instalado Linux, es necesario loguearse (iniciar sesión de usuario). Existen dos maneras principales: usar una terminal virtual de texto o usar una terminal gráfica.^[10]

Analogía simple:

Es como entrar a un edificio: podés usar las escaleras (terminal de texto) o el ascensor (terminal gráfica). Ambos te llevan adonde querés ir, pero la experiencia es diferente.

Terminales Virtuales

Explicación técnica:

Una terminal virtual es una pantalla virtual. Linux tiene generalmente seis terminales de texto y una gráfica. Para cambiar entre terminales de texto: Alt+Fn (donde n es el número). Para pasar de terminal gráfica a texto: Ctrl+Alt+Fn. En VirtualBox: Ctrl-derecho+Fn.^[10]

Analogía simple:

Es como tener varios escritorios virtuales o canales de TV: podés cambiar entre ellos con el control remoto (atajos de teclado). Cada "canal" es independiente y podés trabajar en cosas diferentes en cada uno.

Terminal Gráfica

Explicación técnica:

En el entorno gráfico se pueden abrir terminales virtuales usando programas específicos. En GNOME se accede con Alt+F2, luego escribir "gnome-terminal" y presionar Enter.^[10]

Analogía simple:

Es como abrir una ventana dentro de tu escritorio gráfico donde podés escribir comandos. Como tener un "chat" directo con la computadora dentro de tu entorno visual.

=====

Archivo 11: B2_Comandos.pdf

Tema: Comandos básicos

Identificación de Usuario

Explicación técnica:

El prompt indica el tipo de usuario: # para root (administrador), \$ para usuario común. Para confirmar se usa el comando `whoami`. Si aparece "root", tenés privilegios completos del sistema. ^[11]

Analogía simple:

Es como las credenciales en un hospital: # es como ser el director (podés hacer todo), \$ es como ser visitante (acceso limitado). `whoami` es como mirar tu gafete para recordar quién sos.

Comandos de Sistema

Comando	Función	Caso de uso
<code>whoami</code>	Mostrar usuario actual	Verificar con qué usuario estás trabajando ^[11]
<code>date</code>	Mostrar fecha y hora actual	Ver información temporal del sistema ^[11]
<code>poweroff</code>	Apagar el sistema	Cierre seguro del sistema ^[11]
<code>su -</code>	Cambiar a usuario root	Obtener privilegios de administrador ^[11]

Comandos de Navegación

Comando	Función	Caso de uso
<code>dir</code>	Mostrar contenido de directorios	Listar archivos y carpetas ^[11]
<code>pwd</code>	Mostrar directorio actual	Saber dónde estás ubicado ^[11]
<code>less</code>	Visor de archivos de texto	Leer archivos largos (salir con 'q') ^[11]
<code>more</code>	Visor de texto básico	Leer archivos (más antiguo que less) ^[11]

Comandos Utilitarios

Comando	Función	Caso de uso
<code>cal</code>	Mostrar calendario	Ver fechas (no disponible en todas distribuciones) ^[11]
<code>exit</code>	Cerrar terminal	Salir de la terminal actual ^[11]
<code>logout</code>	Cerrar sesión	Cerrar terminal con sesión activa ^[11]
<code>tty</code>	Mostrar terminal actual	Ver qué terminal virtual estás usando ^[11]
<code>users</code>	Mostrar usuarios conectados	Ver quién está logueado en el sistema ^[11]

Comando	Función	Caso de uso
<code>factor</code>	Factorizar números	Operaciones matemáticas básicas ^[11]

Analogía simple:

- `pwd`: Como preguntar "¿dónde estoy?" en un edificio.
- `dir`: Como ver qué hay en una habitación.
- `less`: Como hojear un libro largo.
- `exit`: Como salir de una habitación.

=====

Archivo 12: B3_Masrecursos.pdf

Tema: Más recursos

Recursos Adicionales

Explicación técnica:

El archivo proporciona enlaces a recursos complementarios para profundizar el aprendizaje:

- Material recomendado por LPI para LPIC1 (certificación)
- Documentación de instalación en Gentoo Wiki
- Guías específicas de instalación de Debian
- Recursos sobre planificación de instalación de Linux ^[12]

Analogía simple:

Son como las "fuentes recomendadas" al final de un libro de texto: si querés saber más o necesitás ayuda específica, estos recursos te van a guiar paso a paso.

=====

Mini-examen: 5 preguntas

1. ¿Cuáles son los tres componentes principales de cualquier distribución GNU/Linux?
2. ¿Qué diferencia hay entre software "libre" y software "gratis"?
3. ¿Cuál es la diferencia principal entre particiones MBR y GPT?
4. ¿Qué significan las siglas PV, VG y LV en LVM?
5. ¿Cómo sabés si estás trabajando como usuario root o como usuario común en Linux?

=====

Práctica en máquina virtual

Ejercicio Integral: Exploración del Sistema

Instrucciones:

1. Abrió una terminal en tu máquina virtual Linux
2. Ejecutá los siguientes comandos en orden:

```
whoami  
pwd  
date  
tty
```

3. Intentá cambiar entre terminales virtuales usando Alt+F2, Alt+F3, Alt+F1
4. Si tenés permisos, ejecutá: `su -` (te va a pedir la contraseña de root)
5. Después de hacer `su -`, ejecutá `whoami` nuevamente
6. Ejecutá `exit` para volver a tu usuario normal
7. Finalmente ejecutá `users` para ver usuarios conectados

Resultado esperado:

- `whoami`: debería mostrar tu nombre de usuario (ej: "hector")
- `pwd`: debería mostrar algo como `"/home/hector"` o `"/home/tuusuario"`
- `date`: debería mostrar la fecha y hora actual del sistema
- `tty`: debería mostrar algo como `"/dev/tty1"` o `"/dev/pts/0"`
- Después de `su -` y volver a ejecutar `whoami`: debería mostrar "root"
- `users`: debería mostrar tu nombre de usuario (y "root" si todavía estás con `su -`)

Si todo funciona correctamente, significa que:

- ✓ Tu sistema Linux está funcionando
- ✓ Podés navegar entre usuarios
- ✓ Entendés los conceptos básicos de identificación en Linux
- ✓ Estás listo para avanzar al siguiente módulo

¡Héctor, con esto tenés una base sólida para empezar tu aventura en Linux! Estos archivos te dan todo lo fundamental para entender cómo funciona este mundo. Ahora a practicar y a no tener miedo de experimentar en tu máquina virtual. 🚀



1. A1_Conceptos_importantes.pdf
2. A2_Que_es_Unix.pdf
3. A3_Manerasde_InstalarLinux.pdf
4. A4_OrigenesdeLinux.pdf

5. A5_DistribucionesdeLinux.pdf
6. A6_GNU_LinuxyLanube.pdf
7. A7_TiposdeParticiones.pdf
8. A8_FormatosyNomenclaturas.pdf
9. A9_LVM.pdf
10. B1_ComotrabajarenLinux.pdf
11. B2_Comandos.pdf
12. B3_Masrecursos.pdf