

1) Características de GNU/Linux:

a) Mencione y explique las características más relevantes de GNU/Linux.

- Es un sistema operativo Unix-like
- Gratuito y de libre distribución
- Es código abierto
- Rápida corrección ante fallas

Además, GNU se refiere a 4 libertades principales:

- Libertad de usar el programa con cualquier propósito
- Libertad de estudiar su funcionamiento
- Libertad para distribuir sus copias
- Libertad para mejorar los programas

b) Mencione otros sistemas operativos y compárelos con GNU/Linux en cuanto a los puntos mencionados en el inciso a).

Windows y Mac Os son sistemas operativos de propietario, es decir, una empresa administra el código y su distribución.

Estos son de paga y no poseen código abierto, por lo tanto no es modificable, estudiable, etc.

Toda corrección de fallas queda en manos del propietario, por lo tanto hay menor necesidad de técnicos especializados.

No todos los sistemas operativos de propietario son de pago, usualmente lo son. Misma comparativa con el software libre, no quiere decir que sea gratuito.

c) ¿Qué es GNU?

Las siglas de GNU significan “GNU’s not Unix” (GNU no es Unix).

Es una colección de paquetes de software que pueden ser utilizados como sistema operativo o adaptarse a otro sistema operativo.

La mayoría del software de GNU está licenciado bajo su propia GPL (General Public License), lo que permite que sea estudiado, distribuido y modificado gratuita y libremente.

d) Indique una breve historia sobre la evolución del proyecto GNU.

Robado de <https://github.com/agusrnfr>

Fue iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre (el sistema GNU). En 1985 Richard Stallman creó la Free Software Foundation (FSF o Fundación para el Software Libre) para proveer soportes logísticos, legales y financieros al proyecto GNU. Esta fundación contrato programadores,

aunque una porción sustancial del desarrollo fue (y continúa siendo) producida por voluntarios.

En 1990, GNU ya contaba con un editor de textos (Emacs), un compilador (GCC) y gran cantidad de bibliotecas que componen un Unix típico pero faltaba el componente principal, el núcleo (Kernel).

Linus Torvalds ya venía trabajando desde 1991 en un Kernel denominado Linux, el cual se distribuía bajo licencia GPL. Múltiples programadores se unieron a Linus en el desarrollo, colaborando a través de Internet y consiguiendo paulatinamente que Linux llegase a ser un núcleo compatible con UNIX. En 1992, el núcleo Linux fue combinado con el sistema GNU, resultando en un SO libre y completamente funcional. El SO formado por esta combinación es usualmente conocido como "GNU/Linux" o como una "distribución Linux" y existen diversas variantes.

e) Explique qué es la multitarea, e indique si GNU/Linux hace uso de ella.

GNU/Linux es multiusuario, multitarea y multiprocesador:

Multitarea: Es la capacidad de un sistema operativo para ejecutar múltiples tareas o procesos al mismo tiempo. El CPU generalmente ejecuta solo una instrucción de una tarea a la vez, pero los sistemas operativos modernos pueden alternar entre tareas rápidamente para parece que se realizan en simultáneo.

Esto se logra mediante la planificación de procesos:

- Preemptive Multitasking: El sistema operativo decide cuándo interrumpir una tarea para dar tiempo de CPU a otra tarea, mejorando la capacidad de respuesta del sistema.
- Cooperative Multitasking: Las tareas ceden voluntariamente el control de la CPU a otras tareas.

Multiusuario: Capacidad de un sistema operativo para permitir que varios usuarios utilicen la misma computadora o recursos de la red al mismo tiempo o en diferentes momentos. (Ejemplo: un servidor).

Cada usuario tiene su propio espacio de trabajo, archivos y permisos. Los sistemas operativos que soportan multiusuario gestionan la seguridad y los permisos para garantizar que un usuario no pueda interferir con el trabajo de otro.

Multiprocesador: Se refiere a un sistema con más de un procesador (CPU) que puede ejecutar múltiples procesos simultáneamente.

Los sistemas multiprocesador pueden distribuir tareas entre los diferentes procesadores, mejorando significativamente el rendimiento, especialmente para tareas intensivas en procesamiento. El sistema operativo debe soportar el procesamiento paralelo:

- SMP (Symmetric Multiprocessing): Todos los procesadores comparten un mismo espacio de memoria y son iguales en términos de acceso a los recursos.
- AMP (Asymmetric Multiprocessing): Un procesador principal controla el sistema y asigna tareas a los procesadores secundarios.

f) ¿Qué es el POSIX?

Robado de <https://github.com/agusrnfr>

POSIX consiste en una familia de estándares especificadas por la IEEE con el objetivo de facilitar la interoperabilidad de sistemas operativos. Además, POSIX establece las reglas para la portabilidad de programas. Por ejemplo, cuando se desarrolla software que cumple con los estándares POSIX existe una gran probabilidad de que se podrá utilizar en sistemas operativos del tipo Unix. Si se ignoran tales reglas, es muy posible que el programa o librería funcione bien en un sistema dado pero que no lo haga en otro.

2) Distribuciones de GNU/Linux:

a) ¿Qué es una distribución de GNU/Linux? Nombre al menos 5 distribuciones de GNU/Linux y cite diferencias básicas entre ellas

Una distribución es una customización de GNU/Linux formada por una versión de Kernel y un conjunto de herramientas y software adicionales, configurados de manera específica para cumplir con diferentes propósitos o satisfacer diferentes necesidades de los usuarios.

Cada distribución tiene características propias y es tarea del usuario elegir una acorde a sus necesidades y gustos personales.

Robado de <https://github.com/agusrnfr>

Distribuciones como **Ubuntu** se centran en ser lo más amigables posible a la hora de instalarse o descargar programas. **Linux Mint** aprovecha el hardware potente para competir con Windows o MacOS. Para computadoras viejas hay distribuciones ligeras como **Puppy Linux**. Para Linux en servidores, **Debian**, y para jugar videojuegos, **SteamOS** es la mejor.

Hay distribuciones que desarrollan compañías comerciales, como **Fedora**, **Mandriva** o la propia Ubuntu, y otras mantenidas por la comunidad Linux, como Debian, que no está relacionada con ninguna empresa y utiliza únicamente software.

b) ¿En qué se diferencia una distribución de otra?

Las distribuciones de Linux varían en varios aspectos, lo que les permite satisfacer diferentes necesidades, preferencias y niveles de experiencia.

- Gestor de paquetes: Cada distribución utiliza un gestor de paquetes específico para instalar, actualizar y gestionar Software:

Debian, Ubuntu, Linux Mint utilizan 'APT' (Advanced Package Tool) con paquetes '.deb'.

Fedora, CentOS, RHEL utilizan 'DNF' (anteriormente 'YUM') con paquetes '.rpm'.

Arch Linux utiliza 'Pacman' con paquetes '.pkg.tar.zst'.

openSuse utiliza 'Zypper' con paquetes '.rpm'.

- Filosofía y propósito: Las distribuciones tienen diferentes enfoques y objetivos:

Debian enfatiza la estabilidad y software libre.

Ubuntu deriva de Debian, se enfoca en la facilidad de uso.

Fedora patrocinada por Red Hat, se centra en incorporar las últimas tecnologías.

Arch Linux ofrece un enfoque minimalista, diseñado para usuario avanzados que prefieren construir su sistema desde cero.

CentOS/RHEL orientadas a servidores y empresas, se centran en la estabilidad a largo plazo y soporte comercial.

- Entornos de Escritorio: Aunque el entorno se puede customizar en casi cualquier distribución, algunas prefieren ciertos entornos:

Ubuntu viene con GNOME por defecto.

Linux Mint ofrece Cinnamon, MATE y Xfce como entornos principales.

Fedora por defecto incluye GNOME, pero ofrece ediciones con entornos como KDE, Plasma y Xfce.

- Soporte y Ciclo de Vida: El tiempo de soporte y la frecuencia de actualizaciones varía:

Ubuntu LTS (Long Term Support) ofrece 5 años de soporte en sus versiones LTS.

Fedora tiene un ciclo de vida de ~13 meses, con versiones nuevas lanzadas cada 6 meses.

Arch Linux es una distribución "rolling release", lo que significa que se actualiza continuamente en lugar de tener versiones periódicas.

Debian Stable tiene un ciclo de lanzamiento de aproximadamente 2 años, priorizando la estabilidad.

- Facilidad de Uso: Las distribuciones varían en su complejidad y la curva de aprendizaje necesaria:

Ubuntu y Linux Mint están diseñadas para ser amigables para principiantes, con instaladores gráficos y configuraciones predeterminadas fáciles de usar.

Debian es robusta y estable, pero su instalación puede ser más compleja.

Arch Linux requiere conocimientos avanzados debido a que la instalación y configuración inicial son completamente manuales.

Fedora es un punto intermedio.

Entre muchas otras diferencias.

- c) ¿Qué es Debian? Acceda al sitio e indique cuáles son los objetivos del proyecto y una breve cronología del mismo.

Robado de <https://github.com/agusrnfr>

El Proyecto Debian es una asociación de personas que han hecho causa común para crear un sistema operativo libre.

Es producido por alrededor de un millar de desarrolladores activos, dispersos por el mundo que ayudan voluntariamente en su tiempo libre. La comunicación se realiza principalmente mediante correo electrónico (listas de correo en lists.debian.org) y de IRC (canal [#debian](irc://irc.debian.org) [irc.debian.org](irc://irc.debian.org)).

Comenzó en agosto de 1993 gracias a Ian Murdock, como una nueva distribución que se realizaría de forma abierta, en la línea del espíritu de Linux y GNU. Debian estaba pensado para ser creada de forma cuidadosa y concienzuda, y ser mantenida y soportada con el mismo cuidado. Comenzó como un grupo de pocos y fuertemente unidos hackers de Software Libre, y gradualmente creció hasta convertirse en una comunidad grande y bien organizada de desarrolladores y usuarios.

3) Estructura de GNU/Linux:

- a) Nombres cuales son los 3 componentes fundamentales de GNU/Linux.

- Kernel (Núcleo)
- Shell (Intérprete de comandos)
- FileSystem (Sistema de archivos)

- b) Mencione y explique la estructura básica del Sistema Operativo GNU/Linux.

El Kernel ejecuta programas y gestiona dispositivos de Hardware, es el encargado de que el Software y el Hardware puedan trabajar juntos. Sus funciones más importantes son la administración de memoria, CPU y E/S.

En si, y en un sentido estricto, es el sistema operativo. Es un núcleo monolítico híbrido, los drivers y código del Kernel se ejecutan en modo privilegiado, lo que lo hace híbrido es la capacidad de cargar y descargar funcionalidad a través de módulos.

La Shell o también conocida como CLI (Command Line Interface) es el modo de comunicación entre el usuario y el sistema operativo.

El FileSystem organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento (fat, ntfs, ext2, ext3, reiser, etc).

El adoptado por GNU/Linux es el Extended (ext2, ext3, etc).

Los directorios más importantes según el FHS (FileSystem Hierarchy Standard) son:

- / Tope de la estructura de directorios (Similar al C:\ en Windows)
- /home Se almacenan archivos de usuarios (Mis Documentos)
- /var Información que varía de tamaño (Logs, DB, spools)
- /etc Archivos de configuración
- /bin Archivos binarios y ejecutables
- /dev Enlace a dispositivos
- /usr Aplicaciones de usuarios

4) Kernel:

- a) ¿Qué es? Indique una breve reseña histórica acerca de la evolución del Kernel de GNU/Linux.

Robado de <https://github.com/agusrnfr>

El kernel (también conocido como núcleo) es la parte fundamental de un sistema operativo. El kernel o núcleo de linux se podría definir como el corazón de este sistema operativo. Es, a grandes rasgos, el encargado de que el software y el hardware de una computadora puedan trabajar juntos.

Historia:

- En 1991 Linus Torvalds inicia la programación de un Kernel Linux basado en Minix.
- El 5 de octubre de 1991, se anuncia la primera versión "oficial" de Linux (0.02)
- En 1992 se combina su desarrollo con GNU, formando GNU/Linux
- La versión 1.0 apareció el 14 de marzo de 1994
- En mayo de 1996 se decide adoptar a Tux como mascota oficial de Linux
- En julio de 1996 se lanza la versión 2.0 y se define la nomenclatura de versionado¹. Se desarrollo hasta febrero de 2004 y termino con la 2.0.40
- En enero de 1999 se lanza la versión 2.2, que provee mejoras de portabilidad entre otras y se desarrolla hasta febrero de 2004 terminando en la versión 2.2.26
- En 2001 se lanza la versión 2.4 y se deja de desarrollar a fines del 2010 con la 2.4.37.11
- La versión 2.4 fue la que catapulto a GNU/Linux como un SO estable y robusto. Durante este periodo es que se comienza a utilizar Linux más asiduamente
- A fines del 2003 se lanza la versión 2.6. Esta versión ha tenido muchas mejoras para el SO dentro de las que se destacan soporte de hilos, mejoras en la planificación y soporte de nuevo hardware
- El 3 de agosto de 2011 se lanza la versión 2.6.39.4 anunciándose la misma desde meses previos como la última en su revisión
- El 17 de julio de 2011 se lanza la versión 3.01 por los 20 años del SO. Es totalmente compatible con 2.6
- La última versión estable es la 4.7.1 (agosto de 2016)

b) ¿Cuáles son sus funciones principales?

Sus funciones más importantes son la administración de memoria, CPU y E/S.

c) ¿Cuál es la versión actual? ¿Cómo se definía el esquema de versionado de Kernel en versiones anteriores a la 2.4? ¿Qué cambió en el versionado se impuso a partir de la versión 2.6?

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release
6.10.6 

mainline:	6.11-rc4	2024-08-18	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]	
stable:	6.10.6	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
stable:	6.9.12 [EOL]	2024-07-27	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	6.6.47	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	6.1.106	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	5.15.165	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	5.10.224	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	5.4.282	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
longterm:	4.19.320	2024-08-19	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse] [changelog]
linux-next:	next-20240823	2024-08-23						[browse]

Última versión disponible 6.10.6 (24 de agosto 2024)

Antes de la serie de Linux 2.6.x, los números pares indicaban la versión “estable” lanzada. Por ejemplo, una para uso de fabricación, como el 1.2, 2.4 ó 2.6. Los números impares, en cambio, como la serie 2.5.x, son versiones de desarrollo, es decir que no son consideradas de producción.

Comenzando con la serie Linux 2.6.x, no hay gran diferencia entre los números pares o impares con respecto a las nuevas herramientas desarrolladas en la misma serie del kernel. Linus Torvalds dictaminó que este será el modelo en el futuro.

- d) ¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?

Sí, es posible tener más de un Kernel instalado a la vez, pero esto no significa que se puedan usar en simultáneo.

Muchas distribuciones de Linux permiten tener múltiples versiones del Kernel instaladas al mismo tiempo, esto se gestiona generalmente a través del gestor de arranque (Bootloader) como GRUB (**GR**and **U**nified **B**ootloader también conocido como GNU GRUB).

También se puede lograr mediante la utilización de máquinas virtuales.

- e) ¿Dónde se encuentra ubicado dentro del FileSystem?

La finalidad del Bootloader es la de cargar una imagen de Kernel (SO) de alguna partición para su ejecución. Se ejecuta luego del código de la BIOS.

Existen 2 modos de instalación:

- En el MBR (puede llegar a utilizar MBR Gap)
- En el sector de arranque de la partición o raíz activa (Volume Boot Record)

El MBR es un sector reservado del disco físico (Cilindro 0, Cabeza 0, Sector 1).

En el MBR solo se encuentra la fase 1 del que solo se encarga de cargar la fase 1.5.

La fase 1.5 se ubica en los siguientes 30KB del disco (MBR Gap) y carga la fase 2.

La fase 2 es la interfaz de usuario y carga el Kernel seleccionado.

Se configura a través del archivo /boot.

- f) ¿El Kernel de GNU/Linux es monolítico? Justifique.

El Kernel es monolítico (un solo proceso) híbrido: Los drivers y código del Kernel

se ejecutan en modo privilegiado con acceso irrestricto al Hardware, algunos se ejecutan en espacio de usuario. Lo que lo hace híbrido es la capacidad de cargar y descargar funcionalidad a través de módulos.

5) Intérprete de comandos (Shell):

a) ¿Qué es?

Es la forma que tiene el usuario de comunicarse con el Sistema Operativo. Recibe comandos que transforma en instrucciones de bajo nivel para poder ejecutar.

b) ¿Cuáles son sus funciones?

Permite al usuario comunicarse mediante comandos de texto con el sistema operativo, el intérprete es el encargado de transformar los comandos en algo que el Sistema Operativo puede ejecutar.

c) Menciona al menos 3 intérpretes de comandos que posee GNU/Linux y compárelos entre ellos:

Bourne Shell (sh), Korn Shell (ksh), Bourne Again Shell (bash)

- Bourne Shell: Sus características incluyen la **simplicidad** debido a su enfoque en la ejecución de comandos simples y scripts, **portabilidad** gracias a que 'sh' está disponible en casi todos los sistemas Unix, su **sintaxis básica**, que soporta operaciones básicas de control de flujo como 'if', 'for', 'while', pero carece de características avanzadas como la edición en la línea de comandos y permite la **utilización de variables** de enterno, además de operaciones simples con ellas.
- Korn Shell: Cuenta con **compatibilidad** con 'sh', por lo tanto todos los scripts creados en 'ksh' pueden ser ejecutados en Bourne Shell sin problema, además posee **funciones avanzadas** como la edición de comandos (similar a 'vi' y 'emacs'), alias y funciones de autocompletado, permite la utilización de **arrays**, facilitando la manipulación de conjunto de datos, una **programación robusta** y es conocido por su buen **rendimiento**.
- Bourne Again Shell: Fue desarrollado con la idea de ser el reemplazo libre de 'sh', pero con más características. **Compatibilidad** total con 'sh' y en gran parte con 'ksh', **interactividad** para la edición en la línea de comandos, autocompletado, historial de comandos, alias y scripts más complejos. También cuenta con **funciones**

avanzadas como programación estructurada con funciones, arrays y operaciones aritméticas avanzadas, **scripting poderoso** gracias a las mejoras para la redirección de E/S, procesamiento de señales y manipulación de variables y una gran **popularidad**, siendo el shell predeterminado en la mayoría de distribuciones de Linux, haciéndolo uno de los más usados.

Característica	Bourne Shell (sh)	Korn Shell (ksh)	Bourne Again Shell (bash)
Año de creación	1979	1980s	1989
Compatibilidad con sh	N/A	Completa	Completa
Edición de línea de comandos	No	Sí, con vi/emacs	Sí, con vi/emacs
Autocompletado	No	Sí	Sí
Historial de comandos	No	Sí	Sí
Soporte para arrays	No	Sí	Sí
Alias de comandos	No	Sí	Sí
Popularidad	Alta en sistemas Unix	Alta en sistemas empresariales	Altísima en Linux

d) ¿Dónde se ubican (path) los comandos propios y externos al Shell?

Los comandos externos al Shell se ubican en /usr/bin, directorio donde están los archivos binarios y ejecutables.

```
leo@leo:~/Documents/GitHub/Facultad/Segundo/Segundo semestre/ISO/Prácticas$ type mkdir
mkdir is /usr/bin/mkdir
```

Los comandos propios no se encuentran en ningún directorio, están integrados directamente en el propio intérprete Shell.

```
leo@leo:~/Documents/GitHub/Facultad/Segundo/Segundo semestre/ISO/Prácticas$ type cd
cd is a shell builtin
```

e) ¿Por qué considera que el Shell no es parte del Kernel de GNU/Linux?

El Shell no es parte del Kernel, es la forma que tenemos de comunicarnos con tal. Cada uno puede utilizar una versión distinta, propia o customizada de Shell debido a que no es algo integrado al Sistema Operativo.

f) ¿Es posible definir un intérprete de comandos distinto para cada usuario?
¿Desde dónde se define? ¿Cualquier usuario puede realizar dicha tarea?

El shell predeterminado de un usuario es el programa que se inicia cuando ese usuario inicia sesión en el sistema.

Los Shells más comunes en Linux incluyen:

- Bash 'bin/bash'
- Zsh 'bin/zsh'
- Ksh 'bin/ksh'

Entre otros.

Con el comando 'getent passwd usuario' se puede identificar el Shell predeterminado para el usuario.

Utilizando 'sudo chsh -s /ruta/nuevoShell usuario' se puede cambiar el Shell predeterminado.

```
leo@leo:~/Documents/GitHub/Facultad/Segundo/Segundo semestre/ISO/Prácticas$ getent passwd leo
leo:x:1000:1000:Leo:/home/leo:/bin/bash
```

Para ver el listado de los Shells disponibles se puede utilizar:

```
leo@leo:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/usr/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/usr/bin/dash
```