**1. Características de GNU/Linux:**

* **(a) Características más relevantes de GNU/Linux:**
  1. **Código abierto:** Su código es libremente accesible y modificable.
  2. **Multitarea:** Permite ejecutar varios procesos a la vez.
  3. **Multiusuario:** Varios usuarios pueden usar el sistema simultáneamente.
  4. **Portabilidad:** Es compatible con diversas plataformas de hardware.
  5. **Seguridad:** Es robusto frente a ataques debido a su diseño modular.
  6. **Eficiencia en la gestión de recursos:** GNU/Linux es conocido por gestionar los recursos de hardware de manera eficiente.
* **(b) Comparación con otros sistemas operativos:**
  1. **Windows:** Es cerrado y menos flexible, aunque más extendido en el mercado. GNU/Linux es más personalizable y seguro.
  2. **macOS:** Basado en UNIX, como GNU/Linux, pero es cerrado y más limitado en hardware. GNU/Linux es más flexible.
  3. **BSD:** Ambos son de código abierto, pero BSD es más estricto en cuanto a licencias, mientras que GNU/Linux es más popular.
* **(c) ¿Qué es GNU?** GNU es un sistema operativo compuesto únicamente por software libre, lanzado bajo la licencia GPL. El acrónimo significa "GNU's Not Unix", en referencia a que es un sistema operativo similar a Unix, pero sin código propietario.
* **(d) Historia del proyecto GNU:** GNU fue iniciado por Richard Stallman en 1983 para crear un sistema operativo libre. El proyecto avanzó rápidamente con la creación de herramientas como el compilador GCC y editores como Emacs, pero necesitaba un núcleo operativo o **kernel** (Linux).
* **(e) Multitarea en GNU/Linux:** La multitarea permite ejecutar múltiples procesos simultáneamente. GNU/Linux implementa plenamente la multitarea, permitiendo que varios programas corran de manera eficiente al mismo tiempo.
* **(f) ¿Qué es POSIX?** POSIX (Portable Operating System Interface) es un conjunto de estándares que definen la compatibilidad entre sistemas operativos. GNU/Linux es compatible con POSIX, lo que asegura que las aplicaciones escritas para otros sistemas compatibles con POSIX puedan ejecutarse en GNU/Linux.

**2. Distribuciones de GNU/Linux:**

* **(a) ¿Qué es una distribución de GNU/Linux?** Una distribución es una versión de GNU/Linux que incluye el kernel de Linux junto con un conjunto de herramientas, aplicaciones y un gestor de paquetes. Ejemplos:
  1. **Debian:** Estable y confiable, orientada a servidores y usuarios avanzados.
  2. **Ubuntu:** Fácil de usar, basada en Debian, orientada a usuarios principiantes.
  3. **Fedora:** Innovadora, utilizada para probar las últimas tecnologías.
  4. **Arch Linux:** Minimalista y altamente personalizable.
* **(b) ¿En qué se diferencia una distribución de otra?** Se diferencian en el gestor de paquetes (APT, RPM, Pacman), el enfoque (usuarios novatos, avanzados, servidores) y el soporte comunitario.
* **(c) ¿Qué es Debian?** Debian es una distribución de GNU/Linux conocida por su estabilidad y su enfoque en el software libre. Su cronología comenzó en 1993. Los objetivos del proyecto incluyen la creación de un sistema operativo libre y accesible para todos​(Practica 1).

**3. Estructura de GNU/Linux:**

* **(a) Tres componentes fundamentales:**
  1. **Kernel:** El núcleo del sistema operativo que interactúa con el hardware.
  2. **Shell:** La interfaz que permite a los usuarios interactuar con el sistema.
  3. **Sistema de archivos:** Estructura para almacenar y gestionar archivos.
* **(b) Estructura básica del sistema operativo:** El sistema se organiza en capas. El kernel gestiona el hardware, el sistema de archivos organiza los datos, y las aplicaciones interactúan a través del shell o entornos gráficos.

**4. Kernel:**

* **(a) ¿Qué es?** El kernel es el núcleo de GNU/Linux que controla todas las interacciones entre el hardware y el software. Proporciona servicios esenciales como la gestión de memoria, procesos, y dispositivos.
* **(b) Funciones principales del kernel:**
  1. **Gestión de memoria.**
  2. **Control de procesos.**
  3. **Gestión de dispositivos de hardware.**
  4. **Seguridad y control de acceso.**
* **(c) Versión actual del kernel:** La versión actual del kernel se puede obtener con el comando uname -r. A partir de la versión 2.6, el esquema de versionado se modificó para hacer lanzamientos más frecuentes.
* **(d) ¿Es posible tener más de un kernel?** Sí, es posible tener varios kernels instalados y elegir cuál usar durante el arranque.
* **(e) Ubicación del kernel en el sistema de archivos:** El kernel está ubicado en el directorio **/boot**, usualmente con nombres como **vmlinuz**.
* **(f) ¿Es monolítico el kernel de GNU/Linux?** Sí, el kernel de Linux es monolítico, lo que significa que todas las funcionalidades esenciales se ejecutan en un solo espacio de memoria. Sin embargo, es modular, permitiendo agregar o quitar funcionalidades.

**5. Intérprete de comandos (Shell):**

* **(a) ¿Qué es?** El shell es un programa que permite a los usuarios interactuar con el sistema operativo mediante la introducción de comandos.
* **(b) Funciones del Shell:**
  1. Ejecutar comandos.
  2. Crear scripts para automatizar tareas.
  3. Gestionar y supervisar procesos.
* **(c) Tres intérpretes de comandos y comparación:**
  1. **Bash:** El más común y versátil.
  2. **Zsh:** Más avanzado, con funciones adicionales como autocompletado.
  3. **Fish:** Fácil de usar, con características modernas como sugerencias instantáneas.
* **(d) Ubicación de los comandos:** Los comandos internos del shell se encuentran en **/bin** y **/usr/bin**, mientras que los externos suelen estar en **/usr/local/bin**.
* **(e) ¿Por qué el shell no es parte del kernel?** El shell opera en el espacio de usuario, mientras que el kernel opera en el espacio de núcleo. El shell es solo una interfaz que interactúa con el kernel, pero no forma parte de él.
* **(f) ¿Es posible definir un shell distinto para cada usuario?** Sí, cada usuario puede tener un shell distinto. Esto se define en el archivo **/etc/passwd**.

**6. Sistema de Archivos (File System):**

* **(a) ¿Qué es?** Un sistema de archivos es una estructura que organiza cómo se almacenan y recuperan los datos en un dispositivo de almacenamiento.
* **(b) Sistemas de archivos soportados:** GNU/Linux soporta varios sistemas de archivos, incluidos **ext4**, **FAT32**, **NTFS** (con software adicional).
* **(c) ¿Es posible visualizar particiones FAT y NTFS?** Sí, GNU/Linux puede montar y acceder a particiones FAT y NTFS mediante herramientas como **ntfs-3g**.
* **(d) Estructura básica de los File System:** Los directorios más importantes son:
  + **/**: Raíz del sistema.
  + **/home**: Directorios de los usuarios.
  + **/bin**: Binarios esenciales.
  + **/etc**: Archivos de configuración. El FHS (Filesystem Hierarchy Standard) define esta estructura.

**7. Particiones:**

* **(a) Definición. Tipos de particiones. Ventajas y Desventajas:**
  + **Definición:** Las particiones son divisiones lógicas del espacio en un disco duro que permiten separar diferentes sistemas de archivos.
  + **Tipos de particiones:**
    1. **Primarias:** Es donde se puede instalar un sistema operativo. Solo pueden haber 4 particiones primarias o 3 primarias y una extendida.
    2. **Extendidas:** Pueden contener varias particiones lógicas.
    3. **Lógicas:** Particiones dentro de una extendida, sin límite práctico.
  + **Ventajas:**
    1. Separación de datos y sistemas operativos.
    2. Mayor control sobre el uso del espacio en disco.
  + **Desventajas:**
    1. Si no se planifica bien, puede haber espacio no utilizado.
    2. Complejidad en la gestión.
* **(b) ¿Cómo se identifican las particiones en GNU/Linux?**
  + En discos **IDE**, las particiones se identifican como **/dev/hda1**, **/dev/hda2**.
  + En discos **SCSI** y **SATA**, las particiones se identifican como **/dev/sda1**, **/dev/sda2**, etc.
  + La letra (a, b, c) indica el disco, y el número indica la partición.
* **(c) ¿Cuántas particiones son necesarias como mínimo para instalar GNU/Linux?**
  + Se requieren al menos dos particiones:
    1. **/ (root):** Es la partición principal donde se instalará el sistema operativo.
    2. **Swap:** Espacio de intercambio. Sirve como memoria virtual.
* **(d) Ejemplos de particionamiento:**
  + **Servidor:** Podrías tener particiones separadas para /home, /var, /boot y /.
  + **Escritorio estándar:** Solo con / y swap.
  + **Sistema multiusuario:** Puede ser útil tener una partición separada para /home y otra para /var para gestionar logs y usuarios.
* **(e) Software para particionar:**
  + **GParted:** Interfaz gráfica para gestionar particiones.
  + **fdisk:** Herramienta en línea de comandos para particionar discos.
  + **parted:** Similar a fdisk pero más avanzado, soporta particiones GPT.
  + **Comparación:** GParted es más amigable para usuarios, mientras que fdisk y parted son más potentes y preferidos por administradores de sistemas.

**8. Arranque (bootstrap) de un Sistema Operativo:**

* **(a) ¿Qué es el BIOS?**  
  Es el firmware que inicializa los dispositivos del sistema (teclado, pantalla, disco duro) y carga el gestor de arranque para iniciar el sistema operativo.
* **(b) ¿Qué es UEFI?**  
  Es el sucesor moderno del BIOS, con más características, soporte para particiones GPT, y mayor seguridad (incluyendo Secure Boot).
* **(c) ¿Qué es el MBR y el MBC?**
  + **MBR (Master Boot Record):** Es un sector de 512 bytes en el primer disco duro que contiene la tabla de particiones y el código de arranque.
  + **MBC (Master Boot Code):** Es el código dentro del MBR que se encarga de iniciar el proceso de carga del sistema operativo.
* **(d) ¿Qué es GPT?**  
  GPT (GUID Partition Table) es un estándar para la tabla de particiones que reemplaza al MBR, soportando discos más grandes y más particiones.
* **(e) Funcionalidad de un "Gestor de Arranque":**  
  Es un software que carga el sistema operativo en memoria. Existen varios tipos:
  + **GRUB:** Muy popular en GNU/Linux.
  + **LILO:** Antiguo, pero usado en algunas distribuciones.
  + Se instala en el MBR o en una partición específica.
* **(f) Pasos desde que se prende una computadora hasta que el Sistema Operativo es cargado (proceso de bootstrap):**
  + El BIOS o UEFI se ejecuta y verifica el hardware.
  + El BIOS o UEFI localiza el gestor de arranque.
  + El gestor de arranque carga el kernel del sistema operativo en memoria.
  + El kernel inicializa el resto del sistema.
* **(g) Proceso de arranque en GNU/Linux:**
  + BIOS o UEFI ejecuta GRUB.
  + GRUB carga el kernel en memoria.
  + El kernel inicia y monta el sistema de archivos.
  + Se inician los procesos de usuario y servicios del sistema.
* **(h) Proceso de parada (shutdown):**
  + Cierre de aplicaciones y procesos.
  + Sincronización de datos pendientes en disco.
  + Desmontaje de sistemas de archivos.
  + Apagado del hardware.
* **(i) ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado?**  
  Sí, se puede hacer "dual boot", instalando GNU/Linux y otro sistema operativo (como Windows) en particiones diferentes.

**9. Archivos:**

* **(a) Identificación de archivos en GNU/Linux:** Los archivos se identifican por su nombre y permisos. Los tipos de archivos incluyen regulares, directorios, enlaces simbólicos, etc. Los permisos (lectura, escritura, ejecución) se representan con una secuencia de letras como rwxr-xr--.
* **(b) Funcionamiento de los editores vi y mcedit, y los comandos cat y more:**
  + **vi:** Editor de texto modal, con modos de edición y comandos.
  + **mcedit:** Editor de texto gráfico asociado a Midnight Commander.
  + **cat:** Muestra el contenido de archivos concatenados.
  + **more:** Muestra el contenido de archivos de forma paginada.
* **(c) Crear un archivo llamado "prueba.exe" usando vi:**
  + Ejecuta vi prueba.exe.
  + Escribe tu nombre y número de alumno.
  + Guarda el archivo y sal usando :wq.
* **(d) Funcionamiento del comando file:**  
  Muestra el tipo de archivo sin necesidad de abrirlo. Puede distinguir entre archivos de texto, ejecutables, imágenes, etc.

**10. Comandos:**

* **(a) Crear la carpeta ISO2017:**
  + Comando: mkdir ISO2017
* **(b) Acceder a la carpeta:**
  + Comando: cd ISO2017
* **(c) Crear dos archivos:**
  + Comando: touch iso2017-1 iso2017-2
* **(d) Listar el contenido del directorio:**
  + Comando: ls
* **(e) Ver la ruta actual:**
  + Comando: pwd
* *(f) Buscar archivos con la cadena "iso":*\*
  + Comando: find . -name "iso\*"
* **(g) Informar la cantidad de espacio libre en disco:**
  + Comando: df
* **(h) Verificar usuarios conectados:**
  + Comando: who
* **(i) Acceder al archivo iso2017-1 e ingresar Nombre y Apellido:**
  + Comando: nano iso2017-1 o vi iso2017-1
* **(j) Mostrar las últimas líneas de un archivo:**
  + Comando: tail

**11. Comandos:**

* **(a) shutdown:** Apaga el sistema.
* **(b) reboot:** Reinicia el sistema.
* **(c) halt:** Detiene todas las operaciones del sistema.
* **(d) locate:** Busca archivos por nombre.
* **(e) uname:** Muestra información del sistema.
* **(f) gmesg:** Muestra mensajes del kernel.
* **(g) lspci:** Lista todos los dispositivos PCI.
* **(h) at:** Programa tareas para que se ejecuten en un momento determinado.
* **(i) netstat:** Muestra las conexiones de red.
* **(j) mount:** Monta un sistema de archivos.
* **(k) umount:** Desmonta un sistema de archivos.
* **(l) head:** Muestra las primeras líneas de un archivo.
* **(m) losetup:** Configura dispositivos loop.
* **(n) write:** Envía mensajes a otros usuarios.
* **(ñ) mkfs:** Crea un sistema de archivos.
* **(o) fdisk:** Manipula la tabla de particiones del disco.

**12. Almacenamiento de comandos:**

* Los comandos del sistema están en directorios como:
  + **/bin**
  + **/usr/bin**
  + **/usr/local/bin**
  + **/sbin**
  + **/usr/sbin**