**Ejercicio 1:**

1. GNU/ Linux es un sistema operativo tipo Unix pero libre, diseñado por miles de programadores el cual es gratuito y de libre distribución con diversas distribuciones y de código abierto
2. **Windows**: no es de software libre, diseñado para que sea lo más fácil de utilizar a diferencia de algunas distribuciones de GNU/ Linux, no tiene tanta estabilidad como GNU/ Linux. Es diseñado por Microsoft, no es de cogido abierto

**IOS:** desarrollado por Apple Inc. Tipo software propietario, tipo Unix- like basado en Darwin

1. GNU es GNU no es Unix
2. Inicio como un proyecto en 1983 para crear un Unix libre, para ello fue necesario crear un marco regulatorio conocido como GPL. Luego se creo el FSF para financiar este proyecto. En 1990 ya contaba con ciertas funcionalidades pero faltaba el núcleo
3. Se entiende por multitarea que es una capacidad para que varios procesos/ apps se ejecuten al mismo tiempo, compartiendo uno o mas procesadores. GNU/ Linux hace uso de esto.
4. **POSIX** (Portable Operating System Interface, X= Unix) es una norma escrita por la IEEE (Instituto de ingenieros eléctricos y electrónicos) que define una interfaz estándar del sistema operativo, incluyendo el interprete de comandos o “shell”

**Ejercicio 2:**

1. Una distribución de GNU/ Linux es una distribución basada en el núcleo de Linux que incluye determinados paquetes de SW para satisfacer las necesidades de un grupo especifico de usuarios, dando origen a ediciones domésticas, empresariales o para servidores
   1. BlackArch Linux: orientada a la seguridad informática. Tiene herramientas para hacking como splatctl
   2. Elementary OS: una de las mas personalizables de Ubuntu, imita a OS X de Apple
   3. ArtistX: distribución discontinuada enfocada en la producción multimedia. Basada en Ubuntu. Diseñada especialmente para formato LiveDVD y USB
   4. Ubuntu: sistema operativo para ordenadores y servidores. Orientado al usuario promedio con facilidad de uso y mejor experiencia
2. Las distribuciones se diferencian según el objetivo de cada uno ya que tienen distintos enfoques
3. Debian es un sistema operativo de tipo libre que usa el núcleo Linux
4. Debian es un sistema operativo de tipo libre que usa una nueva distribución que se realizaría de forma abierta, en la línea del espíritu de Linux y GNU. Estaba pensada para ser mantenida y creada de forma cuidadosa. Comenzó con un grupo de pocos hackers de software y ahora es una comunidad bien organizada

**Ejercicio 3:**

1. Los componentes primordiales son el kernel, Shell, bootloader, software GNU, servidor grafico
2. Multitarea, multiusuario y multiprocesador, portable, comandos, directorios, código abierto
3. Kernel es el componente central que da vida al hardware. Asegura que todos los programas y procesos tengan acceso a laos recursos que necesitan de la maquina

Seguido al kernel está la capa de Shell. Actúa como un intérprete de comandos

Bootloader es un software cuya tarea es cargar el sistema operativo en la memoria

Software GNU proporciona utilidades básicas como ls, cp o rm

Servidor grafico

**Ejercicio 4:**

1. Es un software que constituye una parte fundamental del sistema operativo, es el responsable de facilidad el acceso asegurado de hardware de la computadora

En 1991 se empezó a trabajar en un núcleo de sistema operativo similar al Unix y se diseño monolítico. Se comenzó a colaborar con el código. Hubo varias versiones y se anunció la unión con GNU

1. –
2. La versión actual del Kernel es Linux kernel 5.8, lanzado el 2/8/2020
3. Sí, si hay distintas distribuciones, si hay una sola, no
4. Se encuentra en el boot
5. El kernel es monolítico hibrido. Se ejecutan en un espacio privilegiado con acceso irrestricto al hardware

**Ejercicio 5:**

1. –
2. Respuestas
   1. Programa informático que traduce ordenes de los usuarios mediante instrucciones. Se introducen mediante sintaxis
   2. **LS:** indica que tengo que listar un directorio donde se ejecuta la orden
   3. **CP:** para copiar
   4. Las propios están internos en la consola. Los externos en el disco duro, mediante directorios, indicados como path
   5. Porque es algo único en cambio la Shell tiene varias implementaciones y configuraciones
   6. Es posible, solo lo que puede hacer el administrador. Se cambia al inicio de sesión con programas

**Ejercicio 6:**

1. Es un componente que controla cómo se almacenan y recuperan los datos. Administra y facilita el uso de memorias y periféricos
2. Los más importantes son
3. **/:**  tope de la estructura de directorios
4. /home: almacena archivos de usuarios
5. **/etc:** archivos de configuración
6. **/var:** información que varía de tamaño
7. **/bin:** archivos binarios y ejecutables
8. **/tev:** enlace a dispositivos
9. **/usr:** aplicaciones de usuarios
10. Partición FAT: tabla de asignación de archivos
11. No es posible, debería ser posible montarlos
12. En GNU/Linux el sistema de archivos reside bajo un árbol jerárquico conocido como FHS. El cual es un estándar que establece y brinda detalles de nombres, contenidos, ubicaciones y permisos de archivos y directorios/ es un conjunto de reglas que determina la estructuración de archivos y directorios

**Ejercicio 7:**

1. Una partición es el nombre de cada división presente en una sola unidad física de almacenamiento de discos. Cada uno tiene su propio sistema de archivos
2. Particiones
3. **Primaria:** dependen de la tabla de particiones, detecta el ordenador al arrancar. Puede haber como máximo 4
4. **Extendido/ secundario:** para tener mas de 4 particiones pero no podemos instalar un sistema operativo o sea solo sirve para almacenar datos. Podemos hacer las particiones que queramos, lógicos
5. **Lógica:** las que se hacen dentro de una extendida, se necesita asignarle tamaño. Se conectan como una lista enlazada
6. Se identifican siempre con las letras: /dev seguido de un id para las particiones, que empieza por /sd a esta le sigue una letra del necesario que identifica al disco y un numero que identifica la partición. De manera que el disco será /sda y los particiones /sda 1, /sda2, /sda3, etc
7. Como mínimo son necesarios 1. Es recomendable / y SWAP (para datos temporales)
8. **Active Partition Manager:** compatible con FAT16, FAT32, NTFS y También crea particiones en memorias USB y permite editar manualmente las tablas de particiones

**Disk genius:** recupera datos, administra particiones, copia de seguridad, restauración, utilizades, tiene editor hexadecimal para ver y editar contenido binario

**Minital partition wizard:** para también reconstruir el MBR

**Ejercicio 8:**

1. El basic impt/ output system: define la interfaz del firmware, es el primer programa que se ejecuta cuando se enciende la computadora. Su propósito principal es iniciar y probar el hardware y cargar un gestor de arranque o un sistema operativo desde un dispositivo de almacenamiento de datos y provee una capa de abstracción del hardware
2. La UEFI es el firmware sucesor del BIOS añadiéndole mas funcionalidades. Las diferencias principales con la BIOS es que
   1. La BIOS tiene un diseño con teclado y la UEFI tiene una interfaz más moderna
   2. Se conecta a internet para actualizarse
   3. Se ejecuta en 32/64 bits, BIOS es de 16 bits
   4. Arranque más rápido con UEFI
3. MBR (registro de arranque principal): primer sector de un dispositivo de almacenamiento de datos, se emplea como bootsrap o para almacenar la tabla de particiones o solo para identificar un dispositivo

MBC: es un trozo de código almacenado en el MBR, denominado **código maestro de arranque**. El Bootstrap loader repasa la tabla de particiones buscando una partición activa. Cuando la encuentra, ejecuta su sector inicial, carga el código inicial en memoria y le transfiere el control, el cual ya es capaz de cargar y ejecutar otro programa como sistema operativo o alguna utilidad como el sector de arranque

1. GIP se refiere a GUID partición table scheme, el cual satisface al MBR usando en el BIOS. Es un estándar de particionado para la colocación de la tabla de particiones en un disco duro. No es mas que un estilo distinto de tabla. GPT es el sistema propio para UEFI
2. Es un programa que permite elegir el siguiente código a ejecutar en el proceso de arranque

**GRUB: (grand Unifer Bootloader):** permite tener distintos sistemas operativos en el mismo disco. Permitiendo cual iniciar

**LILO (Linux loader):** funciona con variedad de sistemas de archivos y puedo arrancar un sistema operativo desde el disco duro o un disco externo. Permite 16 imágenes de arranque. Se puede instalar en el MBR

**NTLD:** archivo ejecutado por el sector de arranque y permite elegir sistema de destino. Ha sido un gestor para Windows

**BOOTMGR (boot manager);** gestor de Windows vista/7 que sustituye a NTLD.

1. Se enciende la computadora, arranca el procesador y se inicializa y ejecuta un pequeño programa almacenado en la placa base llamado UEFI. La UEFI carga algunos ajustes de configuración desde la memoria en la placa. Luego la CPU ejecuta la BIOS para probar e inicializar el hardware en el sistema. Después de inicializar el hardware, el UEFI busca un dispositivo de arranque para arrancar el sistema operativo. Luego el gestor de arranque lanza el sistema instalado
2. –
   1. La BIOS/ UEFI busca el cargador de arranque y lo ejecuta. Cuando localiza el cargador, lo traspasa a memoria y le da el control/ se inicia el MBR según el gestor de arranque, el MBR carga y ejecuta el gestor El gestor espera nuestra elección de arranque o sino elige el predeterminado. En definitiva, el gestor carga y ejecuta el kernel. Al iniciar el kernel, se monta el sistema de archivos raíz, según el nivel de ejecución, el sistema iniciará las aplicaciones de los directores
3. Shutdown. Se notifica a los usuarios este hecho y bloquea el sistema para que nadie pueda acceder, salgo el root. Se envía la señal SIGTERM a todos los procesos provocando que terminen su ejecución de forma ordenado. Se envía la señal SIGKILL para que los procesos que no atendieron a SIGTERM concluyan también. Shutdown lleva a cabo su cometido enviando una señal a init para que cambie a uno de estos niveles de ejecución, la función se debe conseguir: apagar u obtener el sistema, entrar en monousuario o reiniciarlo. Cambiado el nivel de ejecución, se invoca a halt, reboot o poweroff
4. Es posible convirtiendo al dispositivo el dualboot. Debo tener espacio suficiente

**Ejercicio 9:**

1. Se identifican por su nombre, pueden tener cualquier carácter menos / y están limitados a 256 caracteres de longitud. No se basa en la extensión para saber el tipo de archivos sino que usa “magic numbers”, es un sistema que determina tipo de archivos mediante unos test que comprueban qué tipo de archivo es
2. **VI:** es un programa que entra en la categoría de editar texto, a diferencia de un procesador de texto, no ofrece herramientas para determinar visualmente cómo quedará

**Mcedit:** es un ejecutable independiente que nos permite visualizar y editar contenido de archivos/ ficheros

**Comando cat:** comando que permite crear, fusiones o imprimir archivos en la pantalla de salida estándar

**Comando more:** comando para solo visualizar el contenido de un archivo y verlo por páginas

**file:** informa el tipo de archivo

**Ejercicio 10:**

1. **MKDIR:** crea carpeta
2. **CD:** accede a una carpeta
3. **Touch:** crea archivo
4. **CD:** lista contenida
5. **PWD:** visualiza ruta
6. **Find:** busca archivos
7. **DF:** informa espacio libre
8. **Who:** verifica usuarios

**Ejercicio 11:**

1. **Poweroff:** -h apaga inmediatamente
2. **Reboot:** reinicia
3. **Halt:** detiene
4. **–**
5. **Uname:** usado para imprimir información básica del sistema
6. **Dmesg:** usado para examinar el buffer
7. **/spci:** lista información sobre el PCI conectados al sistema kernel e imprimir el bufer de mensajes del kernel
8. **At:** ejecuta comandos a la hora especificada
9. **Netstat:** muestra conexiones de red, enrutamiento, estadística de interfaz de red
10. **Mount:** monta dispositivos y particiones para su uso para el sistema operativo
11. **Unmount:** elimina anclaje entre dispositivo especial y el punto de montaje
12. **Head:** muestra de modo predeterminado las diez primeas líneas de un archivo
13. **Losetup:** fija el dispositivo de bucle
14. **Write:** envía un mensaje a otro usuario del sistema, write “nombre de usuario”
15. **Fdisk:** visualiza y administra particiones de disco duro en Linux

**Ejercicio 12:**

1. Poweroff: -n no sincroniza antes de reiniciar o detener
   1. -w no reinicia/ detiene el sistema únicamente escribe la grabación en wtmp
   2. -d no escribe la grabación wtmp
   3. -f fuerza una detención o reinicia, no llama a shutdown
   4. -i apaga todas las interfaces de redes justo antes de detener el sistema
   5. -p cuando se detiene el sistema hace un apagado
   6. Shutdown hh:mm programa apagado
   7. Shutdown +m: m cantidad de minuts
   8. Shutdown +m “mensaje” en minutos y un mensaje
2. Reboot solo reboot
3. Halt -p manda señal para que se apague la fuente de administración
4. Locate. Busca un archivo, se encarga de realizar las búsquedas haciendo uno de una base de realizar las búsquedas
5. Locate -r obtiene archivo con nombre exacto
6. Locate -c: muestra numero de archivos de la búsqueda
7. Locate -i: ignora distinción entre mayúsculas y minúsculas
8. Locate -e: muestra incluso un archivo eliminado
9. Locate -n: -n <número> limita el número de resultados
10. **Uname**
11. Uname -s: obtiene nombre de kernel
12. Uname -r: obtiene versión de kernel
13. -v obtiene versión del SO
14. -o: nombre del SO
15. -n: nombre que identifica nuestro sistema operativo en la red
16. -m: solicita tipo de arquitectura
17. -a: imprime todo
18. **Gmesg:** herramienta que lista avisos temporales generados por el kernel
19. -h: formato para que visualmente sea más comprensible
20. -t: imprime tiempo
21. **Lspci:**