TP1 ISO

1. Características de GNU/Linux:

a. Mencione y explique las características más relevantes de GNU/Linux.  
GNU/LINUX es un sistema operativo de tipo UNIX, es de código abierto y es un software gratuito y de libre distribución. Las caractiscas mas relevantes son:  
-Es multiusuario: Permite que los usuarios accedan a los datos o procesos de una maquina desde diferentes computadoras o terminales  
-Es muktitarea y multiprocesador  
-Es multiplataforma  
-Todo es un archivo, se organizan en una estructura jerarquica de tipo árbol  
-Cualquiera puede utilizarlo modificarlo y redistribuirlo

b. Mencione otros sistemas operativos y compárelos con GNU/Linux en cuanto a los puntos mencionados en el inciso a.  
Windows: Es de código cerrado y no es gratuito. No se puede modificar el sistema operativo de la misma medida. Es mas vulnerable a virus y malware y no es compatible con tantas arquitecturas y dispositivos  
MacOs: Es un SO de código cerrado y solo se ejecuta en dispositivos fabricados por apple. No es tan personalizable como GNU

c. ¿Qué es GNU?

GNU es un sistema operativo UNIX de código abierto y gratuito, iniciado en 1983 por Richard Stallman.

d. Indique una breve historia sobre la evolución del proyecto GNU.  
En 1983 Stallman crea el proyecto con el fin de crear un Unix libre y para asegurar esto creo el marco regulatorio conocido como GPL.  
En 1985 Stallman crea la FSF con el fin de financiar el proyecto, En 1990 ya contaba con un editor de textos, un compilador y gran cantidad de bibliotecas que componen un Unix típico  
Se decide utilizar el núcleo Linux creado por Linus Torvalds en 1991, convirtiéndose en el sistema operativo GNU/Linux  
La combinación de núcleo con las herramientas de software libre permitió la creación de un SO de alta calidad que fue lanzado en 1992  
Desde entonces el proyecto GNU ha seguido evolucionando y mejorando, hoy en dia es el movimiento importante en el mundo de software libre y ha influido en la creacion de muchos otros proyectos de software libre y gratuito.

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

La multitarea es la posibilidad de realizar varias operaciones, funciones, o ejecucuiones al mismo tiempo. Significa que puede estar haciendo varias tareas en segundo plano mientras el usuario esta trabajando en otra  
GnuLinux es un sistema multitarea. Con la ayuda del planificador de procesos, puede ejecutar multiples procesos en paralelos. El planificador es el responsable de asignar los recursos de la computadora a los distintos procesos,

f. ¿Qué es POSIX?

Posix (Interfaz de sistema operativo portátil): Es un estándar que define una interfaz estándar entre el SO y las aplicaciones. Su objetivo es promover la portabilidad de las aplicaciones y el software entre los diferentes SO.

POSIX especifica una serie de interfaces y comandos de línea de comandos que deben estar presentes en todos los SO que cumplan con el estándar.

2. Distribuciones de GNU/Linux:  
a. ¿Qué es una distribución de GNU/Linux? Nombre al menos 4 distribuciones de GNU/Linux y cite diferencias básicas entre ellas.  
  
Una distribución GNU/Linux es una versión del sistema operativo que incluye núcleo Linux, herramientas y utilidades de sistema y software adicional seleccionado por el desarrollador de la distribución.  
Cada distribución de GNU/Linux puede ser única en términos de conjuntos de software, apariencia y características, se distribuye bajo una variedad de licencias de software libre.  
Algunos Ej:  
-Debian  
-Mint  
-Ubuntu

b. ¿En qué se diferencia una distribución de otra?  
Se diferencia en que cada una tiene sus propios objetivos, enfoque y filosofía. Algunas son diseñadas para ser fáciles de usar, otras en estabilidad y seguridad o diseño  
Ademas se diferencian en cuanto a su conjunto de software y paquetes preinstalados y la forma que se gestionan y actualizan.

c. ¿Qué es Debian? Acceda al sitio https://www.debian.org/ e indique cuáles son  
los objetivos del proyecto y una breve cronología del mismo.

DEBIAN es una distribución de GNULinux que se enfoca en la estabilidad, seguridad y libertad del software, es una de las mas antiguas y populares. Anunciada en 1993 y la primera versión estable lanzada en 1996. Los nombres de las versiones fueron tomados de películas de Toy Story XD.

3. Estructura de GNU/Linux:  
a. Nombre cuales son los 3 componentes fundamentales de GNU/Linux.

Los 3 componentes fundamentales son

El kernel, el Shell y el FileSystem

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

4) Kernel:

a. ¿Cuáles son sus funciones principales?  
Este es el núcleo del sistema operativo, gestiona recursos del hardware y proporciona una interfaz de programación y servicios del sistema. Se encarga de que el software y el hardware puedan trabajar juntos.

b. ¿Cuál es la versión actual? ¿Cómo se definía el esquema de versionado del Kernel en versiones anteriores a la 2.4? ¿Qué cambió en el versionado que se impuso a partir de la versión 2.6?  
La versión actual es la 5.16.  
En las versiones anteriores a la 2.4 el esquema del versionado seguía un patron de numeración de versiones impares para las de desarrollo y pares para las estables  
A partir de la 2.6 el esquema d enumeración de versiones cambio a un sistema de números separados por puntos. En donde el primer punto significa una versión importante, el segundo una versión menor y el tercero una corrección de errores

c. ¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?  
Si es posible tener mas de un kernel, pero no dos nucleos corriendo simultáneamente en Linux.

d. ¿Dónde se encuentra ubicado dentro del File System?  
Se encuentra ubicado en el directorio /boot o /lib del sistema Linux

5. Intérprete de comandos (Shell):

a. ¿Qué es?  
El interprete de comandos es una interfaz de línea de comandos para que los usuarios interactúen con el sistema operativo.

b. ¿Cuáles son sus funciones?  
Sus funciones son ejecutar comandos ingresados por el usuario los cuales pueden ser programas ejecutables, o proporcionar comandos para gestionar archivos y directorios. Ademas de configurar el S.O, administrar usuarios, grupos y asi como permisos de archivos y directorios.

c. Mencione al menos 3 intérpretes de comandos que posee GNU/Linux y compárelos entre ellos.  
-Bourne Shell: Es el mas utilizado en la actualidad, su símbolo es $. Shell estándar de casi todos los sistemas U/L  
-Korn Shell: Amplia el shell añadiendo historial de ordenes, edición en línea de ordenes, edición en línea y características ampliadas de programación.

-Bourne Again Shell: Creado para el proyecto GNU. Su símbolo de sistema es nombre\_usuario@nombre\_de\_equipo.

d. ¿Dónde se ubican (path) los comandos propios y externos al Shell?  
No tienen una ubicación especifica en el sistema, ya que son funciones de la propia shell. Los comandos externos se ubican en uno o varios directorios especificados en la variable PATH.

e. ¿Por qué considera que el Shell no es parte del Kernel de GNU/Linux?  
No es parte del kernel ya que no es una parte esencial del S.O. El kernel es el núcleo del sistema operativo y el responsable de la gestión de recursos. Mientras que el shell es una interfaz de línea de comandos que permite al usuario interactuar con el SO y ejecutar comandos.

f. ¿Es posible definir un intérprete de comandos distinto para cada usuario?  
Si, se puede definir para cada usuario  
¿Desde dónde se define?   
En /etc/passwd  
¿Cualquier usuario puede realizar dicha tarea?Cualquier usuario puede definir su o sus interpretes

6. El sistema de Archivos (File System) en Linux:  
a. ¿Qué es?

El file system es el que organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento

b. ¿Cuál es la estructura básica de los File System en GNU/Linux? Mencione los directorios más importantes e indique qué tipo de información se encuentra en ellos. ¿A qué hace referencia la sigla FHS?  
El sistema de archivos sigue una estructura jerarquica en la que todo se organiza en una única estructura de árbol:  
/: Raiz de la estructura de directorios.

/home: Se almacenan los archivos de usuarios

/var: Información que varia en tamaño

/etc: archivos de configuración del sistema

/bin: archivos binarios y ejecutables.

/dev: enlace a dispositivos

/usr: aplicaciones de usuarios y que no son esenciales para el funcionamiento del sistema.

FHS hace referencia a Filesystem Hierarchy Standard, que es el estándar que siguen todos los sistemas operativos unix para organizar el sistema de archivos.

c. Mencione sistemas de archivos soportados por GNU/Linux.  
Algunos de los sitemas utilizados hoy en dia son ext2, ext3, reiserfs, XFS.

d. ¿Es posible visualizar particiones del tipo FAT y NTFS (que son de Windows) en GNU/Linux?  
Si, es posible visualizar particiones del tipo FAT y NTFS.

7. Particiones:

a. Definición. Tipos de particiones. Ventajas y Desventajas.  
Una partición es una división lógica de un disco duro físico en uno o mas partes separadas. Cada uno con su propio sistema de archivos. Puede tener un tamaño y formato de sistema de archivos diferente.  
  
Existen dos tipos de particiones:  
Primarias: Contienen un sistema de archivos y pueden ser montadas en el sistema. En una tabala de particiones solo pueden haber cuatro particiones primarias.

Extendidas: Sirve para contener unidades lógicas en su enterior. Solo puede existir una partición de este tipo por disco. No se define un tipo de FS sobre ella.

Logica: Ocupa la totalidad o parte de la partición extendida, se le define un tipo de FS, se conecta como una lista enlazada.

Ventajas: Las particiones permiten organizar la información en el disco. Brinda mayor estabilidad ya qie al presentarse un problema en esa partición, no afecta a las demás. Tambien se pueden aplicar diferentes permisos y políticas de seguridad a cada una de ellas.  
Desventajas: La creación y gestión puede ser compleja. Se puede desperdiciar el espacio del disco si no se asigna correctamente. Podemos tardar en encontrar algo si existen muchas particiones.

b. ¿Cómo se identifican las particiones en GNU/Linux? (Considere discos IDE, SCSI y SATA).  
En el primer disco duro IDE se conoce como /dev/hda. Si tenemos un segundo disco duro IDE se llamara /dev/hdb, etc

En SCSI y SATA se denominan /dev/sda, /dev/sdb etc. En cada disco son representadas añadiendo un numero decimal al nombre del disco sda1 sd2.

c. ¿Cuántas particiones son necesarias como mínimo para instalar GNU/Linux?  
Nómbrelas indicando tipo de partición, identificación, tipo de File System y punto de montaje.

d. Dar ejemplos de diversos casos de particionamiento dependiendo del tipo de tarea que se deba realizar en su sistema operativo.

e. ¿Qué tipo de software para particionar existe? Menciónelos y compare

a) Una particio es una division logica de un disco duro fisico en uno o mas partes separadas. Cada una con su propio sistma de archivos. Cada partición se trata como una unidad separada por el S.O y puede tener un tamanio y formato de sistema de archivos diferente.

Existe dos tipos de particiones:  
Primarias: Contienen un sistema de archivos y pueden ser montadas en el sistema. Solo pueden haber cuatro particiones primarias en una tabla de particiones

Extendidas: Sirve para contener unidades lógicas en su interior. Solo puede existir una partición de este tipo por disco. No se define un tipo de FS sobre ella.

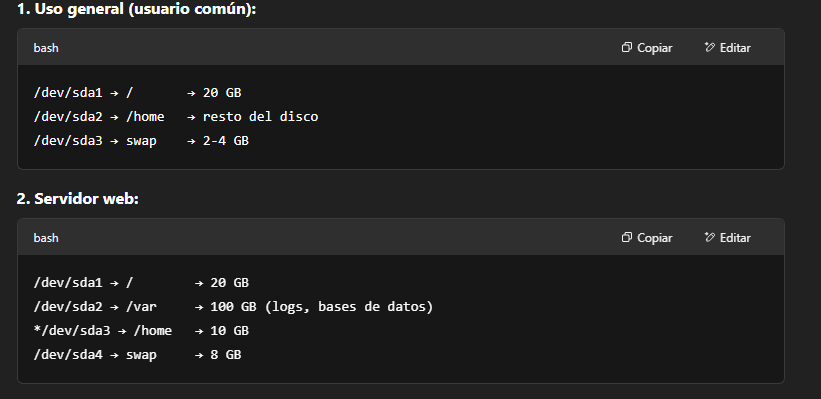
Logicas: Ocupa la totalidad o parte de la partición extendida. Se define un FS. Se conectan como una lista enlazada.

Ventajas: La particiones permite organizar la informacio e el dsco. Brinda mayor estailidad, ya quie al presentarse un problema en una no afecta a las demás.

Desventajas: La creación y gestión puede ser compleja. Se pueden desperdiciar espacio en el disco si no se asigna correctamente. Podemos tarar mas si tenemos muchas particiones.

b)Se representan como archivos en /dev

c) Es necesaria una como minimo pero es recomendable crear al menos dos particiones, una para el sistema de archivos / y otra para la memoria de intercambio (swap).

d) 

e) Existen dos tipos de software para particionar:

-Destructivo: Crea, elimia o redimesioa particiones sin borrar los datos existentes. (fdisk)

-No destructivo: Se pierden los datos al crear nuevas particiones o formatearlas. (gparted)

8. Arranque (bootstrap) de un Sistema Operativo:

a. ¿Qué es el BIOS? ¿Qué tarea realiza?  
La bios es un software que incializa y comprueba todos los componentes del hardware durante el arranque de la maquina. Prepara al equipo para que el SO se cargue y se ejecute. Esta almacennado en un chip de la placa madre.

b. ¿Qué es UEFI? ¿Cuál es su función?  
El Unified Extensible Firmware Interface, es una alianza entre varias companias con el obetivo de modernizar el proceso e arranque.  
Su función es preparar el hardware y el sistema para el inicio del SO. Es mas seguro y mas flexible que el bios clásico.

c. ¿Qué es el MBR? ¿Qué es el MBC?   
MBR: Es el primer sector del disco físico de 512 bytes que contiene el código de arranque (MBC) y la tabla de particiones del disco. Tambien contiene el gestor de aranque (bootloader) que iicia el proceso de carga del SO. El MBR se ejecuta al arracar la PC y su fucion es iniciar el proceso de carga del SO.  
La tabla de particiones de disco es una estructura que contiene información sobre como se divide y organiza el espacio de almacenamiento del disco en particiones  
MBC: es el codigo que se encuentra en el gestor de arranque de un disco duro y que se carga durante el proceso de arranque del sistema. El MBC se carga en la memoria y ejecuta el bootloader, que es un programa responsable de cargar el kernel del SO y pasar el control al mismo.

d. ¿A qué hacen referencia las siglas GPT? ¿Qué sustituye? Indique cuál es su formato  
Las siglas GPT hacen referencia a GUID Partition Table que es un tipo de tabla de particiones usado en los discos duro que reemplaza al MBR usado en los sistemas BIOS tradicionales.

Usa identificadores unicos para identificar las particiones en un disco duro, permitiendo la creacion de un gran numero de particiones en un disco y supera la limitacion de 4 primarias que tenia el MBR.

e. ¿Cuál es la funcionalidad de un “Gestor de Arranque”?   
El gestor de arranque es un programa cuya finalidad es cargar la imagen del kernel del sistema operativo en memoria y darle inicio. Se ejecuta luego del BIOS o UEFI y es una parte escencial del proceso de arrnaque  
¿Qué tipos existen?  
GRUB, LILO  
 ¿Dónde se instalan? Cite gestores de arranque conocidos.  
Puede estar ubicado en el primer sector del medio de arranque MBR o en una partición especifica el disco (común en UEFI)

f. ¿Cuáles son los pasos que se suceden desde que se prende una computadora hasta que el Sistema Operativo es cargado (proceso de bootstrap)?

Al encender la computadora, comienza el proceso de arranque, conocido como **bootstrap**. Los pasos son los siguientes:

**POST (Power-On Self Test):**  
**El firmware de la computadora (BIOS/UEFI)** inicia el proceso de **autoprueba de encendido (POST)**. En esta etapa, se verifica que el hardware de la computadora funcione correctamente: **memoria RAM**, **discos duros**, **periféricos**, entre otros.

**Carga del Bootloader:**  
Una vez que se completa el POST y todo el hardware está verificado, el **firmware** carga el **bootloader** desde el **sector de arranque del disco duro** o, en sistemas UEFI, desde la **partición EFI**.

**Selección de Sistema Operativo:**  
El **bootloader** presenta una lista de **sistemas operativos disponibles** para que el usuario elija el que desea iniciar.  
Si solo hay un sistema operativo instalado, el bootloader lo carga automáticamente sin mostrar la lista.

**Carga del Kernel:**  
El bootloader **carga el kernel del sistema operativo** en la **memoria RAM** y le pasa el control al kernel.

**Inicialización del Hardware:**  
Una vez el kernel toma el control, comienza a **inicializar el hardware de la computadora**.  
Carga los **controladores** necesarios para que todos los dispositivos funcionen correctamente.

**Inicio de Servicios:**  
El kernel **inicia los servicios** esenciales del sistema operativo, como la red, la gestión de usuarios, y otros servicios básicos para operar el sistema.

**Entorno Gráfico o Línea de Comandos:**  
Finalmente, dependiendo de la configuración, el sistema operativo **muestra el entorno gráfico** o la **línea de comandos** para que el usuario interactúe con el sistema.

g. Analice el proceso de arranque en GNU/Linux y describa sus principales pasos.

1- Encender la computadora

2- Se ejecuta la BIOS o UEFI que revisan y preparan todas las partes principales de la PC

3- Se ejecuta luego de la BIOS o UEFI el chequeo llamado POST antes del arranque que prueba todas las diferentes piezas de hardware. Si encuentra un problema antes de empezar, se mostrara por pantalla un mensaje de error o incluso mediante pitidos de la placa madre.

4-Procede el cargador de arranque que tiene como funcion ubicar el kernel del SO en un dispositivo de almacenamiento, luego lo carga rn memoria y a continuacion se ejecuta el codigo del kernel. Los mas comunes de linux son Lilo y Grub 2.

5- Ya cargado el kernel en memoria se le entrega el control para finalizar el proceso de inicio. El kernel toma el control de los recursos de la pc y comienza a iniciar todos los procesos y servicios en segundo plano

6- Se ejecuta el proceso INIT que tiene la responsabilidad de que el sistema quede completamente listo para usar. El mas usado en la actualidad es el systemd.

h. ¿Cuáles son los pasos que se suceden en el proceso de parada (shutdown) de GNU/Linux?

1. El usuario decide apagar el sistema:  
   El proceso de apagado comienza cuando el usuario emite el comando para apagar el sistema, ya sea desde la terminal (shutdown, poweroff, halt) o a través de la interfaz gráfica.
2. Notificación a aplicaciones y procesos:  
   El sistema notifica a las aplicaciones y procesos en ejecución que se va a apagar, dándoles tiempo para finalizar sus tareas de manera ordenada. En algunos casos, los procesos pueden ser interrumpidos si no responden después de un tiempo determinado.
3. Detención de servicios:  
   Los servicios en segundo plano (como servidores de bases de datos, redes, etc.) son detenidos ordenadamente, evitando que queden procesos en ejecución que puedan corromper el sistema.
4. Desmontaje de los sistemas de archivos:  
   Los sistemas de archivos montados se desmontan para garantizar que no haya datos pendientes de escritura y evitar la corrupción de los mismos. Esto incluye cualquier partición o dispositivo de almacenamiento conectado.
5. Desactivación de dispositivos y controladores:  
   Los dispositivos y controladores asociados al hardware son desactivados de manera controlada, permitiendo que cualquier tarea pendiente en los dispositivos de hardware sea completada.
6. Finalización de tareas pendientes por el kernel:  
   El kernel se encarga de finalizar todas las tareas pendientes, como la cancelación y destrucción de procesos en ejecución. Además, se asegura de que todo esté sincronizado para que no haya pérdida de información.
7. Apagado de los dispositivos y detención del sistema:  
   Finalmente, el sistema apaga los dispositivos y corta la energía, deteniendo el sistema operativo y completando el proceso de apagado.

i. ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique.  
Si, es posible gracias a las particiones del disco o a travez de maquinas virtuales.

9. Archivos y editores:

a. ¿Cómo se identifican los archivos en GNU/Linux?

Se identifican a través de su nombre y sus permisos y su tipo

b. Investigue el funcionamiento de los editores vim, mano y mcedit, y los comandos cat, more y less.

Vi: Es el editor de texto clasico en UNIX que se ejecuta en la terminal. Tiene funciones avanzadas como buscar, remplazar, mover y copiar texto. Tambien tiene un modo de comandos que permite ejecutar comandos del SO

Mcedit: Es un editor de trxto grafico que se ejecuta en la terminal. Tiene una interfaz grafica que facilita la edicion de texto. Tambien permite ejecutrar comandos del SO.

Comandos  
CAT: comando que muestra el contenido de un archivo de texto en la terminal.

More: comando que muestra el contenido de texto en la terminal, pero con la capacidad de desplazarse por el contenido pagina por pagina. Se usa para mostrar el contenido de archivos mas grandes que cat.

c. Cree un archivo llamado “prueba.exe” en su directorio personal usando el vim.  
El mismo debe contener su número de alumno y su nombre.  
vim ~/prueba.exe

21154/7 Nahuel Tranquillini  
esc

:wq

enter

d. Investigue el funcionamiento del comando file.  
 Pruébelo con diferentes archivos. ¿Qué diferencia nota?  
El comando file identifica el tipo de contenido de un archivo, sin importar su extensión.

e. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso de archivos:

i. cd: cambia de directorio, sirve para navegar entre carpetas:  
 cd .. vuelve al directorio anterior

cd ~ vuelve a home

ii. mkdir: crea carpetas nuevas

iii. rmdir: borra carpetas pero solo si están vacias  
rm -r carpeta borra la carpeta y todo lo que tiene adentro recursivamnente

iv. ln crea enlaces a archivos, ambos apuntan al mismo contenido, solo funciona con archivos y no carpetas

v. df: muestra cuanto espacio esta ocupado/libre  
df -h lo muestra en un formato mas legible

vi. tail muestra as ultimas lineas de un archive  
tail -n numero muestra las ultimas n líneas

vii. who: ver quien esta conectado en el sistema

viii. ls: muestra el contenido de una carpeta

ix. pwd: muestra el directorio actual

x. cp: copia archivos y carpetas  
cp -r c1 c2 copia directorios completos

xi. mv: mueve o renombra archivos  
mv archivo /otra/carpeta/ lo mueve

mv viejo.txt nuevo.txt renombra

xii. find: busca archivos por nobre tipo tamanio etc.

10. Indique qué comando es necesario utilizar para realizar cada una de las siguientes acciones. Investigue su funcionamiento y parámetros más importantes:

a. Cree la carpeta ISOCSO  
cd ISOCSO

b. Acceda a la carpeta  
ls ISOCSO

c. Cree dos archivos con los nombres isocso.txt e isocso.csv  
touch isocso.txt isocso.cvs

d. Liste el contenido del directorio actual  
ls

e. Visualizar la ruta donde estoy situado  
pwd

f. Busque todos los archivos en los que su nombre contiene la cadena “iso\*”  
find . -type f -name “\*iso\*”

g. Informar la cantidad de espacio libre en disco  
df -h

h. Verifique los usuarios conectado al sistema  
who

i. Editar a el archivo isocso.txt e ingresar Nombre y Apellido

vim isocso.txt

j. Mostrar en pantalla las últimas líneas de un archivo  
tail -5 archivo

11. Investigue el funcionamiento, parámetros y ubicación (directorio) de los siguientes comandos:

➢ man  
funcion: muestra el manual de ayuda de un commando  
parámetro: -k busca pr palabra clave  
ubicación: /usr/bin/man

➢ shutdown:  
funcion: apaga o reinicia el sistema de forma segura  
parámetro: -h (halt=apagar) -r (reboot) +5 apagar en 5 minutos o now  
ubicación: /usr/sbin/shutdown

➢ reboot  
funcion: reinicia el sistema inmediatamente  
ubicación: /usr/sbin/reboot

➢ halt  
funcion: muestra el manual de ayuda de un comando  
 ubicación: /usr/sbin/halt

➢ locate  
función: busca archivos usando una base de datos   
ubicación:/usr/bin/locate

➢ úname  
funcion: muestra información del sistema  
parámetro: -r versio del kernel -a toda la info  
ubicación: /usr/bin/uname

➢ dmesg  
función: muestra mensajes del kernel  
ubicación: /bin/dmesg

➢ lspci  
función: muestra los dispositivos PCI conectados al sistema.  
ubicación: /usr/bin/lspci

➢ at  
función: programa tareas para que se ejecuten una vez en el futuro  
ej: echo “shutdown -h now” | at 22:30  
ub: /usr/bin/at

➢ head  
función: muestra las primeras líneas de un archivo  
Parámetro: -n 10 muestra las primeras 10 lineas  
ubicación: /usr/bin/head

➢ tail

función: muestra las ultimas líneas de un archivo  
Parámetro: -n 10 muestra las ultimas 10 lineas  
ubicación: /usr/bin/tall

12. Procesos:

a. ¿Qué es un proceso? ¿A que hacen referencia las siglas PID y PPID? ¿Todos los procesos tienen estos atributos en GNU/Linux? Justifique. Indique qué otros atributos tiene un proceso.  
Un proceso es una instancia de un programa en ejecución. Cuando se ejcuta un comando o se abre una aplicación, el sistema operativo crea un proceso para gestionar esa ejecución.

PID (Process ID): es el identificador único que el sistema le asigna a cada proces.

PPID(Parent Process ID): Es el PID de prceso padre, osea el que inicio este nueo proceso. EJ: si ejecuito desde la terminal gedit el proceso terminal es el padre (tiene un PPID) y el gedit tiene su propio PID.

Todos los procesos tienen PID y PPID, incluso el primer proceso del sistema (init) tiene PPID=0, indicando que no fue iniciado por otro proceso

Algunos otros atributos son:  
-Estado: ejecución,suspendido,esperand  
-Priridad: que tan urgente es  
-Tiempo de CPU: Cuanto tiempo ha usado el procesador.

b. Investigue el funcionamiento, parámetros y ubicación (directorio) de los siguientes comandos relacionados a procesos. En caso de que algún comando no venga por defecto en la distribución que utiliza deberá proceder a instalarlo:

i. top: muestra los procesos actives en tiempo real (uso de cpu, memoria,etc)  
parámetros: -d <segundos> refresca cada n segundos  
 -p <PID> muestra solo cierto proceso  
ubicación: /usr/bin/top

ii. htop: es una interfaz mejorada que top<

parametros: -u <usuario> muestra solo procesos de ese usuario  
ubicacion: /usr/bin/htop (no viene por defecto)

iii. ps: muestra los procesos en ejecucion  
parametros: aux: lista todo  
 -u <usuario> muestra solo los procesos de un usuario  
 -p <pid> muestra un proceso especifico  
ubicacion: /bin/ps o /usr/bin/ps

iv. pstree: muestra los procesos en forma de arbol jerarquico  
parametros: -p muestra los PID  
 -u muestra los UID  
ubicacion: /usr/bin/pstree (no viene por defecto)

v. kill: envia señales a un proceso (terminandolo)  
ej: kill <pid>  
ubicacion: /bin/kill

vi. pgrep: busca procesos por nombre y devuelve su PID  
ej: pgrep firefox  
parametros: -u <usuario>  
ubicacion: /usr/bin/pgrep

vii. pkill: mata a procesos por nombre directamente  
parametros: -u <usuario> solo mata procesos de ese usuario  
ubicacion: /usr/bin/ppkill

viii. killall: mata a todos los procesos con un mismo nombre  
parametros: -u <usuario>  
ubicacion: /usr/bin/killall

ix. renice: Cambia la prioridad de un proceso que ya esta corriendo  
ej: renice -n 10 -p 1234  
parametros: -n valor de prioridad nuevo  
 -p <pid> pid del proceso

ubicacion: /usr/bin/renice

nice: ejecuta un comando con una prioridad definida  
ej: nice -n 10 ./mi\_programa  
parametros: -n <valor> nice(mas alto menor prioridad)  
ubicacion: /usr/bin/nice

13)Responda en forma sintética sobre los siguientes conceptos:

a. ¿Qué es un SO?   
Un sistema operativo es un software que actua como interediario entre el hardware y los programas de usuario, gestionando recursos y facilitando su uso.

b. Enumere qué componentes/aspectos del Hardware son necesarios para cumplir los objetivos de un SO.   
Algunos componentes necesarios para cumplir los objetivos de un SO son:  
-Procesador  
-Memoria RAM  
-Dispositivos de ES  
-Dispositivos de almacenamiento  
-Reloj del sistema timer  
-Drivers

c. Enumere componentes de un SO  
-Nucleo  
-Manejador de procesos y memoria  
-Sistema de archivos  
-Manejador de ES  
-Sistema de llamadas al sistema  
-Planificador

d. ¿Que es una llamada al sistema (system call)? ¿Cómo es posible implementarlas?   
Una system call es la forma en que un programa solicita servicios al SO (ej: abrir un archivo). Se implementan mediante interrupciones que pasan del modo usuario al modo kernel.

e. Defina Programa y Proceso.  
Programa: Conjunto de instrucciones estaticas.  
Proceso: Programa en ejecucion con su contexto (estado, registros,memoria etc)

f.¿Cuál es la información mínima que el SO debe tener sobre un proceso? ¿En qué estructura de datos asociada almacena dicha información?   
Debe tener el PCB que guarda:  
-ID del prceso  
-Estado  
-Contador de programa  
-Registros  
-Informacion de memoria  
-Info de E/S  
-Info de planifiacion

g. ¿Qué objetivos persiguen los algoritmos de planificación (scheduling).  
-Maximizar el uso del CPU  
-Reducir el tiempo de espera  
-Mejorar el tiempo de respuesta  
-Aumentar el rendimiento en general  
-Garantizar equidad entre procesos

h. ¿Qué significa que un algoritmo de scheduling sea apropiativo o no apropiativo (Preemptive o Non-Preemptive)?   
Apropiativo: El SO puede quitarle la CPU a un proceso en ejecucion  
No apropiativo: El proceso retiene a la CPU hasta que termina o la bloquea

i. ¿Qué tareas realizan los siguientes módulos de planificación?:

i. Short Term Scheduler: Decide que proceso se ejecuta en el CPU

ii. Long Term Scheduler: Controla cuantos procesos ingresan al sistema

iii. Medium Term Scheduler: Suspende y reanuda procesos para optimizar el uso de memoria

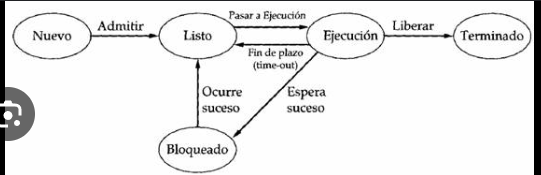
j.¿Qué tareas realiza el Dispatcher? ¿Y el Loader?   
Dispatcher: Cambia el contexto entre procesos, pasa el control de la CPU  
Loader: Carga el programa en memoria para su ejecucion

k. ¿Qué significa que un proceso sea “CPU Bound” y “I/O Bound”?  
CPU BOUND: Usa intensamente el procesador, limitado principalmente por la velocidad de este. Pasa la mayor parte del tiempo haciendo calculos intensivos y usa poco los dispositivos de IO

I/O Bound: Esta limitado por la velocidad de acceso a los disp de E/S. Pasa mas tiempo esperando respuestas de E/S que usando la CPU

l. ¿Cuáles son los estados posibles por los que puede atravesar un proceso? ¿Qué representa que un proceso se encuentre en los estados enumerados? Utilizando un diagrama explique las transiciones entre los estados.

1-Nuevo: El proceso fue creado pero aun no esta listo para ejecutarse  
2-Listo: Esta esperando su turno para usar la CPU  
3-Ejecutando: Actualmente ejecutado por la cpu  
4-Bloqueado: Esta esperando que termine una operación de E/S u otro recurso  
5-Terminado: Finalizo su ejecucion



m. ¿Cuáles de los schedulers mencionados anteriormente se encargan de las transiciones entre los estados enumerados?   
Long term: De nuevo a Listo  
Short term: Listo -> running y running ->Listo  
Medium term: Se encarga de suspender o reactivar procesos, de listo a suspendio

n. Defina Tiempo de retorno (TR) y Tiempo de espera (TE) para un proceso.   
TR: Es el tiempo total que tarda un proceso dede que entra al sistema hasta que se termina  
TE: Es el tiempo que el proceso pasa en la cola sin estar ejecutandose

o. Defina Tiempo Promedio de Retorno (TPR) y Tiempo promedio de espera (TPE) para un lote de procesos.   
TPR: Es el prommemdio de todos los tiempos de retorno de todos los procesos   
TPE: Es el promedio de los tiempos de espera de todos los procesos.

p. Defina tiempo de respuesta.  
Es el tiempo desde que un proceso llega al sistema hasta que empieza a ejecutarse por primera vez.

14) Para los siguientes algoritmos de scheduling:

➢ FCFS (First Come First Served)

➢ SJF (Shortest Job First)

➢ Round Robin

➢ Prioridades

a. Explique su funcionamiento mediante un ejemplo.

b. ¿Alguno de ellos cuentan con parámetros para su funcionamiento? Identifique y enunciarlos

c. Cual es el más adecuado según los tipos de procesos y/o SO.

d. Cite ventajas y desventajas de su uso

**FCFS (First Come First Served)**:

* **Funcionamiento**: Este algoritmo atiende los procesos en el orden en que llegan, como una cola FIFO (First In, First Out).
* **Ejemplo**: Si los procesos llegan en el orden P1 (4 unidades de CPU), P2 (3 unidades) y P3 (2 unidades), se ejecutan en ese mismo orden.
* **Parámetros**: Solo necesita saber el tiempo de llegada y la duracion del proceso
* **Adecuación**: Es útil en sistemas simples o cuando los tiempos de ejecución son similares.
* **Ventajas**: Fácil de implementar.
* **Desventajas**: Puede generar largos tiempos de espera para procesos cortos si hay procesos largos antes (problema de convoy).

**SJF (Shortest Job First)**:

* **Funcionamiento**: Prioriza los procesos con menor tiempo de ejecución. Puede ser no apropiativo o apropiativo (SRTF - Shortest Remaining Time First).
* **Ejemplo**: Si llegan P1 (6 unidades), P2 (2 unidades) y P3 (4 unidades), se ejecutan en el orden P2, P3, P1.
* **Parámetros**: Requiere conocer o estimar el tiempo de ejecución de los procesos.
* **Adecuación**: Ideal para sistemas con procesos cortos y predecibles.
* **Ventajas**: Minimiza el tiempo promedio de espera.
* **Desventajas**: Puede causar inanición para procesos largos.

**Round Robin**:

* **Funcionamiento**: Asigna un tiempo fijo (quantum) a cada proceso y los alterna en ciclos.
* **Ejemplo**: Con un quantum de 4, si llegan P1 (6 unidades), P2 (8 unidades) y P3 (4 unidades), se ejecutan en rondas hasta completar.
* **Parámetros**: El quantum es el parámetro clave.
* **Adecuación**: Adecuado para sistemas interactivos donde la equidad es importante.
* **Ventajas**: Justo y evita inanición.
* **Desventajas**: Si el quantum es muy grande, se comporta como FCFS; si es muy pequeño, aumenta el overhead.

**Prioridades**:

* **Funcionamiento**: Ejecuta los procesos según su prioridad. Puede ser estático o dinámico.
* **Ejemplo**: Si P1 tiene prioridad 1, P2 prioridad 2 y P3 prioridad 3, se ejecutan en ese orden.
* **Parámetros**: Las prioridades asignadas a los procesos.
* **Adecuación**: Útil en sistemas donde ciertos procesos son más críticos.
* **Ventajas**: Permite atender procesos importantes primero.
* **Desventajas**: Puede causar inanición para procesos de baja prioridad.

15- Para el algoritmo Round Robin, existen 2 variantes: Timer Fijo y Timer Variable.

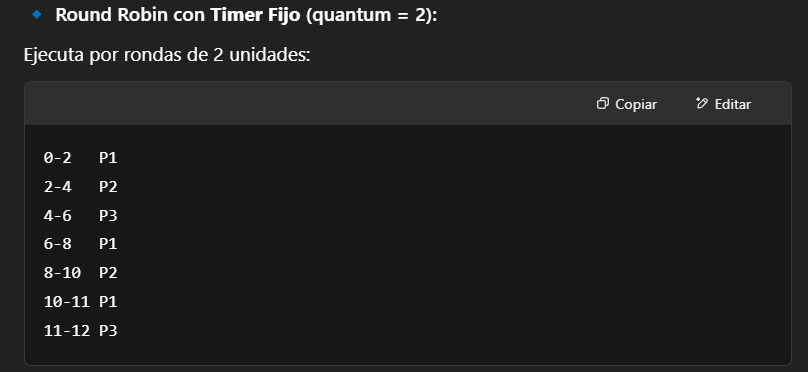
a.¿Qué significan estas 2 variantes?

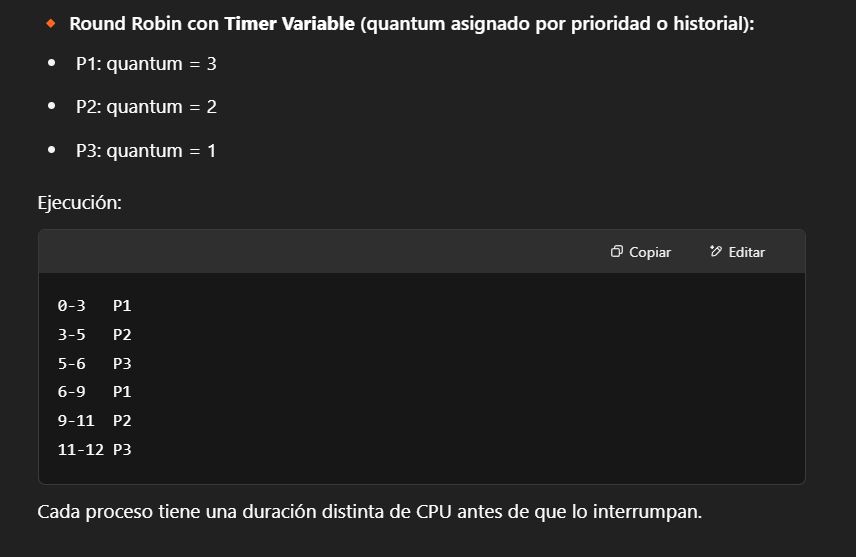
Timer Fijo:El quantum (tiempo máximo que puede usar la CPU un proceso antes de ser interrumpido) es igual para todos los procesos y se mantiene constante durante toda la ejecución.

Timer Variable:El quantum puede cambiar dependiendo del proceso o de otras condiciones (por ejemplo, su prioridad, su historial, etc.). No todos los procesos tienen el mismo quantum.

b. Explique mediante un ejemplo sus diferencias.







c. En cada variante ¿dónde debería residir la información del Quantum?   
Timer Fijo: El quantum se guarda en una estructura global del sistema operativo (por ejemplo, una variable del planificador). Todos los procesos comparten ese mismo valor.

Timer Variable:El quantum debe estar guardado en el PCB (Process Control Block) de cada proceso, ya que cada uno puede tener un quantum distinto.

16)

17)

18)

19)

20) Inanicion (Starvation)

a. Que significa?  
Inanicion significa que un proceso queda esperando indefinidamente porque nunca le dan la CPU para ser ejecutado. El proceso se encuentra listo en la cola pero nunca le dan CPU.

b. Cuales algoritmos vistos puede provocarla?  
-SJF Y SRTF ya que los procesos largos pueden quedar esperanndo indefinidamente  
-Prioridades ya que si siempre llegan procesos de mayor prioridad, los de baja pueden no ejecutarse nunca

c. Existe alguna tecnica que evite la inanicion para los algoritmos mencionados en el inciso b?

-Aging (envejecimiento): Consiste en aumentar la prioridad de un proceso a medida que pasa el tiempo sin ser seleccionado para su ejecucion.  
-Establecimiento de limites de espera  
-Prioridades dinamicas: Ajustan las prioridades de los procesos dinamicamente según el comportamiento y las necesidades del proceso.bbuenque

**Round Robin:**  
No diferencia procesos de CPU e I/O. Los procesos de I/O, que usan poco la CPU, tienen que esperar igual su turno, generando demoras innecesarias. Además, hay muchos cambios de contexto, lo que baja la eficiencia.

**SRTF:**  
Favorece a los procesos de I/O porque tienen ráfagas cortas, pero perjudica a los procesos de CPU, que pueden quedar esperando mucho tiempo (hambre o starvation) ya que siempre les pasan procesos más cortos por adelante.

22) **Round Robin:**  
No diferencia procesos de CPU e I/O. Los procesos de I/O, que usan poco la CPU, tienen que esperar igual su turno, generando demoras innecesarias. Además, hay muchos cambios de contexto, lo que baja la eficiencia.

**SRTF:**  
Favorece a los procesos de I/O porque tienen ráfagas cortas, pero perjudica a los procesos de CPU, que pueden quedar esperando mucho tiempo (hambre o starvation) ya que siempre les pasan procesos más cortos por adelante.

24. Suponga que un SO utiliza un algoritmo de VRR con Timer Variable para el planificar sus procesos. Para ello, el quantum es representado por un contador, que es decrementado en 1 unidad cada vez que ocurre una interrupción de reloj. ¿Bajo este esquema, puede suceder que el quantum de un proceso nunca llegue a 0 (cero)? Justifique su respuesta.  
Si podria ocurrir si un proceso termina su ejecucion en una rafaga de CPU antes de que se termine su Quantum.

25) RELLENO ZZZ

26) Colas Multinivel   
Actualmente los algoritmos de planificación vistos se han ido combinando para formar algoritmos más eficientes. Así surge el algoritmo de Colas Multinivel, donde la cola de procesos listos es dividida en varias colas, teniendo cada una su propio algoritmo de planificación.

a. Suponga que se tiene dos tipos de procesos: Interactivos y Batch. Cada uno de estos procesos se coloca en una cola según su tipo. ¿Qué algoritmo de los vistos utilizará para administrar cada una de estas colas?.

A su vez, se utiliza un algoritmo para administrar cada cola que se crea. Así, por ejemplo, el algoritmo podría determinar mediante prioridades sobre qué cola elegir un proceso.

b. Para el caso de las dos colas vistas en a: ¿Qué algoritmo utilizaría para planificarlas?

a) Procesos interactivos: Estos son procesos que necesitan interaccion con el usuario, esperan respuestas rapidas del sistema. Ej navegador web  
Utilizaria el algoritmo RR ya que permite repartir el tiempo de la CPU equitativamente y responde rapido  
Procesos batch: Es un proceso que no necesita interaccion humana directa mientras se ejecuta. Se ejecuta en segundo plano como un lote de tareas. Ej: copia de seguridad Utilizaria un algoritmo SJF o SRTF asi priorizan los procesos mas cortos y optimizan el TPE

b) Se podria utilizar un esquema de prioridades con aging.

27 y 28 muy largos de diagramar en papel