

I – INTRODUCCIÓN A S.O.

1. Contesta correctamente las siguientes cuestiones

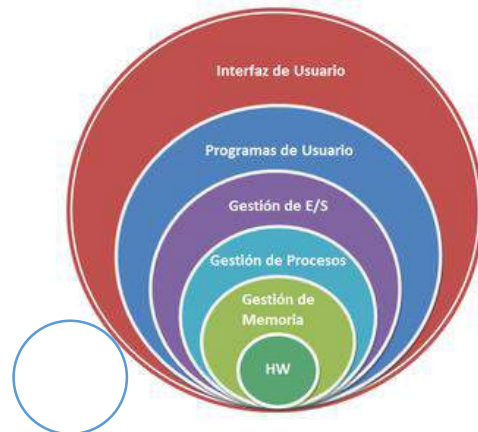
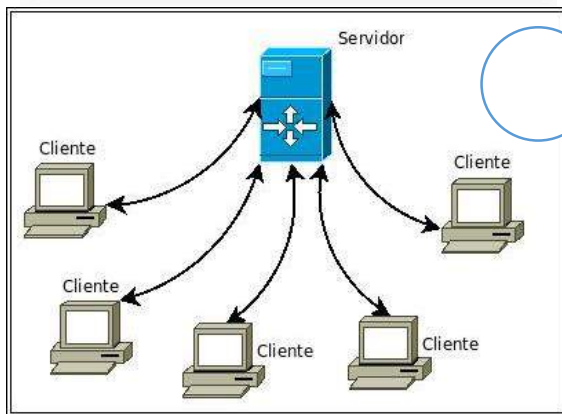
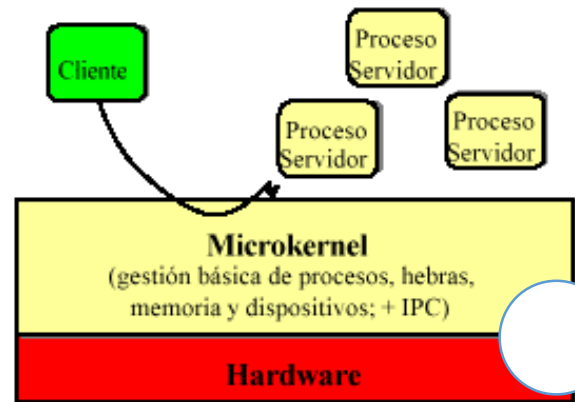
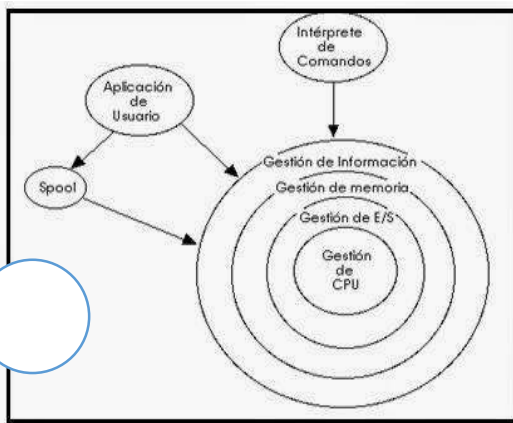
- a) Es el programa principal que se ejecuta en toda computadora de propósito general. (sistema operativo)
- b) Menciona las tres funciones primarias del sistema operativo. (abstracción, administración de recursos, aislamiento)
- c) En que se especializaban las computadoras en su primera época. (en tareas de cálculo intensivo)

II – ESTRUCTURAS Y FUNCIONES BÁSICAS

1.-Organizaciones de los SO

Compara las imágenes con las diferentes organizaciones de los SO y especifica cual es cual dentro del círculo.

1. Organización Modular
2. Organización Por Micro-Kernel
3. Organización Por Anillos Concéntricos
4. Organización Cliente-Servidor



2.-Arquitectura de CPU

Unidad de Control: El cerebro del microprocesador que genera todas las señales que lo gobiernan.

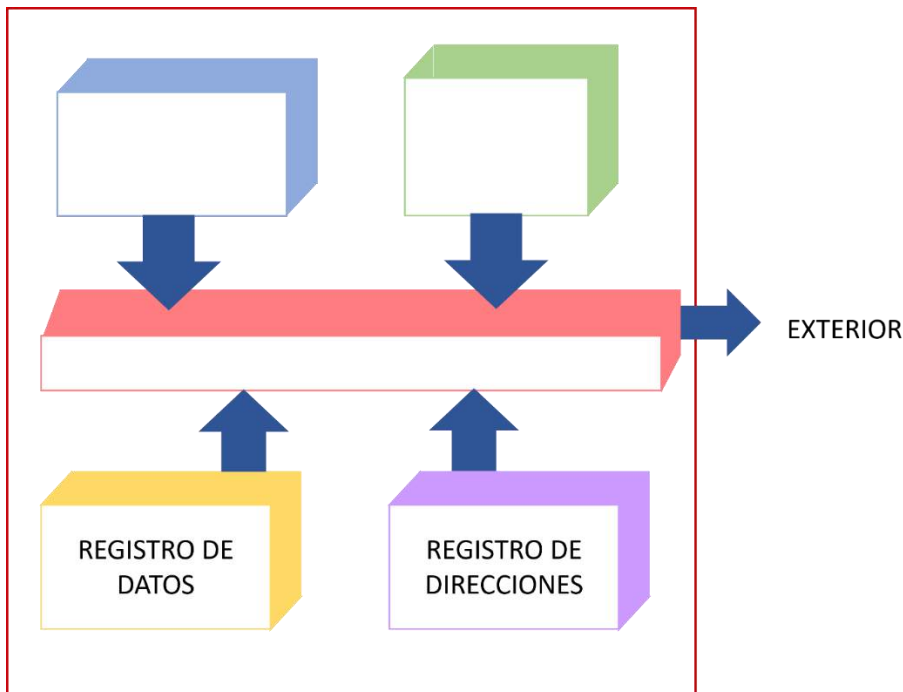
Unidad aritmético-lógica: conocida por sus siglas en inglés: ALU, es el bloque utilizado por la CPU para realizar los cálculos aritméticos o lógicos. La CPU se comunica con la ALU por medio del bus interno.

Buses internos: Empleados para comunicar todos los bloques de la CPU y para conectarse con los buses externos que dan a la CPU acceso al resto de componentes del ordenador.

Registros internos: Registros en los que se almacenan los datos para las operaciones y las direcciones de memoria a las que se quiera acceder.

Con base a la información anterior rellena la imagen siguiente:

1. Buses internos
2. Unidad aritmético-lógica
3. Registros internos
4. Unidad de control



3.-Buses & Bridges

Relaciona las columnas:

1	Son líneas de comunicación entre el procesador y los demás componentes del chipset
2	Son los encargados de enviar las órdenes desde la placa madre al procesador.

_____	Puentes (Bridges)
_____	Canales (Buses)

4.-Interrupciones y excepciones

Contesta las preguntas:

1. Describe que es una interrupción

2. Describe que es una excepción

III – CONCEPTO Y ESTADOS DE UN PROCESO

1. Responde correctamente las siguientes cuestiones:

- a) Es la imagen en memoria de un programa, junto con la información relacionada con el estado de su ejecución. (proceso)
- b) Es una entidad pasiva. (un programa)
- c) Es una entidad activa. (un proceso)

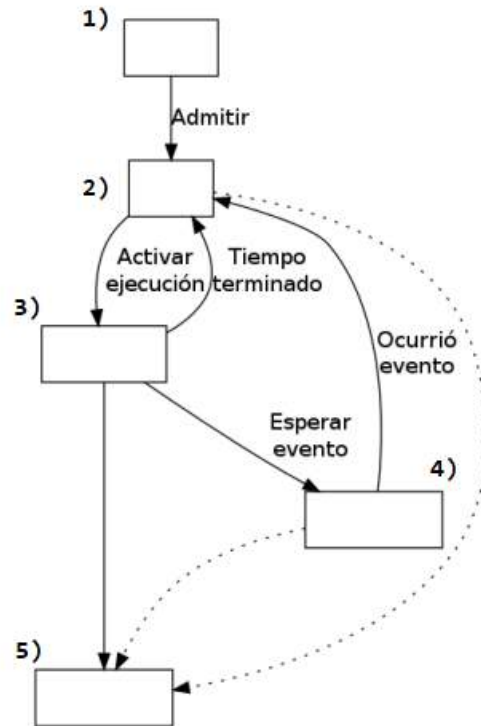
2. Relaciona correctamente los siguientes conceptos y definiciones referente a los estados de ejecución de un proceso.

Estado	Definición	
Nuevo	Se solicita al sistema operativo la creación de un proceso, y sus recursos y estructuras están siendo creadas.	(Nuevo)
Zombie	Está listo para iniciar o continuar su ejecución, pero el sistema no le ha asignado un procesador.	(Listo)
Terminado	El proceso está siendo ejecutado en este momento. Sus instrucciones están siendo procesadas en algún procesador	(En ejecución)
Listo	En espera de algún evento para poder continuar su ejecución (aun si hubiera un procesador disponible, no podría avanzar)	(Bloqueado)
Bloqueado	El proceso ha finalizado su ejecución, pero el sistema operativo debe realizar ciertas operaciones de limpieza para poder eliminarlo de la lista.	(Zombie)
En ejecución	El proceso terminó de ejecutarse; sus estructuras están a la espera de ser limpiadas por el sistema operativo.	(Terminado)

3. Completa correctamente el siguiente diagrama de transición entre los estados de un proceso.

LISTA DE RESPUESTAS (sobra 1)

- Listo (2)
- Zombie
- Nuevo (1)
- Bloqueado (4)
- En ejecución (3)
- Terminado (5)



IV – BLOQUEOS MUTUOS Y POLÍTICAS

1. Responde correctamente las siguientes cuestiones:

- a) Es la condición en la cual los procesos reclaman control exclusivo de los recursos que piden. (condición de *exclusión mutua*)
- b) Es la condición en la cual los procesos mantienen los recursos que ya les han sido asignados mientras esperan por recursos adicionales. (condición de *espera por*)
- c) Es la condición en la cual los recursos no pueden ser extraídos de los procesos que los tienen hasta su completa utilización. (condición de *no apropiatividad*)
- d) Es la condición en la cual hay una cadena circular de procesos en la que cada uno mantiene a uno o más recursos que son requeridos por el siguiente proceso de la cadena. (condición de *espera circular*)
- e) Es el nombre con el cual se le refiere a las políticas que otorgan los recursos lo antes posible cuando son solicitadas. (liberales)
- f) Es el nombre con el cual se le refiere a las políticas que controlan más la asignación de recursos. (conservadoras)

2. Relaciona las siguientes definiciones con sus conceptos referente a las diferentes estrategias que pueden ser empleadas para enfrentar situaciones de bloqueo

Estrategia	Definición	
Evación	El sistema permite que ocurran los bloqueos, pero busca determinar si ha ocurrido y tomar medidas para eliminarlo.	(Detección y recuperación)
Detección y recuperación	Se centra en modelar el comportamiento del sistema para que elimine toda posibilidad de que se produzca un bloqueo. Resulta en una utilización subóptima de recursos	(Prevención)
Prevención	Busca imponer condiciones menos estrictas que en la prevención, para intentar lograr una mejor utilización de los recursos. Si bien no puede evitar todas las posibilidades de un bloqueo, cuando éste se produce busca evitar sus consecuencias.	(Evación)

V – ALGORITMOS Y PLANIFICACIÓN

1. Responde correctamente las siguientes cuestiones

- a) Se refiere a cómo un sistema operativo determina el orden en que ira cediendo el uso del procesador a los procesos que lo vayan solicitando. (planificación de procesos)
- b) Menciona los tres tipos principales de planificación. (a largo plazo, a mediano plazo, a corto plazo)
- c) Es el tipo de planificación en donde se decide que procesos serán los siguientes en ser iniciados. (a largo plazo)
- d) Es el tipo de planificación en donde se decide cuales procesos es conveniente bloquear en determinado momento. (a mediano plazo)
- e) Es el tipo de planificación en donde se decide como compartir momento a momento al equipo entre todos los procesos que requieren de sus recursos. (a corto plazo)

2. Relaciona ambas columnas

Algoritmo	Definición	
Ronda egoísta (SRR, selfish round robin)	Es el tipo de algoritmo en donde cada proceso se ejecuta en el orden en que fue llegando, y hasta que suelta el control.	Primero llegado, primero servido (FCFS)
Ronda (Round Robin)	Es el tipo de algoritmo en donde se toma en cuenta el tiempo que requieren los procesos que forman la lista, y es entonces en donde se ejecuta el más corto de los presentes	El proceso más corto a continuación (SPN, shortest process next)
Primero llegado, primero servido (FCFS)	Es el tipo de algoritmo en donde cada proceso puede ejecutarse por un solo quantum (q), en caso de que no a terminado de ejecutarse al final de su quantum será interrumpido y puesto al final de la lista de procesos listos, para que espere su turno nuevamente.	Ronda (Round Robin)
El proceso más corto a continuación (SPN, shortest process next)	Es el tipo de algoritmo en donde los procesos que ya han pasado tiempo ejecutando se ven favorecidos a los recién llegados. En este algoritmo los nuevos procesos no son programados directamente para su ejecución, si no que se les forma en la cola de procesos nuevos.	Ronda egoísta (SRR, selfish round robin)

VI – INTERPRETE DE COMANDOS

1. Comandos Básicos en Terminal Windows

Comando	Funcionalidad
CD	Muestra el nombre del directorio actual o cambia a otro directorio.
CHDIR	Muestra el nombre del directorio actual o cambia a otro directorio.
CLS	Borra la pantalla.
CMD	Inicia una nueva instancia del intérprete de comandos de Windows.
COPY	Copia uno o más archivos a otro lugar (en otra ubicación).
DATE	Muestra o establece la fecha.
DEL	Elimina uno o más archivos.
DIR	Muestra una lista de archivos y subdirectorios en un directorio.
ECHO	Muestra mensajes, o activa y desactiva el eco.
EXIT	Sale del programa cmd.exe (intérprete de comandos).
HELP	Proporciona información de ayuda para los comandos Windows.
MKDIR	Crea un directorio.
MOVE	Mueve uno o más archivos de un directorio a otro.
RD	Elimina un directorio.
RENAME	Cambia el nombre de uno o más archivos.
RMDIR	Elimina (quita) un directorio.
SYSTEMINFO	Muestra las propiedades y la configuración específicas del equipo.
TASKKILL	Termina o interrumpe un proceso o aplicación que se está ejecutando.
TASKLIST	Muestra todas las tareas en ejecución, incluidos los servicios.
TIME	Muestra o establece la hora del sistema.
TITLE	Establece el título de la ventana de una sesión de cmd.exe.
VER	Muestra la versión de Windows.
VOL	Muestra la etiqueta del volumen y el número de serie del disco.
CD	Muestra el nombre del directorio actual o cambia a otro directorio.

2. Comandos Básicos en PowerShell

Comando	Funcionalidad
GET-HELP	Proporciona la información necesaria sobre cmdlets, comandos, funciones, scripts y flujos de trabajo de PowerShell
GET-COMMAND	Muestra la lista de comandos de una función específica
INVOKE-COMMAND	Ayuda a ejecutar un comando o un script de PowerShell, local o remotamente en una o varias computadoras
INVOKE-EXPRESSION	Proporciona una forma de ejecutar un script desde Windows PowerShell
INVOKE-WEBREQUEST	Ayuda a descargar, iniciar sesión y buscar información sobre sitios web y servicios web mientras se trabaja en Windows PowerShell
GET-ITEM	Busca información sobre un elemento en cualquier ubicación determinada
COPY-ITEM	Copiar archivos y directorios en el disco de almacenamiento
REMOVE-ITEM	Elimina elementos como archivos, carpetas, funciones, claves de registro y variable
GET-CONTENT	Ayuda a visualizar el contenido de un archivo de texto sin abrir el archivo
SET-CONTENT	Ayuda a guardar texto en archivos de texto.
GET-VARIABLE	Permite visualizar valores de variables
SET-VARIABLE	Permite asignar, cambiar o restablecer el valor de una variable
GET-PROCESS	Muestra una lista de los procesos que se están ejecutando actualmente.
START-PROCESS	Facilita el inicio de uno o más procesos en la computadora
STOP-PROCESS	Permite detener instancias específicas o todas las instancias de un proceso que se este ejecutando en la computadora.
GET-SERVICE	Muestra información sobre servicios específicos
START-SERVICE	Ayuda a iniciar un servicio en la computadora
STOP-SERVICE	Ayuda a detener un servicio en la computadora

3. Comandos Básicos en Terminal Linux

Comando	Funcionalidad
adduser	Crea una cuenta de usuario
basename	Muestra el nombre del archivo
cat	Muestra el contenido de los archivos
cd	Navega por el árbol de ficheros
chmod	Gestiona los permisos de acceso
cmp	Compara archivos a nivel de byte
cp	Copia archivos o directorios
date	Muestra la hora del sistema
deluser	Cierra la cuenta de usuario
delgroup	Borra grupos de usuarios
df	Muestra el espacio libre en el disco duro
dirname	Muestra la ruta al archivo
du	Muestra el espacio ocupado en disco
file	Muestra el tipo de archivo
free	Muestra la carga de la memoria RAM
groupmod	Configura grupos de usuarios
hostname	Muestra el nombre del host
id	Solicita los ID de usuarios y grupos
ip	Administra interfaces de red
ls	Muestra el contenido del directorio como una lista
mkdir	Crea un directorio
mv	Mueve archivos o directorios
nano	Editor de textos nano
netstat	Consulta el estado de las interfaces de red
passwd	Cambia la contraseña del usuario
ping	Comprueba la conexión de red
ps	Muestra una lista de todos los procesos activos
pwd	Muestra el nombre del directorio
reboot	Reinicia el sistema
rename	Cambia el nombre de archivos
rm	Borra archivo o directorio
rmdir	Borra el directorio
shutdown	Apaga el sistema
su	Trabaja con los permisos de otro usuario
sudo	Ejecuta programas con los permisos de otros usuarios
usermod	Configura las cuentas de usuario
vim	Editor de textos VIM

VII – NAVEGACIÓN POR LINEA DE COMANDOS

1. Trabajando con directorios. Realiza los siguientes ejemplos de comandos

Comando	Sintaxis	Ejemplo
cp	cp [Opciones] archivo_fuente directorio_destino cp [Opciones] archivo_fuente archivo_destino	cp manual_linux_v1 ../.././doc/linux
mv	mv [Opciones] fuente destino	mv manual_linux_v1 manuales/linux mv manual_linux_v1 manual_linux_v1_doc
rm	rm [Opciones] archivos	rm manual_linux_v1 rm -r documentos/
mkdir	mkdir [Opciones] nombre_directorio	mkdir manuales
rmdir	rmdir [Opciones] directorio	rmdir manual
ls	ls [Opciones] [nombre_directorio o archivo]	ls -a ls -l ls -la
cd	cd [directorio]	cd manuales
pwd	pwd	
chmod	chmod [Opciones] [permiso_descripción] archivo	chmod u+rx manual_linux
cat	cat [-opciones] archivos	cat archivo1 cat archivo1, archivo2, archivo3 > todo

2. Ruta absoluta y Ruta relativa

Ruta absoluta: se indica toda la ruta del archivo incluyendo el directorio raíz.

Ejemplo: C:\carpeta1\carpeta2\archivo1.doc

Ruta relativa: se indica la ruta a partir de donde esté en ese momento situado. No se incluye el directorio raíz.

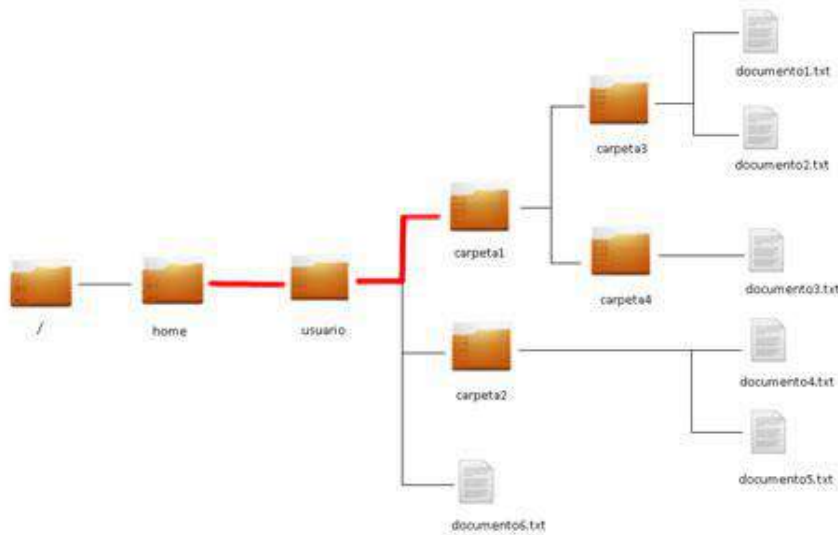
Ejemplo: → Ruta actual: C:\carpeta1

→ Archivo al que queremos acceder: archivo1

→ Carpeta donde se localiza el archivo: carpeta2

→ Ruta relativa: carpeta2\archivo1

Realiza los siguientes ejercicios de ruta absoluta y ruta relativa tomando en cuenta la siguiente imagen



→ Ruta Absoluta

1.- La Ruta absoluta para el archivo *“documento2.txt”*

2.- La ruta absoluta para el archivo *“documento6.txt”*

3.- La Ruta absoluta para el archivo *“documento3.txt”*

4.- La Ruta absoluta para la carpeta *“carpeta2”*

→ Ruta Relativa

1.- La Ruta relativa para el archivo *“documento5.txt”* partiendo desde la carpeta *“usuario”*

2.- La ruta absoluta para la carpeta *“carpeta3”* partiendo desde la carpeta *“home”*

3.- La Ruta absoluta para el archivo *“documento3.txt”* partiendo desde la carpeta *“/”*

4.- La Ruta absoluta para la carpeta *“carpeta4”* partiendo desde la carpeta *“home”*

VIII – EXPLORANDO EL SISTEMA LINUX

1. Jerarquía de sistemas de archivos

Directorio	Descripción
/	<i>Jerarquía primaria</i> , denominado <i>root</i> o la raíz, directorio principal, contenedor del absolutamente todo el sistema de archivos en Linux.
/bin/	Contiene las binarias de comandos indispensables, para que estén disponibles bien sea para una sesión única o para multiusuarios. Se incluyen, por ejemplo, ls, cp, cat, mkdir, rm, entre otras
/boot/	El arranque del sistema.
/dev/	Contiene los accesos hacia los dispositivos. Tanto hardware o virtuales.
/etc/	En este se incluyen los archivos configurativos del sistema. Ha habido controversia respecto al significado de su nombre, pero Interpretaciones más recientes lo denominan como “Configuración de texto editable”.
/etc/opt/	Archivos configurativos de los programas localizados dentro del directorio /opt.
/etc/X11/	Archivos configurativos del X Window System, versión 11.
/etc/sgml/	Archivos configurativos de SGML.
/etc/xml/	Archivos configurativos de XML.
/home/	Contiene los directorios de trabajo de todos los usuarios, excepto el del superusuario (administrador, root). Contiene archivos guardados, ajustes personales, etc. <i>A menudo es instalada en un disco o partición separada</i> . Cada usuario tiene su propio directorio dentro de esta carpeta.
/lib/	Se localizan todas las librerías fundamentales compartidas de los programas instalados incluyendo también las utilizadas por el núcleo.
/media/	Contiene los <i>puntos de montaje</i> de los medios extraíbles de almacenamiento.
/mnt/	Es similar a /media, pero normalmente utilizado por los usuarios. Con el fin de “montar” por ejemplo discos duros y particiones temporales.
/opt/	Contiene información de aplicaciones que no guardan las opciones de configuración en este directorio, es decir, los usuarios comparten la aplicación mas no las opciones de configuración de la misma.
/proc/	Contiene archivos que documentan el núcleo y el estado de sus procesos en específicos momentos.
/root/	Principal directorio del usuario root. Es como el /home pero para el superusuario (administrador) del sistema.
/sbin/	Ejecutables o binarios esenciales para el funcionamiento, comandos y programas exclusivos del administrador del sistema o usuarios que tengan permisos para hacer uso de ellos.

/srv/	Contiene los datos servidos por el sistema.
/tmp/	Contiene los archivos temporales.
/usr/	<i>jerarquía secundaria</i> de datos de usuario; contiene la mayoría de utilidades que son de propósito multiusuario pero que sin embargo son de solo lectura. Esta carpeta incluso puede ser compartida con otros usuarios de red local.
/usr/bin/	Comandos binarios no administrativos para todos los usuarios.
/usr/include/	Archivos de inclusión estándar.
/usr/lib/	Conjunto de librerías o binarios compartidos. Nunca hay dos librerías idénticas en el mismo sistema, lo cual optimiza el uso de memoria y provee mayor orden.
/usr/sbin/	Binarios que no son esenciales; por ejemplo, <i>daemons</i> para tener varios servicios de red.
/usr/share/	Contiene los datos que son compartidos pero que son independientes de la arquitectura.
/usr/src/	Contiene los códigos fuentes de algunas aplicaciones.
/usr/X11R6/	Directorio relacionado al entorno gráfico.
/usr/local/	<i>Jerarquía terciaria</i> para datos locales, es decir, específicos a este host.
/var/	Contiene los archivos variables del sistema, como logs, bases de datos, correo electrónico.
/var/cache/	Similar a /tmp, contiene la memoria cache de algunas aplicaciones.
/var/crash/	Contiene información referente a errores o caídas del sistema.
/var/games/	Es un directorio que no es imprescindible y su propósito es almacenar información referente a los juegos del sistema.
/var/lock/	Se localizan los archivos que tienen el estado de los recursos en uso.
/var/log/	Archivos de <i>logs</i> .
/var/mail/	Archivos de mensajes de usuarios, correos similares.
/var/opt/	Contiene los datos que pueden ser variables del directorio /opt.
/var/run/	Acceso a información desde el último arranque del sistema. Por ejemplo, usuarios conectados en ese instante, o demonios en ejecución.
/var/spool/	Incluye las tareas que están en espera para procesar. Por ejemplo, correos no leídos o colas de impresión.
/var/spool/mail/	Localización de los correos de usuarios desaprobados.
/var/tmp/	Contiene archivos temporales, su diferencia con /tmp es el hecho de no borrarse al reiniciar el sistema.

IX – MANIPULANDO ARCHIVOS Y DIRECTORIOS

1.- Wildcard y ejemplos

Los wildcards son caracteres que representan un conjunto o rango. Con ellos se construyen las expresiones regulares. Suelen denominarse también caracteres comodines.

Algunos de los wildcards más utilizados son:

- * El asterisco representa cualquier conjunto de caracteres de un nombre de archivo.
- ? La interrogación representa un solo carácter (sea el que sea),
- [] Los corchetes representan clases de caracteres. Por ejemplo, para buscar cualquier carácter que represente un dígito pondríamos [0-9]
- Como hemos visto en el ejemplo anterior, un guion indica un rango de caracteres.
- ~ La virguilla (sí, se llama así) al principio de una palabra se traduce como el nombre de tu directorio home. Si le añades el nombre de un usuario, se traduce como la ruta del directorio home de ese user.

Ejemplos:

- # **ls *.php** → Lista todos los archivos php.
- # **rm *.gz** → Elimina todos los archivos GZIP.
- # **cat backup*** → Muestra el contenido de todos los archivos que empiecen por “backup”.
- # **ls test?** → Lista todos los archivos que empiecen por test y tengan un solo carácter adicional (mostraría test4 pero no test15).

X – TRABAJANDO CON COMANDOS LINUX

1.- Documentación de comandos Linux

<i>Comandos</i>	<i>Descripción</i>
ls	Lista los ficheros de un directorio en concreto
ls -l	Lista también las propiedades y atributos
ls -a	Lista ficheros incluidos los ocultos de sistema
cd nom_directorio	Cambia de directorio
more nom_fichero	Muestra el contenido de un fichero de forma paginada
ls -la more	Lista los ficheros de forma paginada
mv [ruta1]fichero1 [ruta2]fichero2	Mueve y/o renombra un fichero
rm archivo o directorio	Elimina archivos o directorios
rm -R directorio	Borra un directorio recursivamente
cp archivo1 archivo2	Realizar una copia de un fichero
tail nom_archivo	Muestra las últimas líneas de un archivo de forma estática
tail -f nom_archivo	Muestra las últimas líneas de El fichero de manera dinámica
head -numero nom_archivo	Muestra las primeras número líneas de un fichero
mkdir nom_directorio	Crea un directorio
rmdir nom_directorio	Elimina un directorio
chmod xxx nom_fichero	Cambiar los permisos de acceso de un fichero
chown usuario fichero/directorio	Cambia el propietario de un fichero directorio
ps aux	Muestra una lista de los procesos activos
kill -x(de 0 a 9) PID	Elimina un proceso vía n PID que lo identifica
mount	Vemos el listado de dispositivos montados
mount /dev/fd0/mnt/floppy	Montaje de la disquetera
mount /mnt/chrom	Punto de montaje del CD rom
umount	Desmontar los puntos de montaje anteriores
hostname	Visualiza el nombre de la máquina
fsck	Escanea el disco
init 0	Apaga la máquina de manera correcta
init 6	Reiniciar la máquina

<code>compress/uncompress</code>	Comandos de compresión de archivos .z
<code>gunzip nombre_archivo.gz</code>	Descompresión de archivos .gz
<code>tar xvf fichero.tar</code>	Descomprimir archivos .tar
<code>top</code>	Lista procesos de CPU
<code>find / -name nom -print</code>	Encuentra ficheros según patrón
<code>pwd</code>	Visualizar el directorio actual
<code>grep 'cadena' archive</code>	Muestra las líneas del archivo que contiene la cadena
<code>date</code>	Da la hora y la fecha del sistema
<code>cal</code>	Muestra el calendario del mes actual
<code>clear</code>	Borra la pantalla
<code>who</code>	Informa de los usuarios conectados
<code>whoami</code>	Más información sobre nuestra máquina
<code>finger</code>	Información más completa que who
<code>su</code>	Guion como root (necesario passwd)
<code>su nombre_usuario</code>	Estando como root entramos como otro usuario
<code>fdisk</code>	Gestión de particiones

2.- Creación de alias

Creación de alias temporales

Lo que debemos hacer es escribir la palabra alias en la terminal. Después tendremos que utilizar el nombre que queremos usar para ejecutar un comando. Esto irá seguido por el signo '=' y la llamada al comando que queramos utilizar.

La sintaxis para seguir es la siguiente:

```
1 alias nombreAlias="tu comando personalizado aquí"
```

Este sería un ejemplo real:

```
1 alias htdocs="cd /opt/lampp/htdocs"
```

Realiza las siguientes creaciones de alias:

1. alias disco= 'df -h'
2. alias actualizarsistem= 'sudo apt update && sudo apt upgrade'
3. alias pingredlocal= 'ping Dirección IP Local' [Ejem. 192.168.0.6]
4. alias pinggoogle= 'ping Google.com'

Ejecuta los comandos con el nombre de alias y verifica su funcionamiento.

Creación de alias permanentes

Para mantener los alias entre sesiones, vas a tener que guardarlos en el **archivo de perfil para la configuración de shell de tu usuario**. Estos podrían ser:

Bash → ~/.bashrc

ZSH → ~/.zshrc

Fish → ~/.config/fish/config.fish

La sintaxis que hay que utilizar en este caso, es la misma que cuando creamos uno temporal. La única diferencia viene del hecho de que esta vez lo guardaremos en un archivo. Entonces, por ejemplo, en bash, puedes abrir el archivo .bashrc con tu editor favorito:

```
1 vim ~/.bashrc
```

Dentro del archivo, busca un lugar en el archivo donde guardar los alias.

Realiza las siguientes creaciones de alias:

1. #Mis alias personalizados

alias home= 'cd /home'

alias actualizarsistem= 'sudo apt update && sudo apt upgrade'

alias pingfb = 'ping Facebook.com'

alias pingsiase= 'ping www.uanl.mx/enlinea/'

XI – REDIRECCIONAMIENTO I/O

1.- Entrada, Salida y Error Estándar

Archivos abiertos se asocian a un número de identificador, llamados files descriptor. Estos pueden ser utilizados en argumentos de redirección. Los tres archivos estándar tienen siempre el mismo número asignado:

Nombre	Descriptor
Entrada estándar	0
Salida Estándar	1
Error Estándar	2

Ejemplos:

- 1.- Para redireccionar el error estándar a un archivo
\$ programa 2> archivo.txt
- 2.- Para combinar la salida del error estándar con la salida estándar
\$ programa > archivo 2>&1
- 3.- Para guardar ambas salidas de flujo:
\$ programa > stdout.txt 2> stderr.txt

XII – VIENDO COMO EL INTERPRETE DE COMANDOS

El intérprete de comandos permite la comunicación entre el sistema y el usuario. Existen funciones, como la automatización de tareas, que se resuelven mejor desde la línea de comandos (CLI Command Line Interface).

1.- Expansión

Expansión de tilde

Las apariciones de la virgulilla (o tilde de la ñ) dentro de una línea, que no se encuentren entrecomilladas, se expanden de la siguiente manera:

Variables	Valor
~	Se expande al valor de la variable [HOME]
~login	Si "login" es un nombre de usuario del sistema, se expande a la ruta absoluta del directorio de inicio de sesión de ese usuario. Si no, no se expande.

Con base al siguiente script `script_var-shell.sh` realice las siguientes actividades:

```
#!/bin/sh
echo \@=$@
echo \*=$*
echo \$0=$0
echo \$1=$1
echo \$2=$2
echo Cambio parametros posicionales
set uno dos tres
echo \$1=$1
echo \$2=$2
echo Desplazo
shift
echo \$1=$1
echo \$2=$2
echo \$-=$-
echo $#=$#
echo \$?=$?
firefox &
ps w
echo \$=$$
echo \$!=\$!
```

TAREAS

Comente la línea del script `script_var-shell.sh`, donde aparece la palabra "firefox". Ejecute los siguientes comandos observando el valor de las variables posicionales.

```
./script_var-shell.sh ~ ~root  
./script_var-shell.sh ~noexiste ~dit
```

Expansión de ruta

Los campos que incluyan los caracteres `*`, `?` y `[` (asterisco, interrogación y apertura de corchetes) no entrecomillados serán sustituidos por la lista de ficheros que cumplan ese patrón. Si no hay ningún fichero con ese patrón no se sustituye nada.

TAREAS

Utilice el script `script_var-shell.sh`, visto en el apartado 3.3, pero modifíquelo eliminando las líneas con la palabra "firefox", para evitar que moleste en las siguientes pruebas. Ejecute los siguientes comandos observando el valor de la variable especial `@`:

```
./script_var-shell.sh s*_for?.sh  
./script_var-shell.sh s*_for*.sh  
./script_var-shell.sh s*_exp*.sh  
./script_var-shell.sh s*_exp*[12].sh
```

```
./script_var-shell.sh s*_e*.sh
```

Expansión aritmética

El formato para realizar una expansión aritmética es el siguiente:

```
$(expresión)
```

TAREAS

Mire el contenido del script `script_expansion3.sh`, que deberá contener lo siguiente:

```
script_expansion3.sh
```

```
#!/bin/sh
VAR=1
VAR=$VAR+1
echo $VAR
RES1=$(( $VAR ))+1
echo $RES1
VAR=1
RES2=$((VAR+1)) #VAR no necesita $
echo $RES2
VARb=b
echo $(( $VARb+1 )) #VARb necesita $
```

Compruebe que dispone del permiso de ejecución. Invóquelo mediante el comando:

```
./script_expansion3.sh
```

Analice el resultado.

Expansión de llaves

La **expansión de llaves** es un mecanismo por el cual se pueden generar cadenas arbitrarias.

Los patrones para la expansión de llaves tienen la forma de un preámbulo opcional, seguido y encerrado entre llaves de, o bien una secuencia de cadenas separadas por comas, o bien de expresiones de rango, y acaba con un epílogo opcional.

Dentro de unas mismas llaves no pueden mezclarse secuencias y rangos, o bien una cosa o la otra.

En el ejemplo siguiente, se especifica una secuencia con preámbulo (prefijo) y epílogo (sufijo)

```
$ echo x{aa,bb,cc}y
xaay xbbby xccy
```

TAREAS Realice las siguientes expansiones y analice los resultados.

```
$ echo x{0..16}y
$ echo x{00..16..2}y
$ echo x{00..16..2}y
$ echo x{a..p..2}y
$ echo {aa,pp,ff}{1..3}
```

Expansión de parámetros

El formato general para incluir una expansión de variables o parámetros, como se ha visto en apartados anteriores es:

```
${PAR}
```


Las llaves pueden omitirse, salvo cuando se trate de un parámetro posicional con más de un dígito o cuando se quiera separar el nombre de la variable de otros caracteres. Por ejemplo:

```
echo $PAR #puede omitirse
echo ${10} #no puede omitirse
${PAR}TEXTO #no se omite
```

<code>\${PAR:-alternativo}</code>	Valor de la variable. Si la variable no tiene ningún valor, la construcción se sustituye por el valor <code>alternativo</code> .
<code>\${PAR:=alternativo}</code>	Ídem al anterior, pero asignando el valor <code>alternativo</code> a la variable.
<code>\${PAR%sufijo}</code>	Elimina el <code>sufijo</code> más pequeño del valor de la variable. <code>sufijo</code> es un patrón como los utilizados en la expansión de ruta. Si en vez de <code>%</code> se pone <code>%%</code> se elimina el <code>sufijo</code> más grande.
<code>\${PAR#prefijo}</code>	Elimina el <code>prefijo</code> más pequeño del valor de la variable. <code>prefijo</code> es un patrón como los utilizados en la expansión de ruta. Si en vez de <code>#</code> se pone <code>##</code> se elimina el <code>prefijo</code> más grande.

Los siguientes scripts muestran un posible ejemplo de ambas construcciones:

`script_expansion1.sh`

```
#!/bin/sh
VAR=1
echo $VAR
unset VAR
echo ${VAR:-2}
echo $VAR
FICH=fichero.c
echo ${FICH%.c}.o
```

script_expansion2.sh

```
#!/bin/sh
VAR=1
echo $VAR
unset VAR
echo ${VAR:=2}
echo $VAR
FICH=/usr/bin/prueba
echo ${FICH##*/}
```

TAREAS

Mire el contenido de los scripts anteriores en su sistema, invóquelos y analice los resultados.

2.- Comillado

En Bash las comillas se utilizan para delimitar cadenas que tienen espacios (o tabuladores, o saltos de línea). Si un fichero tiene un nombre con espacios, debe entrecomillarse.

Tipos de comillas:

Comilla simple (apóstrofe ')

La comilla simple delimita cadenas en las que no se realiza interpolación de variables. Es decir, no se substituyen por su contenido.

Comilla doble (")

Delimita cadenas en las que se realiza interpolación de variables. Se substituyen por su contenido.

TAREAS

Mire los resultados.

```
$ color_favorito=rojo
$ echo 'Mi color favorito es $color_favorito'
$ echo "Mi color favorito es $color_favorito"
```

XIII – ATAJOS AVANZADOS DE TECLADO

1.- Edición de línea de comandos

Completa correctamente con la tecla correspondiente para el atajo

- a) La tecla (tab) muestra las opciones de completar el comando actual o lo completa directamente.
- b) Ctrl + (C) se usa para salir de un proceso.
- c) Ctrl + (Z) se usa para mandar el proceso actual al fondo.
- d) Ctrl + (D) cierra automáticamente la terminal actual.
- e) Ctrl + (L) limpia la terminal haciendo lo mismo que el comando clear.
- f) Ctrl + (E) manda el cursor al final de la línea.
- g) Ctrl + (A) manda el cursor al inicio de la línea.
- h) Ctrl + (U) limpia todo lo que escribiste.
- i) Ctrl + (K) limpia todo después del cursor hasta el final de la línea.
- j) Ctrl + (W) elimina la palabra que este antes del cursor.
- k) Ctrl + (Y) pega el texto que hayas borrado con los anteriores atajos.

2.- Uso y expansión de historial

Completa correctamente con la tecla correspondiente para el atajo

- a) Ctrl + (P) te deja ver el comando anterior y puedes presionarlo varias veces para ver los anteriores.
- b) Ctrl + (N) para complementar el anterior te permite ver el siguiente comando si te pasaste retrocediendo.

c) Ctrl + (R) te permite hacer una búsqueda en todo el historial de comandos usados.

XIV – PERMISOS UNIX

1.- Permisos de usuario, grupo y propietarios en archivos y directorios

1 - Relaciona las siguientes descripciones de los permisos en archivos

Letra	Descripción	
r	Permiso de ejecutar el contenido del archivo como programa	(x)
w	Permisos de leer la información almacenada en el archivo	(r)
x	Permisos para escribir información en el archivo o sobre escribir información	(w)

2 - Relaciona las siguientes descripciones de los permisos en directorios

Letra	Descripción	
r	Permiso para crear, borrar o renombrar archivos dentro del directorio	(w)
w	Permiso para cambiar o usar el directorio como parte Intermedia da un archivo	(x)
x	Permiso para ver el interior del directorio	(r)

3 - Define el orden de a quien se le asignan los permisos.

Orden	Usuario	
1	Grupo de usuarios definido.	(2)
2	Otros, todos los usuarios particulares del sistema	(3)
3	Propietario, quien creo el archivo o directorio	(1)

4 - Escribe que permisos daría el siguiente comando

chmod 644: (el propietario puede leer y escribir, los demás solo pueden leer)

chmod 755: (El propietario puede leer, escribir y ejecutar el archivo mientras los demás leer y escribir solamente)

chmod 700: (El propietario tiene control total del archivo y los demás no tienen ningún acceso)

chmod 777: (Todos los usuarios pueden leer, escribir y ejecutar el archivo o directorio)

2.- Permisos especiales

1 - Escribe el permiso especial correspondiente con la descripción

SUID = Cuando se activa sobre un fichero significa que el que lo ejecute va a tener los mismos permisos que el que creó el archivo. Esto es útil en algunas ocasiones, aunque hay que utilizarlo con cuidado, ya que puede acarrear problemas de seguridad

SGID = Todo archivo que lo tenga activo, al ser ejecutado, tendrás los privilegios del grupo al que pertenece.

Sticky bit = Cuando se le asigna a un directorio, significa que los elementos que hay en ese directorio solo pueden ser renombrados o borrados por su propietario o bien por root. El resto de los usuarios que tengan permisos de lectura y escritura, los podrán leer y modificar, pero no borrar.

3.- Cambio de identidad

1 - Responde correctamente las siguientes cuestiones

a) Ubicación del archivo donde todos los usuarios son almacenados (/etc/passwd)

b) A que hace referencia la x después del usuario en el archivo de usuarios (A la contraseña encriptada)

c) Menciona los tres tipos de usuarios en un entorno unix (Root, especiales y normales)

2 - Escribe los comandos correspondientes para lo que se pide

a) Crear un usuario llamado Appa

(useradd Appa)

b) Modificar el usuario Appa para que tenga el comentario “Mascota del avatar”

(usermod -c “Mascota del avatar” Appa)

c) Establecer una contraseña para el usuario Appa

(passwd Appa)

d) Modificar el nombre de la cuenta Appa a Momo

(usermod -l Appa Momo)

f) Cambiar al usuario Momo

(su Momo)

g) Borrar el usuario Momo

(userdel Momo)

XV – PROCESOS EN SISTEMAS LINUX

1.- visualización de procesos

Compara las imágenes con las diferentes formas de visualizar procesos y especifica que comando se usó para cada una.

1. top
2. ps
3. pstree

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	731	1.1	2.2	889464	91268	tty7	SsL+	13:22	0:04	/usr/lib
root	732	0.0	0.0	2812	1684	tty1	Ss+	13:22	0:00	/sbin/ag
root	1716	0.1	0.1	12088	7988	pts/0	Ss	13:24	0:00	zsh
root	1734	0.0	0.0	9772	3580	pts/0	S	13:24	0:00	zsh
root	4086	0.0	0.0	2852	992	pts/0	Sl	13:24	0:00	/home/ha
root	4086	0.0	0.0	9572	3348	pts/0	R+	13:29	0:00	ps -a -u

1

```

# total, 1 running, 179 sleeping, 0 stopped, 0 zombie
# 0 st, 0.0 sy, 0.0 ni, 98.5 id, 0.0 wa, 0.0 hi, 0.2 si, 0.
# swab: 3962.2 total, 1397.0 free, 1166.4 used, 1338.8 buff/cache
# Mem: 1874.5 total, 1874.5 free, 0.0 used, 2499.0 avail Mem

# PID USER      SH  NI    VIRT    RES    SHR     S    %CPU  %MEM     time+
#----
1137 hayabusa 20  0 370744 74532 50336 S    1.0  1.3  0:10.17
1137 hayabusa 20  0 1185972 25324 19736 S    0.3  0.6  0:01.00
2907 hayabusa 20  0 3747984 363884 51564 S    0.3  0.1  0:33.16
4332 hayabusa 20  0 9956 3696 3992 R    0.2  0.1  0:00.01
1 root      0  0 98680 18388 7786 S    0.0  0.3  0:02.76
2 root     0  0 0 0 0 S    0.0  0.0  0:00.02
3 root     0 -20 0 0 0 T    0.0  0.0  0:00.00
4 root     0 -20 0 0 0 T    0.1  0.0  0:00.00
6 root     0 -20 0 0 0 T    0.0  0.0  0:00.00
7 root     0  0 0 0 0 S    0.1  0.0  0:00.10
8 root     0  0 0 0 0 T    0.0  0.0  0:00.72
9 root     0 -20 0 0 0 T    0.1  0.0  0:00.00
10 root    20  0 0 0 0 S    0.5  0.0  0:00.00
11 root    20  0 0 0 0 S    0.5  0.0  0:00.00
12 root    20  0 0 0 0 S    0.5  0.0  0:00.04
13 root    20  0 0 0 0 S    0.1  0.0  0:00.31
14 root    rt  0 0 0 S    0.5  0.0  0:00.00
15 root    20  0 0 0 0 S    0.5  0.0  0:00.00
16 root    20  0 0 0 0 S    0.5  0.0  0:00.00
17 root    rt  0 0 0 S    0.5  0.0  0:00.00
18 root    20  0 0 0 0 S    0.5  0.0  0:00.04

```



2.- Control de procesos

Con el comando top se puede interactuar con los procesos. Relaciona el comportamiento con la letra que lo causa

Sale del programa:

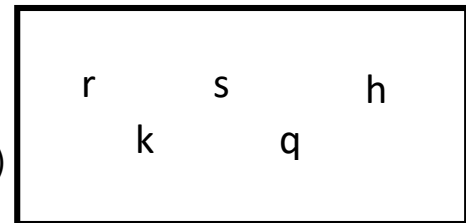
Muestra la pantalla de ayuda: (q)

Pide el id de un proceso y una señal y la manda al proceso: (h)

Pipe el id de un proceso y muestra la información del proceso: (k)

Cambia la prioridad de un proceso: (r)

Cambia el número de segundos entre actualizaciones: (s)



3.- Señales

Relaciona la señal con su descripción

<i>Letra</i>	<i>Descripción</i>	
INT	Fuerza el proceso que pare de ejecutar; sin dar oportunidad al cierre correcto	(KILL)
TERM	Enviado por el kernel cuando se ejecuta un logout o se desconecta un modem	(HUP)
KILL	Para la ejecución de un proceso, Enviada por el kernel cuando presionas Ctrl+C en una terminal	(INT)
TSTP	Detiene temporalmente un proceso, enviada por el kernel al presionar Ctrl+Z en una terminal	(TSTP)
HUP	Pide al proceso que cierre correctamente	(TERM)

XVI – ENTORNO LINUX

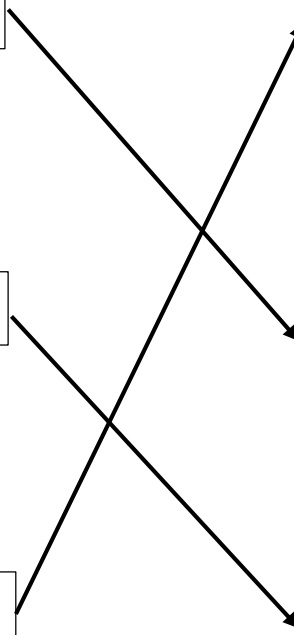
1.-Examinar y modificar el entorno

Relaciona la imagen del entorno con su nombre.

Gnome

Deepin

KDE




2.- Edición de archivos con editor de texto

Relaciona el editor de texto con su nombre

1- nano

2- vim

2



A screenshot of the nano text editor. The window title is 'test.txt'. The content shows lines 1 through 11, each with the word 'test' followed by a number. The status bar at the bottom indicates 'test.txt 1:1 All'.

2



A screenshot of the vim text editor. The window title is 'test.txt'. The content shows lines 1 through 10, each with the word 'test' followed by a number. The status bar at the bottom displays a list of commands: Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, and Go To Line.