Caracteristicas de GNU/Linux

Es multiusuario, multitarea, altamente portable, gratuito, libre y opensource. Posee diversos interpretes, permite el manejo de usuarios y permisos. Todo es un archivo.

GNU = Es un sistema operativo basado en unix creado para ser libre. G Not Unix.

Multitasking = para tener varios trabajos listos para ejecutar, deben guardarse en memoria, requiriendo algun tipo de gestion de memoria, si varios trabajos estan listos para su ejecucion, el procesador elije cual ejecutar primero y el orden de ejecucion de los siguientes para realizar todos los trabajos en simultaneo.

Posix = Es una interfaz de sistema operativo portable, regida por estandares de la IEEE con el objetivo de facilitar la interoperabilidad del sistema operativo.

Una distribucion de un sistema operativo es un conjunto de aplicaciones que brindan mejoras y servicios adicionales. (Debian, SuSe, Red Hat, Slackware)

Estructura de GNU/Linux

Posee tres componentes fundamentales: El kernel, shell y fylesystem.

Kernel: Administra la cpu, la memoria y la E/S. Monolitico hibrido. Monolitico porque drivers y codigo se ejecutan en modo privilegiado e hibrido porque carga y descarga funcionalidades mediante modulos). Se encarga de que el sw y el hw trabajen juntos. Version estable actual 5.8.x. Se puede tener mas de un kernel instalado en la misma maquina. Se encuentra en el fylesystem dentro del directorio /boot.

Shell: Es el medio de comunicacion entre el sistema operativo y el usuario. Ejecuta programas a partir del ingreso de comandos. Cada usuario puede tener un shell, es programable y personalizable. Actua como interfaz, lee entradas del usuario y las traduce en instrucciones que el sistema es capaz de entender y ejecutar. (sh/ksh/csh). Los comandos propios y externos se ubican en /usr/bin.

FileSystem: Es la forma en que se organizan y administran los archivos en un sistema de computo. El adoptado por GNU Linux es el extended(v2,v3,v4). Directorios mas importantes segun FHS: /, /home, /var, /etc, /bin, /dev, /usr.

Particiones

Es la forma de dividir logicamente el disco fisico, (fat, ntfs, ext).

Particion primaria = Division cruda del disco (maximo 4 por disco). Se almacena su info en el MBR..

Particion extendida = Contiene unidades logicas en su interior. Solo puede haber una particion de este tipo en el disco. No se le define un tipo de FileSystem

Particion logica = Ocupa la totalidad o una parte de la particion extendida. Se le define un tipo de FileSystem.

Se necesita como minimo una particion para instalar GNU Linux (/ ). De tipo ext.

/home espacio libre despues de crear otras particiones.

/boot espacio para guardar la informacion del arranque de la maquina.

/etc espacio para guardar la informacion de las configuraciones.

/swap espacio para precargar informacion que luego ira a la ram.

Para particionar discos se utiliza un particionador de discos, hay destructivos(fdisk) y no destructivos (flips, gparted).

Arranque

El BIOS es un sw de bajo nivel que se halla en la motherboard. Cuando arranca la computadora el BIOS se ejecuta, realizando el Power-On Self-Test, luego se lee el primer sector del disco de inicio llamado MBR.

UEFI = Unified Extensible Firmware Interface. Define la interfaz entre el gestor de arranque y el firmware de manera que sea mas sencilla y comprensible. Aporta mayor velocidad en el arranque de los equipos.

MBR = Master Boot Record. Es el primer sector del disco. Alli contiene un codigo de arranque denominado MBC (Master Boot Code), y una marca de 2 bytes que indica su presencia, o bien contiene la tabla de particiones.

GPT = GUID Partition Table. Soluciona las limitaciones del MBR con respecto a la cantidad de particiones que se pueden crear. Se pueden crear hasta 128 particiones. Utiliza un modo de direccionamiento logico en vez del modo cilindro-cabeza-sector del MBR.

Gestor de Arranque = Es un pequeño programa que arranca el sistema operativo. En Linux se puede optar por distintos gestores de arranque (LILO, GRUB, GAG).

BOOTSTRAP = -1 Se ejecuta el codigo de la BIOS -2 El hw lee el sector de arranque -3 Se carga el gestor de arranque -4 Se carga el kernel

SHUTDOWN = Se notifica a los usuarios del shutdown y se bloquea el sistema de interrupciones para que solo sea accesible por el usuario *root*. Luego se envia la señal SIGTERM a todos los procesos no definidos en *inittab* para el siguiente runlevel. Se envia la señal SIGKILL para que los procesos que no hayan terminado previamente ahora lo hagan. Luego envia una señal a *init* para que cambie su nivel de ejecucion y concluya su cometido segun el nivel.

SYSTEM V

INIT = es ejecutado por el kernel. Se utiliza para cargar todos los subprocesos necesarios para el correcto funcionamiento del SO.

RunLevel = Es un numero que condiciona el modo de arranque de Linux. Cada numero (0..6) es responsable de iniciar o detener una serie de servicios. Se encuentran definidos en /etc/inittab. (id:runlevel:action:process)

Script RC = Ejecuta comandos para iniciar y detener servicios del sistema que correspondan al runlevel requerido. Se almacenan en /etc/init.d

Insserv = Es una herramienta de bajo nivel que determina el orden de los scripts que luego iniciara usando las cabeceras a los scripts.

Upstart = Permite la ejecucion de trabajos en forma asincronica a traves de eventos.

System V es sincronico mientras que Upstart es event-based.

Los ScriptsRC del System V en el Upstart son reemplazados por trabajos, que son scripts

de texto plano que define las tareas a ejecutar ante determinados eventos. Se encuentran en /etc/init.conf

Systemd = Es un sistema que centraliza la administracion de demonios y librerias del Sistema que mejora el paralelismo de booteo.

Activacion por Socket = cuando el socket recibe una conexión spawnea el servicio y le pasa el socket.

CGROUP = herramienta que permite organizar un grupo de procesos relacionados de forma jerarquica, limitando el uso de recursos y posibilitando el tracking bajo el subsistema de cgroups.

PROCESOS: es un programa en ejecucion, todos tienen un atributo PID que es el numero que lo identifica en el sistema. Todos los procesos lo tienen a menos que se utilice el systemd, que hace tracking mediante cgroups.

pipe: la finalidad del pipe es conectar la stdin de un comando con las stdout del siguiente, ...

SHELL = es un interprete de comandos interactivo. En sistemas \*nix es configurable, proveen estructuras de control que permiten programar shell scripts.

Shell Scripting = programacion de scripts con el fin de crear aplicaciones interactivas que automatizan tareas. (sh, bash, dash, csh, ksh). Es extremadamente simple para crear procesos y manipular salidas. Ademas es independiente de la plataforma. Se puede probar en el interprete interactivo.

Bash: soporta strings y arrays, los nombres son case sensitive. No es fuertemente tipado, es un "untyped language" porque todos los tipos de datos son strings.

Shebang: es una linea que especifica el interprete que se utilizara para ejecutar el script, en nuestro caso: #!/bin/bash

$0 contiene la invocacion al script

$1, $2, $3, ... contiene cada uno de los argumentos

$# contiene la cantidad de argumentos recibidos

$\* contiene la lista de todos los argumentos

$? contiene en todo momento el valor de retorno del ultimo comando ejecutado

*echo* imprime un string

*break [n]* corta la ejecucion de n niveles de loops

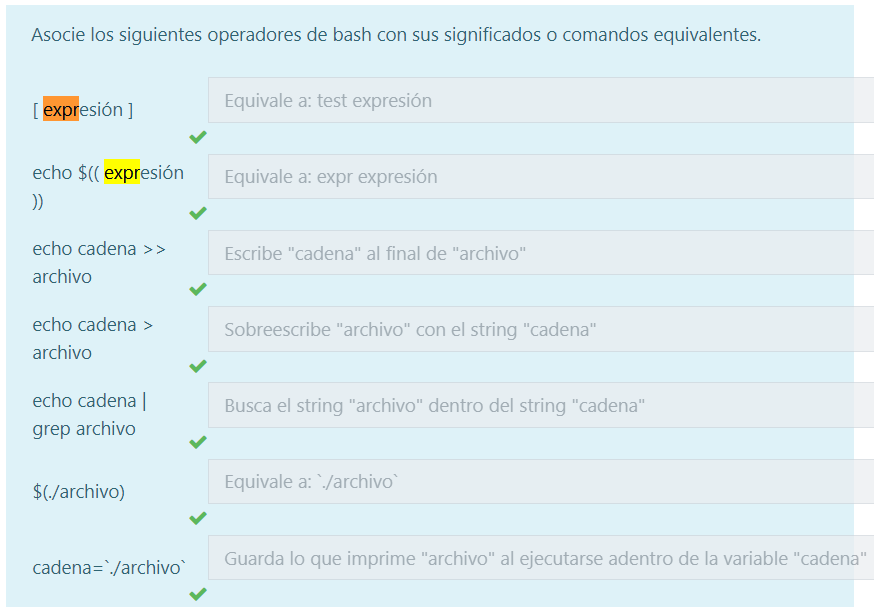
*continue [n]* salta a la siguiente iteracion del n-esimo loop que continue esta instruccion

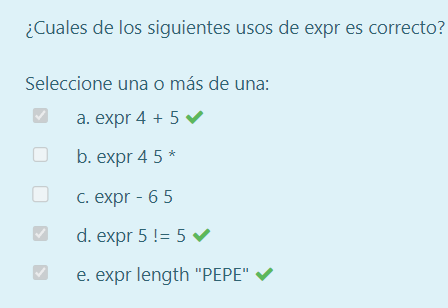
*exit* causa la terminacion de un script devolviendo un valor entre 0 y 255

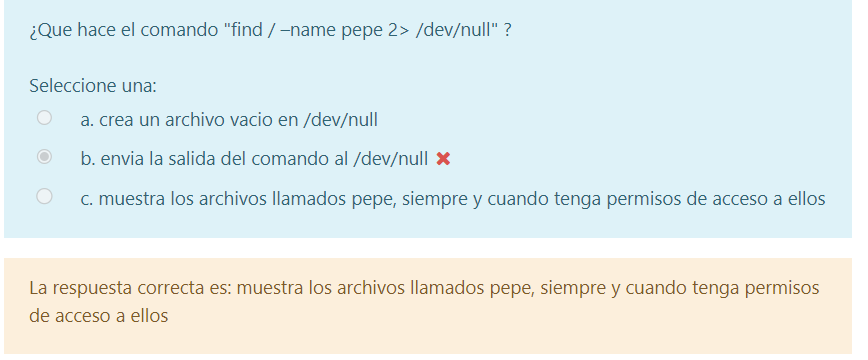
*return* devuelve un valor desde una funcion entre 0 y 255

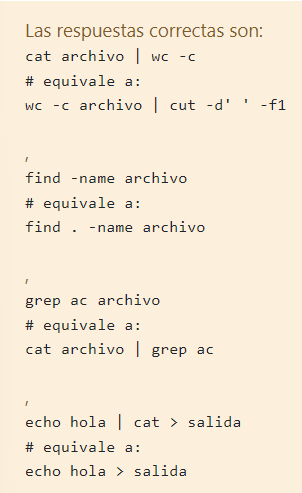
*local* variable local a una funcion (las variables por defecto son globales)

*export* para exponer una variable global a los procesos hijos

**







COMANDOS IMPORTANTES SHELL

cat: mostrar y concatenar texto de uno o mas ficheros.

more: muestra en pantalla y tambien permite la navegacion a traves de la terminal.

file: indica el tipo de archivo

locate: me indica el path donde se encuentra un archivo

head: imprime las primeras 10 lineas de un archivo

write: envia un mensaje a otro usuario

ps: imprime los procesos ejecutandose en este momento -A

who: muestra información sobre los usuarios que están actualmente conectados.

chmod: cambia los permisos sobre un archivo, los permisos se declaran en formato octal o letra a letra.

chown: cambia el owner del archivo

cd: change directory

mkdir: make directory

rmdir: remove directory

du: disk usage

df: dis usage of a file

ln: crea un enlace a [objetivo] con el nombre [nombre-enlace]

ls: muestra los directorios que se encuentran dentro del path.

pwd: muestra el path actual

cp: copia archivos y directorios de [origen] a [destino]

mv: mueve y renombra archivos

> redireccion destructiva

>> redireccion no destructiva

kill: envia la signal SIGTERM a un determinado proceso

killall: envia la signal SIGTERM a todos los procesos

tar: empaqueta archivos -cvf (conjunto de archivos) -xvf (paquete.tar)

gzip: comprime archivos .tar // -d(descomprime tar.gz)

se pueden combinar usando tar -cvzf (conjunto de archivos) y -xvzf (archivo.tar.gz)

cat: imprime y concatena archivos.

cut: recorta lineas de un texto con modos indicados por parametros, como el -d: (separado por ':') -f1 (la primer columna)

wc -l: cuenta la cantidad de lineas que se leen desde la entrada estandar

grep: busca todos los archivos que contengan la cadena especificada dentro del directorio especificado (se usa para buscar nombres)

find: busca todos los archivos dentro del directorio especificado que sean lo especificado por parametros. (se usa para buscar por tipos de archivos y extensiones)

< hace que el archivo sea la entrada del comando (cuando el comando intente leer la entrada del teclado va a leer el contenido del archivo)

test: evalua si un archivo o directorio existe y otras cosas mas

touch: crea un archivo de texto en el path especificado