Laboratorio Linux Nº 1

Introducción

Linux es un sistema operativo gratuito y de libre distribución, cuya principal ventaja es su portabilidad. Existen varias versiones y distribuciones que se adaptan a casi todos los tipos de computadoras que hay en el mercado. Actualmente, Linux ofrece todas las características comunes en sistemas Unix modernos: multitarea real, multiusuario, consolas virtuales, compatibilidad con tipos de archivos estándar, capacidad de operar de red, soporte completo de hardware, entorno gráfico, librerías compartidas, seguridad, entre otras.
En cuanto a su arquitectura, se distingue por tener un **kernel** (núcleo) sobre él cual se encuentra una capa de Shell (interfaz de usuario). El kernel, que es la parte más cercana al hardware, se considera como el corazón del sistema. Dentro de él se encuentran los siguientes módulos:

- ✓ Gestión de procesos
- ✓ Gestión de archivos
- ✓ Gestión de memoria
- ✓ Gestión de entrada/salida
- ✓ Interfaz de llamadas al sistema

Existen dos versiones del kernel: la de producción, que es la estable y la de desarrollo.
Una **distribución** de Linux es un conjunto de herramientas, utilerías y programas que facilitan el trabajo con el sistema. Las distribuciones pueden basarse en diferentes versiones del kernel, incluir diferentes aplicaciones y ofrecer diversos procedimientos de instalación o actualización. Sin embargo, el único elemento común en todas ellas es el kernel.

Linux cuenta con varios programas encargados de interpretar los comandos ingresados por el usuario, ejecutando las acciones correspondientes. Estos programas se conocen como **shells**. El trabajo con Linux generalmente se realiza introduciendo comandos manualmente. El shell es capaz de interpretar una amplia gama de comandos y sentencias, además de permitir la creación de programas y scripts, llamados "shellscripts", para automatizar tareas. Existen varios tipos de shells, que se encuentran como archivos ejecutables en el directorio /bin. Los más utilizados son:

- ✓ ash: A Shell
- ✓ csh: C Shell (%)
- ✓ tscsh: extension del C shell
- ✓ ksh: Korn Shell ($)
- ✓ bsh: Bourne Shell ($)
- ✓ bash: Bourne Again Shell ($)

Los símbolos % y $ representan los prompts predeterminados de los distintos tipos de shells. El prompt es el indicador que aparece en la línea de comandos de una terminal, indica que está listo para recibir comandos.

Para conocer qué shells ya se encuentran instaladas en el sistema, ejecutar:
$cat /etc/shells

El Shell más utilizado es el bash, la cual suele venir preinstalada en la mayoría de las distribuciones. Combina características de csh y ksh, y ofrece las siguientes funciones:

- ✓ Completamiento automático: al escribir varios caracteres, se puede presionar la tecla TAB.
- ✓ Historial de comandos: permite moverse por los comandos previos con las teclas de cursor arriba y abajo
- ✓ Estructuras de control: incluye comandos como if, for, while, select y case

- ✓ Definición de funciones y alias: permite crear funciones para subrutinas y alias para asociar nombres a comandos con opciones y argumentos específicos.

Para cambiar de Shell, simplemente basta con escribir el nombre del Shell deseado.

Una **terminal** es un programa que recibe comandos del teclado y los envía al sistema operativo para su ejecución. En la mayoría de sistemas Linux, el programa encargado de esta función es bash, un intérprete de órdenes. Sin embargo, existen otros intérpretes disponibles (ya mencionados).

"En Linux hay GUIs (interfaz gráfica de usuario) donde se puede utilizar el mouse y conseguir hacer el trabajo sin previamente estudiar la documentación. El entorno tradicional de Unix es un CLI (interfaz de línea de comando), donde se escribe comandos para decirle al sistema lo que debe hacer. Este método es más rápido y más de poderoso, pero requiere descubrir cuáles son los comandos."
-- *man intro* (manual provisto por el intérprete)

Para ciertas tareas, especialmente la configuración del sistema, el uso de la terminal es más eficiente. Además, muchos foros y páginas de ayuda proporciona soluciones en forma de comandos de terminal.

Para abrir la terminal en entornos basados en Gnome (IGU), sigue los siguientes pasos:

Menú aplicaciones → Accesorios → Terminal

Los **comodines** son caracteres especiales que pueden reemplazar partes o la totalidad de nombres de archivos y directorios, facilitando la ejecución de comandos sobre múltiples elementos.

*: sustituye cualquier secuencia de caracteres.
Ejemplo: $ ls *.txt → muestra todos los archivos con extensión .txt

?: sustituye un único carácter.
Ejemplo: $ ls archivo**?**.txt → muestra archivo1.txt, archivo2.txt, etc.

Las **variables de entorno** son valores asignados por el sistema operativo al iniciar sesión y se pueden utilizar desde cualquier terminal.
Para ver las todas las variables de entorno definidas, se utiliza el comando: $ env
Algunas de las más utilizadas son:

- ✓ HOME: ruta del directorio personal del usuario.
- ✓ USER: nombre de usuario asignado.
- ✓ SHELL: ruta del intérprete de órdenes predeterminado.
- ✓ HOSTNAME: nombre asignado al equipo.
- ✓ PATH: rutas donde el sistema busca los comandos cuando no se especifica una ubicación.

Un **comando** es una instrucción que el usuario proporciona al sistema, ya sea desde la línea de comandos o mediante una llamada a programa. Generalmente, los comandos están contenidos en archivos ejecutables.
Los comandos pueden aceptar **parámetros de entrada**, que suelen indicarse con:
Un guion simple **-** → ejemplo: $ ls -l
Un guion doble **--** → ejemplo: $ grep --help

La **línea de comandos** o consola (shell) permite al usuario escribir instrucciones que, al presionar la tecla Enter, ejecutan procesos internos del sistema operativo o inician programas externos.
Sin embargo, las líneas de comandos no se ejecutan exactamente como se escriben, sino que son procesadas previamente por el intérprete de comandos, el cual realiza las siguientes funciones:

1. Muestra el prompt del sistema.
2. Expande los comodines.
3. Gestiona los redireccionamientos de entrada y salida.
4. Si el comando es interno, lo ejecuta directamente.
5. Si el comando es externo, lo busca en los directorios especificados en la variable PATH y solicita su ejecución al sistema operativo.
6. Si el comando no se encuentra, muestra un mensaje de error.

Los comandos del sistema se dividen en **internos** y **externos**.

- ✓ Comandos internos: Son instrucciones ejecutadas directamente por el intérprete de comandos, sin necesidad de buscar un archivo en el sistema.
  Ejemplo: $ cd (cambiar de directorio)

- ✓ Comandos externos: Son programas almacenados en el sistema de archivos que deben localizarse antes de ejecutarse. El intérprete de comandos busca estos archivos en los directorios específicos en PATH
  Ejemplo: $ ../ejecutame.exe (ejecuta un programa desde una ubicación específica)

Ejercitación

→ Abrir una terminal

1. Identificación de la distribución y arquitectura del sistema

Identificar el prompt de mi Shell. Además, qué otra información es visible?

PS1="[\u@\h \W]\$" -setea / podría elegir otro simbolito
\u = nombre de usuario
\h = nombre del sistema
\W = nombre de la carpeta actual
\$ = símbolo de usuarios normales

PS1="" -vaciar
source ~/.bashrc -vuelvo a la configuración por defecto

Escribe los siguientes comandos y anota la información que muestran:
arch → ¿Qué arquitectura tiene el procesador?
uname -a → ¿Qué información del sistema muestra?

Otras opciones: probar con los argumentos -o –all --version

Muestra la memoria RAM utilizada:
free -m

2. Uso de la Terminal y exploración del sistema
pwd → Muestra el directorio actual en el que se encuentra el usuario.
cd .. → Retrocede un nivel en la estructura de directorios.
mkdir prueba → Crea un directorio llamado "prueba".
rmdir prueba → Elimina un directorio vacío llamado "prueba".

clear → Limpia la terminal
whoami → Muestra el usuario actual
history → Muestra el historial de comandos usados
history -cw → Borra el historial de comandos usados
man comando → Muestra la documentación y opciones del comando.
uptime → Muestra el tiempo que lleva encendida la computadora
date → Muestra la fecha y horas actuales
cal 03 2024 → Muestra el calendario de marzo 2024
ps -u → Muestra los procesos en ejecución del usuario actual

3. Variables de entorno
env ó printenv ó printenv | less → Muestra todas las variables de entorno.
echo $HOME → Muestra el directorio personal del usuario.
echo $USER → Muestra el nombre del usuario actual
echo $PATH → Muestra las rutas que el SO utiliza para buscar archivos ejecutables
echo $SHELL → ¿Qué muestra?
echo $HOSTNAME → ¿Y este?
export MI_VARIABLE="Hola Linux" → Crea una variable de entorno temporal.
echo $MI_VARIABLE → Muestra el valor de la variable creada.

4. Comandos básicos para interactuar con archivos
touch archivo.txt → Crea un archivo vacío llamado "archivo.txt".
echo "Hola Mundo" > archivo.txt → Escribe "Hola Mundo" en el archivo.
cat archivo.txt → Muestra el contenido del archivo.
rm archivo.txt → Elimina el archivo.

ls → Lista los archivos y directorios del directorio actual.
ls -l → Lista los archivos y directorios con información detallada: permisos, propietario, tamaño, y fecha modificación.
ls -a → Lista los archivos, incluidos los ocultos (comienzan con .)
ls -F → Agrega un símbolo al final según el tipo (/directorios *ejecutables)

Crea un directorio llamado prueba_comandos y entra en él:
mkdir prueba_comandos && cd prueba_comandos
mkdir prueba_comandos; cd prueba_comandos

Crea varios archivos con nombres similares:
touch archivo1.txt archivo2.txt archivo3.txt notas1.log notas2.log

Lista solo los archivos que terminen en .txt:
ls *.txt

Lista solo los archivos que comiencen con "notas":
ls archi*

Redirige la lista de archivos a un nuevo archivo listado.txt:
ls > listado.txt
Muestra el contenido de listado.txt:
cat listado.txt
Agrega al final del archivo listado.txt la fecha actual:
date >> listado.txt
Muestra el contenido actualizado de listado.txt.