Cátedra de Sistemas Operativos

Tutorial de Introducción a Linux

Versión 1.3

Trabajo Práctico Nro. 1

1. Introducción

Este trabajo práctico pretende introducir al alumno al uso, configuración y breve administración del sistema operativo que será usado como plataforma para los trabajos prácticos posteriores. Se podrá realizar en el laboratorio o bien en una máquina virtual del alumno.

2. Objetivos

- Adquirir los conocimientos básicos necesarios para poder usar un sistema operativo moderno de tipo UNIX, como lo es Linux.
- Conocer y comprender las distintas herramientas de administración y configuración de dicho sistema operativo.
- Generar una base de información necesaria para la elaboración de los trabajos prácticos posteriores.

3. Desarrollo

Nota: tenga presente que se le solicita que informe que marca y versión utiliza de Linux, y que este Trabajo Práctico puede ser base para un trabajo practico posterior. Asimismo, dependiendo de la versión, algunos comandos pueden tener una sintaxis diferente. La entrega deberá ser solo en formato digital. En caso de ser recursante deberá utilizar una versión de Linux diferente a la presentada anteriormente (se conservan los históricos por 4 cuatrimestres).
IMPORTANTE: este trabajo práctico no evalúa la instalación de la maquina virtual, ya que no es condicionante para la realización del mismo.

3.1. Ayuda

3.1.1. man: es un programa que formatea y muestra las páginas del manual de referencia del sistema. El formato de uso básico es “man tema” donde tema es el nombre de la página del manual que se quiere ver.

3.1.1.1. ¿Qué tipo de información provee el man, como la organiza internamente y como busca dentro de la misma? Para saberlo, tipee “man man” (sin comillas), use los cursores UP y DOWN para recorrer la pagina.

3.1.1.2. Salga de la página anterior tipeando “q”.

3.1.1.3. Investigue que hace el comando ls tipeando “man ls”.

3.1.1.4. Liste de manera ordenada (por tamaño del archivo) mediante el comando “ls” la mayor cantidad de información posible sobre todos los archivos que se encuentren en su directorio home (ej: /home/guest) incluyendo aquellos que empiezan con un punto.

3.1.1.4.1. Tip: cuando se está viendo una página del manual, puede buscar cadenas tipeando “/cadenaABuscar” sin comillas.

3.1.1.4.2. Tip: para posicionarse dentro de un directorio debe hacer “cd directorio”. Ejemplo: cd /usr/local
3.1.1.5. Supongamos que se desea conocer el prototipo de la función de ANSI C printf(). Tipee “man printf” y vea que sucede. ¿Es la página que estábamos buscando?

3.1.2. whatis → Investigue que hace ese comando

3.1.2.1. Tipee whatis printf. Como podemos ver, existen resultados en más de una sección.
Punto 4.1.1.1 si es necesario y busque como indicar en qué sección buscar).

3.1.2.2. Al escribir por ejemplo “cd /bin” nos desplazamos hacia el directorio /bin. ¿Pero que hace exactamente “cd”? Tipee “whatis cd”.

3.1.3. whereis → Investigue que hace ese comando

3.1.3.1. Tipee “whereis ls”, “whereis socket” y “whereis printf” .

3.1.3.2. Del punto 3.1.1.4 seguimos sin tener una descripción formal de “cd”. Tipee “whereis cd” y vea que sucede.

3.1.4. help → Investigue que hace el comando help tipeando “help help”.

3.1.4.1. Tipee “help cd”.

3.1.4.2. ¡Finalmente logramos el objetivo! Dados los resultados del punto 3.1.2, 3.1.2, 3.1.3 y este punto, ¿Qué diferencia existe (no funcionalmente hablando) entre “cd” y por ejemplo “ls”? ¿Cuál de estos dos comandos es un “built-in command”?

3.1.5. apropos → Investigue que hace el comando apropos.

3.1.5.1. Supongamos que estamos buscando una función de C que se encarga de suspender la ejecución del proceso que la llama por un tiempo determinado. Tipee “apropos suspend” y vea si encuentra una función que cumpla con tales características.

3.1.6. info: es un programa para leer documentación, entre la cual se incluyen tutoriales para efectuar distintas tareas en Linux. Este se compone de una estructura del tipo árbol, dividido en nodos de información. Cada nodo describe un tópico específico con un determinado nivel de detalle, el mismo se encuentra señalado con un * (asterisco) y se puede acceder a él posicionando el cursor encima y teclando <enter>.

3.1.6.1. Investigue un poco más el comando info tipeando “man info”.

3.1.6.2. Ingrese al programa info tipeando “info”.

3.1.6.3. Para navegar entre los nodos de información, algunas opciones son:

3.1.6.3.1. u: desplaza al nodo superior.

3.1.6.3.2. n: desplaza al nodo siguiente.

3.1.6.3.3. p: desplaza al nodo previo.

3.2. Teclado / Terminales

3.2.1. ¿Qué sucede si tecleo cat /e <tab> p <tab>? (donde tab es la tecla tabulación). Presione <tab> nuevamente ¿Qué pasó ahora?

3.2.2. ¿Qué sucede si tecleo cat /e <tab> pas <tab>?

3.2.3. En este punto analizaremos las distintas terminales que hay en un sistema GNU/Linux. Ejecute los siguientes comandos e indique cuál fue el resultado:

3.2.3.1. who

3.2.3.2. Presione la tecla <alt>, y sin soltarla presione cualquiera de las teclas de función. En la pantalla debería aparecer el login del sistema, de lo contrario, ejecute el paso nuevamente presionando otra tecla de función. Si ya tiene el login del sistema vuelva a loguearse.

3.2.3.3. Ejecute nuevamente el comando who. ¿Qué diferencias encuentra con la primera vez que lo ejecutó?

3.2.3.4. Ejecute el comando whoami ¿qué muestra?, ¿Qué diferencias tiene con el comando ejecutado en el punto anterior?

3.2.3.5. Repita el paso 3.2.3.2 y el 3.2.3.3 hasta que no encuentre ninguna sesión para abrir.

3.2.3.6. Una vez terminado el punto anterior, Ud. se encontrará sesionado en el sistema como mínimo seis veces. Lo que acaba de hacer es abrir seis terminales virtuales (que podrían ser usadas por distintos usuarios, con diferentes perfiles), en la misma máquina. Así como existen terminales virtuales dentro del mismo equipo, si Ud. cuenta con una red, o con terminales tipo serie, podría abrir tantas sesiones de trabajo como Ud. quiera o necesite.

3.2.3.7. “Todo en Linux es un archivo”, y las terminales no son la excepción. Cada Terminal está representada por un archivo llamado ttyx donde x es un número de Terminal, y dichos archivos se encuentran dentro del directorio /dev. Investigue el comando “tty”.

3.2.3.8. Tipee “whatis echo” para saber rápidamente qué hace el comando echo. Luego asegúrese de estar logueado en la 1er y 2da Terminal, y desde la 1er terminal tipee “echo hola! > /dev/tty2”. ¿Qué pasó? (más adelante aprenderá en detalle el uso del ‘>’).

3.3. Sistema

3.3.1. Investigue los comandos:

3.3.1.1. poweroff

3.3.1.2. reboot

3.4. Usuarios

3.4.1. ¿Qué es la cuenta de superusuario (root) y para qué se utiliza? (probablemente tenga que buscarlo en internet).

3.4.2. Investigue que sucede con la cuenta de root en Ubuntu (el sistema operativo de uso en la cátedra). Investigue como realizar comandos a nombre del usuario root.

3.4.3. Investigue los comandos “su” y “login”

3.4.4. Loguearse como superusuario (root), y realizar los siguientes pasos (éste punto no puede ser realizado en el laboratorio):

3.4.5. adduser/addgroup

3.4.5.1. Investigue que hace el comando adduser/addgroup.

3.4.5.2. Cree un nuevo usuario, cree un nuevo grupo, y agregue el usuario a ese grupo.

3.4.6. deluser/delgroup

3.4.6.1. Investigue que hace el comando deluser/delgroup.

3.4.6.2. Borre el usuario creado anteriormente (incluyendo el borrado de su directorio en home y todos sus archivos).

3.4.7. Investigue cómo hacer para saber todos los grupos a los que pertenece un usuario.

3.5. Archivos

3.5.1. ¿Qué hacen los siguientes comandos?

3.5.1.1. cp

3.5.1.2. mv

3.5.1.3. rm

3.5.1.4. scp

3.5.1.5. telnet

3.5.1.6. ssh

3.5.1.7. touch

3.5.2. A la hora de referirse a archivos, se puede usar tanto su dirección relativa (al directorio en el que se encuentra situado) o absoluta. Sitúese como root dentro del directorio /root. Luego copie el archivo .bashrc a la ruta absoluta /var/.bashrc. Ahora, mueva ese archivo desde esa dirección hasta /home/.bashrc sin desplazarse del directorio inicial (/root).

3.6. Permisos

3.6.1. Cree un archivo tipeando “ls > archivo”.

3.6.2. Tipee ls –l en dicho directorio: los primeros 10 caracteres corresponden a los permisos. Investigue como se estructuran los permisos de un archivo (puede tipear info y luego ir a la sección de “permisos de archivo” o “file permissions”).

3.6.3. chmod/chown

3.6.3.1. Investigue que hacen esos comandos.

3.6.3.2. Haga que el archivo “archivo” creado anteriormente pueda ser modificado por cualquier usuario.

3.6.3.3. Compruebe que logró el punto anterior logueandose en otra Terminal con otro usuario y modificando dicho archivo (tipeando nuevamente “ls > archivo”).

3.6.3.4. Loguéese con el usuario original y quite los todos los permisos del archivo (lectura, escritura y ejecución) a todos los usuarios distintos del dueño y de los que pertenecen al mismo grupo. Luego, haga que el nuevo dueño del archivo sea el otro usuario.

3.6.3.5. ¿Cómo haría para volver a poseer dicho archivo sin loguearse con el nuevo dueño del archivo?

3.6.3.6. Investigue qué es el “SUID bit” (busque en “man chmod”).

3.6.3.7. Investigue como aplica la estructura de los permisos a los directorios.

3.6.3.8. Loguéese como root en otra Terminal y cree un directorio tipeando “mkdir /undir”.

3.6.3.9. Haga que cualquier usuario tenga todo tipo de permisos sobre ese directorio.

3.6.3.10. Deshaga lo que acaba de hacer, y cree el subdirectorio “subdir” dentro de “/undir”.

3.6.3.11. Investigue como cambiar los permisos de manera recursiva sobre /undir para que todos sus archivos, subdirectorios y archivos dentro de los subdirectorios se vean afectados.

3.7. Directorios

3.7.1. ¿Para qué se usa el comando cd? Ejecute las siguientes variantes de cd y observe cuál fue el resultado obtenido:

3.7.1.1. cd /

3.7.1.2. cd

3.7.1.3. cd /etc

3.7.1.4. cd .

3.7.1.5. cd ..

3.7.2. mkdir, rmdir, rm (nuevamente)

3.7.2.1. Investigue dichos comandos.

3.7.2.2. Borre un directorio que no se encuentra vacío.

3.7.2.3. Dentro de /home/<usuario> cree el directorio undir.

3.7.2.4. Ingrese a dicho directorio, y tipee lo siguiente para crear muchos archivos “while (true) do ps > $RANDOM.text; done;”. Tipee ctrl.+c luego de 5 seg para finalizar el comando. Luego tipee ls para corroborar la creación de los archivos.

3.7.2.5. Tipee “rm *” e investigue que pasó.

3.8. Filtros

3.8.1. ¿Cuál es la diferencia de los comandos more, less y cat? Cree un archivo de texto tipeando “ps –fea > texto” y visualícelo con los distintos comandos.

3.8.1.1. Investigue como buscar cadenas de texto cuando se visualiza un archivo con less. ¿Y cómo se hace para repetir la búsqueda? ¿Y para repetir la búsqueda hacia atrás? Esto le servirá cuando lea páginas del man! (ya que se leen mediante el less).

3.8.2. ¿Cuál es la diferencia entre tail y head?. ¿Qué hace la opción –f del comando tail?

3.8.2.1. Loguéese en una Terminal y tipee “echo > a.txt” para crear el archivo “a.txt”. Luego tipee “tail –f a.txt”.

3.8.2.2. Desde otra Terminal, tipee “while (true) do date >> a.txt; sleep 2; done;”.

3.8.2.3. Vuelva a la terminal anterior y vea lo que sucede.

3.8.2.4. No se olvide de finalizar el comando de la 2da Terminal! (con ctrl.+c)

3.8.3. ¿Qué es lo que realiza el comando sort?

3.8.4. ¿Qué es lo que realiza el comando uniq?

3.8.5. grep

3.8.5.1. ¿Para qué sirve?

3.8.5.2. Busque en el archivo “texto” todas las líneas que contengan la palabra “root”.

3.8.6 find

3.8.6.1. ¿Para qué sirve?

3.8.5.1. Busque un patrón determinado en un directorio utilizando find y grep.

3.9. Redirecionamiento de E/S

3.9.1. Antes de que un comando sea ejecutado, su entrada/salida estándar pueden ser redireccionados usando una notación especial del shell. Investíguelo tipeando “man bash” y llegando luego a la sección “REDIRECTIONS”.

3.9.1.1. Redireccionando la salida estándar.

3.9.1.1.1. Ejecute el comando “ps -fea” y redirija su salida a un archivo llamado “salida.txt”.
3.9.1.1.2. Ídem punto anterior, pero que se agregue la salida del comando al final del archivo.

3.9.1.2. Redireccionando la entrada estándar.

3.9.1.2.1. El comando “grep cadena archivo” imprime las líneas de archivo que contengan “cadena”. Investigue cómo hacer para lograr el mismo resultado sin especificarle a grep un archivo (investigue si es necesario qué hace grep cuando no se le especifica un archivo).

3.10. Pipelines (Tuberías)

3.10.1. El carácter | (pipe) se usa para conectar la salida estándar de un comando con la entrada estándar de otro. Investíguelo tipeando “man bash” y llegando luego a la sección “Pipelines” (o “tuberías” en castellano).

3.10.2. Haciendo uso de ps y grep, liste todos los procesos del usuario root.

3.10.3. Usando pgrep, liste todos los PIDs (Process Ids) de procesos que tengan “bash” en su comando de ejecución, redirija la salida a un archivo de texto, y repita esto último 2 veces más (agregando al final del archivo). Luego, liste el contenido del archivo de manera ordenada, eliminando las líneas repetidas y almacene dicho listado en un archivo (todo esto en un mismo comando!).

3.11. Vim

3.11.1. Vim es uno de los editores de texto que vienen por defecto instalados en todo sistema Linux.

3.11.2. Tipee “man vim” para investigar un poco sus características.

3.11.3. Para crear un archivo y editarlo con el vim, tipee “vim archivo.txt”.

3.11.4. Para comenzar a escribir debe ingresar al modo edición, presionando la tecla a. Escriba un poco de texto y luego salga del modo de edición presionando <ESC>.

3.11.5. Para grabar el archivo, presione: w (estando fuera del modo edición).

3.11.6. Para salir del editor, presione: q (estando fuera del modo edición).

3.11.7. Para profundizar sobre el vim (más adelante, cuando sea necesario) puede tipear “vim”. Así entrará a una pantalla desde la cual podrá tipear “:help” y ingresar a la ayuda. También puede tipear vimtutor, un programa que ofrece un tutorial completo del vim.

3.11.8. En caso de no sentirse cómodo con el vim, investigue el programa “nano”

4. Shell scripting y otras cuestiones

4.1.1. Variables de entorno

4.1.1.1. ¿Qué es una variable de entorno? ¿Qué significa exportar una variable?¿Cómo se la exporta?

4.1.1.2. Cree un script llamado environ.sh que exporte una variable y luego imprima su valor en pantalla.

4.1.1.3. Después ejecute dicho script en la forma “./environ.sh”. Al finalizar el script verifique si dicha variable se encuentra en el entorno del shell actual. Investigue qué ocurrió.

4.1.1.4. Ejecute el mismo script en la forma “source ./environ.sh”

4.1.1.5. ¿Qué ocurrió ahora? Investigue sobre el comando “source”

4.1.2. Procesos en primer y segundo plano

4.1.2.1. ¿Cómo se ejecutan comandos en segundo plano (background) en Linux

4.1.2.2. ¿Cómo hace para conocer qué procesos corren en segundo plano?

4.1.2.3. ¿Cómo trae un proceso a primer plano nuevamente?

4.1.3. /proc →. ¿Qué información contiene este directorio? Investigue exhaustivamente su contenido.

4.1.3.1. ¿Qué representan los directorios numéricos?

4.1.3.2. /proc/net: ¿Qué información contiene este directorio? A medida que envía datos a través de la red (por ejemplo al navegar con firefox), investigue qué archivos son modificados y qué representan esos valores cambiantes.

4.1.4. ¿Qué información contiene el archivo /etc/passwd?

4.1.5. ¿Para qué se utiliza el comando chsh?

4.1.6. SSH: Herramienta de logueo y ejecución de comandos remoto. Utiliza diversos algoritmos de encripción para proveer seguridad a la comunicación.

4.1.6.1. Investigue su uso y como automatizar el logueo de un usuario en una máquina remota (es decir, sin requerir el ingreso de password), leer man ssh-keygen y analizar el uso del archivo authorized_keys.

4.1.7. Procesamiento de textos (awk): awk es un lenguaje de programación destinado al procesamiento de textos.

4.1.8. ¿Qué realizan las siguientes operaciones?:

4.1.8.1. ls -l | awk '{print $6}'

4.1.8.2. cat /etc/passwd | awk 'BEGIN {FS=”:”} {print $1,$7}'

4.1.9. Algunas lecturas de man recomendadas:

4.1.9.1. bash, chsh, ssh, ssh-keygen, netstat, lsof, proc, stat, ps, grep, make, chmod, cut, awk, pidof, renice, nice, ifconfig, tar, gzip, kill, cat, echo, install, bc, tty, who, w, wc.

4.1.10. Algunos comandos built-in a investigar:

4.1.10.1. read, “.” (punto) o su alias source, fg, bg, jobs, export, unset, kill