

Laboratorio de Sistemas Operativos

Autor de la práctica: Dr. Juan Arturo Nolasco.

Revisores: M.C. Roberto Aceves, Ing. Jorge Villaseñor,
MC. Raul Fuentes, Dr. Jose I. Icaza



Práctica 1: UNIX como usuario. Comandos básicos

Objetivo:

Aprenderás el uso básico de la consola (shell) de LINUX y al final de esta práctica será capaz de:

- Utilizar los comandos básicos de Unix (ls,mkdir,cd, etc.)
- Consultar el **manual** en Línea de Unix (“man pages”) para descubrir cómo se utilizan algunos de los comandos
- Hacer uso del carácter de “pipe” (“ | ”) que se utiliza para ligar la salida de un comando con la entrada de otro.

¿Por qué un sistema basado en UNIX?

”UNIX es básicamente un sistema operativo simple, pero debes de ser un genio para entender la simplicidad” - Dennis Ritchie co-creador de UNIX y del lenguaje C (1941-2011)

En esta práctica nos enfocaremos en el manejo intermedio- avanzado de sistemas basados en UNIX y para ello utilizaremos la **interfaz de comando**, que es la forma más tradicional de manejar estos S.O., es decir, dejaremos a un lado la **interfaz gráfica de usuario (GUI)**.

¿Por qué el laboratorio está enfocado a Linux? Pues bien, la razón radica primero en la importancia del kernel UNIX/Linux como base de muchos otros sistemas operativos tales como Android, IOS, OS/X; como base de Apache, el software de servidor que se utiliza en la gran mayoría de los servidores de páginas web; y como base de muchos sistemas embebidos que se utilizan en todo tipo de artefactos. Y segundo porque se trata de **un proyecto de código abierto** y por lo mismo es completamente accesible para el alumno para generar ingeniería inversa en una o más de sus partes y realizar modificaciones si lo considera útil para su aprendizaje.

Información de Interés: [Richard M. Stallman: Importancia del uso del Software Libre en el ámbito educativo](#)

Comandos básicos: ls, cat, mkdir, rm...

Iniciaremos con lo más básico: moverse entre *directorios* (el equivalente a folders en otros sistemas operativos), desplegar el contenido de directorios, editar un archivo, etc; Para ello estaremos trabajando en

el directorio del usuario (/home/<user>) ya que sólo en ella tenemos total libertad. Teclea el comando `cd $home` cuando quieras dirigirte directamente a casa...

ls:	Lista archivos y directorios subordinados al directorio actual	
mkdir:	Crea un directorio.	
rmdir:	Borra un directorio.	Nota: el directorio debe de estar vacío.
cd:	Cambia el directorio actual a otro.	
less:	Permite visualizar un documento de texto.	
cat:	Lee, concatena y despliega archivos de texto en stdout*.	
head:	Despliega en stdout las primeras líneas de un archivo de texto.	
tail:	Despliega en stdout* las últimas líneas de un archivo de texto.	
rm:	Remueve (borra) archivos.	
pwd:	(Present Working Directory) Imprime la dirección del directorio actual	
tree:	Muestra la estructura de árbol de los directorios.	
touch:	Actualiza el "timestamp" de un archivo y si no existe el archivo lo crea.	

* **stdout**: representa la salida o resultado del comando, que por default es la terminal pero puede redirigirse hacia un archivo o bien hacia la entrada (stdin o standard input) de otro comando.

- Para redirigir stdout a otro archivo se utiliza "> archivo". Ejemplo: \$ ls > listado.
- Para agregar un archivo al final de otro, ">>": \$ cat unArchivo >> listado
- Para redirigir stdout al stdin de otro comando: "|": \$ cat unArchivo | head -10

Los comandos poseen opciones y/o parámetros que sirven para modificar su funcionamiento. Para una descripción breve del comando puede utilizarse el comando **whatis <comando>**, por ejemplo `whatis cd` ó `whatis whatis`. Para una mayor descripción del comando, consultar el **manual** ejecutando el comando **man <comando>**. Para salir del manual presiona la tecla "q". Para encontrar comandos que incluyan en su descripción una cierta palabra, teclea **man -k <palabra>**. "-k" (de keyword) es una de las opciones del comando man

== = Laboratorio == =

- Algunas de las actividades a realizar en esta práctica se encuentran descritas en este documento; sus respuestas deben registrarse en el siguiente formulario:
- - **Enlace al formulario:** [Laboratorio - UNIX como Usuario](#)
 - **Material de Apoyo:** [UNIX como usuario](#)

Actividad 1 - Manipulando el shell

1. El "shell" es el comando que interpreta y ejecuta todos los otros comandos. Cuando arrancas linux, el shell se comienza a ejecutar automáticamente y se queda esperando a que teclees otros comandos. Hay varios "shells" que se han desarrollado a lo largo de los años. **Verificar el shell**, asegurarse que se está utilizando el shell "**bash**", ejecutar :

```
[user@gecko ~]$ echo $BASH
/bin/bash
```

Si no aparece la segunda línea significa que nos hallamos en otro shell, para escoger BASH bastará con ejecutar el comando "bash" como se muestra en el siguiente ejemplo:

```
sh-4.2$ echo $BASH
/usr/bin/sh
sh-4.2$ bash
[user@gecko ~]$ echo $BASH
```

/bin/**bash**

2. Con el shell bash ya inicializado escriba “wh” y a continuación presione la tecla TAB en dos ocasiones. Notará que esto provoca que se desplieguen todos los comandos que inician con “wh” como se muestra a continuación:

```
[rfuentess@muuf ~]$ wh
whatis  whereis  which  while  whiptail  who  whoami
```

A excepción de **whiptail**, **while** y **whois**, la mayoría son comandos útiles para ver status de usuario, descripciones de programas y ubicación de los mismos.

En base a los comandos mostrados hasta este punto y la información que proporcionen sus respectivos MAN, conteste las siguientes preguntas:

1. ¿Cuáles comandos nos muestran el listado de usuarios activos en el sistema?
2. ¿Cuál sería el comando para desplegar la fecha del último “boot” (Reinicio) del sistema? Si el comando requiere determinadas opciones para desplegar esa fecha, inclúyelas en tu respuesta
3. Si un archivo tuviese exclusivamente 3 líneas de texto, ¿cuál sería la diferencia de utilizar los comandos head, tail, more y cat?
4. Si queremos leer el archivo **/etc/passwd** (el cual contiene el listado de usuarios del sistema) ¿Cuál sería el más apropiado entre los comandos head, tail, more y cat? ¿Por qué?
5. ¿Cuál es el comando que se recomienda utilizar en lugar de *more*?

Actividad 2 - ¿Dónde estamos?

Supongamos que nuestro usuario de nombre “fulano” tiene la estructura en su directorio HOME -obtenida mediante el comando *tree*- de la siguiente forma::

```
.
├── Desktop
├── Documents
│   └── Pruebas
├── Downloads
├── Music
├── Pictures
├── Public
├── Sesion1
│   ├── Archivo1
│   └── Archivo2.txt
├── Templates
└── Videos
```

Conteste las siguientes preguntas

1. ¿Que diferencia existe entre **Archivo1** y **Archivo2.txt**? (*pista*: En linux las “extensiones” como .txt no indican el tipo de archivo, sólo se utilizan como convenciones)

Si la línea en bash aparece como: fulano@host: /etc\$

1. ¿Cual es el comando para desplegar todo el contenido de Archivo2.txt utilizando direccionamiento

relativo al directorio en el que nos encontramos? Si el comando requiere determinadas opciones, inclúyelas

2. ¿Cual es el comando para desplegar el contenido del folder o directorio Sesión1, incluyendo los directorios lógicos (también llamados simbólicos) (*shortcuts* en otros s.o.) y en orden alfabético, utilizando direccionamiento absoluto (es decir, comenzando por la raíz del sistema de archivos, "/")?
3. ¿Cual es otro comando que produzca la misma información que el comando *tree*?
4. Valide su respuesta anterior con su propio directorio **HOME**, utilizando tanto *tree* como el comando sugerido por usted.

Actividad 3

Sumando todo lo visto en esta práctica, realice las siguientes actividades.

Ejecute los siguientes comandos

```
mkdir $HOME/Operativos
touch $HOME/Operativos/Arch1
touch $HOME/Operativos/Arch2
touch $HOME/Operativos/Arch3
```

Conteste las siguientes preguntas:

1. Comando para copiar el contenido del archivo */etc/passwd* a Arch1
2. Comandos para guardar en otro archivo Arch2, las descripciones de *whatis* de todos los comandos *wh* (Primera actividad)
3. Guarde la ubicación del comando *ping* en **Arch3**
4. Anexe el contenido de **Arch1** y **Arch2** en **Arch3 (Sobre-escribalo)**
5. Desde *\$HOME/Operativos* ejecute el comando *mkdir ./Acto1*
6. Mueva el archivo **Arch3** al directorio creado en el paso anterior.
7. Despliegue la primera línea de **Arch3** con direccionamiento absoluto
8. Estando en su carpeta Home y utilizando direccionamiento relativo, genere un archivo llamado **Arch4** en el directorio **Acto1** que contenga las últimas 5 líneas de **Arch2**.
9. Utilizando solamente los comandos **tail** y **head**, despliegue la segunda línea de **Arch4**.
10. Utilice solamente un único comando para borrar todo el contenido del directorio **Operativos**

NOTA: Para crear y editar archivos utilice el editor que le parezca más conveniente. Si no conoce algún editor se recomienda utilizar *gedit* en modo gráfico o *nano* en consola.

Información Adicional (Tutoriales)

- [Basic Linux Commands](#)
- [UNIX Tutorial for Beginners \(Tutorial One, Two & Three\):](#)
- [Listado de todos los comandos](#)

[1] Path Definition. The Linux Information Project. 2007. <http://www.bellevuelinux.org/path.html>

References

Como encontrar documentación dentro de mi propio sistema GNU/Linux - Desde Linux. (n.d.). Retrieved May 19, 2014, from

<http://blog.desdelinux.net/como-encontrar-documentacion-dentro-de-mi-propio-sistema-gnulinux/>