

Sistemas Operativos

Practica de laboratorio #1

Sistemas Operativos
Daniel Ochoa, Antonny García,
{dochoa, agarcia}@fiec.espol.edu.ec

Objetivos:

- Demostrar las funciones del Sistema Operativo que fueron aprendidas en clase.
- Identificar la estructura de los directorios de un sistema Operativo.
- Identificar la estructura de un Sistema Operativo basado en UNIX.
- Realizar tareas de monitoreo de recursos empleando un shell.
- Comunicar programas usando un shell.
- Encontrar APIs y sus funciones.
- Identificar los modos de operación de un sistema.

Requerimientos:

- Un computador con GNU/Linux.
- Versión de Kernel: al menos 2.6.24

Actividad No 1: Sistema de archivos.

El sistema de archivos proporciona una forma de separar y buscar datos en un disco duro a través de una organización jerárquica basado en directorios. El directorio / se denomina directorio raíz. A partir de este se ubican los demás directorios, entre los cuales se encuentran:

/home	Datos de usuarios
/dev	Dispositivos del sistema
/var/log	Mensajes generados por procesos en background
/usr/src/linux....kernel	Código fuente kernel

Para acceder a cada directorio, se usa el comando **cd** <nombre directorio>. Con **cd ..** Se regresa al directorio superior. Acceda a al directorio **/bin** y observe su contenido usando el comando **ls**. Reconoce a que corresponden los archivos en ese directorio?

A

Actividad No 2: Excepciones.

Las excepciones son interrupciones activadas por hardware cuando el CPU recibe una señal eléctrica desde el dispositivo. El comando **lsusb** lista dispositivos con interfaz usb. Observe la información que brinda el comando. Ahora conecte un dispositivo usb, vuelva a escribir el comando. ¿Pudo observar algún cambio?

B

Ahora ejecute **lspci** que tipo de información que proporciona la salida de este comando.

C

Sugerencia: Para saber opciones de un comando escriba **man** <nombre del comando>

Actividad No 3: Uso de recursos (procesos).

Se pueden supervisar los procesos y sus recursos desde el shell. Un comando útil es **top**. Liste 2 ejemplos de tipos de recursos se pueden ver en uso? Para terminar la ejecución de **top**, presione las teclas **ctrl + C**, ó **ctrl + Z** .

D

- 1.
- 2.

Top también provee de un resumen del estado de los procesos. Cuántos procesos hay y en que estado se encuentran en el sistema?

E

También puedes ver la lista de procesos en ejecución usando el comando **ps** y sus opciones, prueba las combinaciones descritas abajo, que diferencias hay entre ellos?

F

- | | |
|---------------|--|
| 1. ps -ax | |
| 2. ps -axjf | |
| 3. ps -u root | |

Usando el resultado **F** es posible deducir que shell se está usando. Anótelo a en la casilla de abajo.

G

Actividad No 4: Uso de recursos (sistema).

Además de los recursos por proceso se pueden ver el uso total de los recursos del sistema. Algunos ejemplos están abajo, explica que se obtiene al ejecutarlos:

H

- | | |
|---------|--|
| 1. df | |
| 2. free | |

Como en UNIX todo es un archivo es posible leer archivos de dispositivo para ver el estado del mismo. El CPU se representa con el archivo **/proc/cpuinfo**. Use el comando **more** para ver cuantos cores hay y que velocidad tienen.

I

Actividad No 5: Comunicación entre procesos.

Los procesos que se cargan desde un shell se pueden comunicar entre si usando caracteres especiales. Describa que hacen **|** y **>** en los siguientes comandos:

J

top -n 1 grep init	
----------------------	--

K

ps ax > archivo.txt more archivo.txt	
---	--

Procesos que usan comunicación remota (por ejemplo en internet) usan la el API de sockets para enviar datos formateados en el protocolo adecuado a la tarjeta de red. Ejecute **tcpdump** en el shell y describa su salida.

L

Los procesos pueden correr en background, es decir sin estar asociado a un shell. Los mensajes se pueden grabar en archivos log. Ejecute el comando **tail -f /var/log/messages** en una nueva consola para er los mensajes del sistema. En la consola original escriba el comando **su**, ingrese cualquier palabra, pulse la tecla enter, que ocurrió?

M

Actividad No 6: Programas del S.O.

Es posible conocer que procesos que se ejecutarán cuando arranca el S.O (servicio). En el shell acceda al directorio **/etc/init.d** con el comando **cd**. El comando **./<nombre servicio> start|stop|restart|status** sirve para subir|bajar|reiniciar|monitorear un servicio. Qué pasa si se detiene el servicio network?.

N

Actividad No 7: APIs del sistema

Con el comando **ldd** se puede conocer de que librerías depende un programa. La forma de usarlo es la siguiente: **ldd <directorio/nombre programa>**. Escriba las dependencias del programa **/bin/cat**.

O

Para ver las funciones que provee un API se usa el comando **nm -D <directorio_librería/ nombre_librería>**. Busque en que librería utilizada por el programa **cat** esta implementada la funcion **printf**.

P

Sugerencia: después del comando nm usa "| grep printf"

Actividad No 8: Configuración del sistema (opcional).

Una vez que el kernel se encuentra en memoria, este inicia el proceso init. El archivo **/etc/rc-default** define que programas se van a ejecutar. Esto cambia según el runlevel (nivel de ejecución) del sistema. Existen al menos 7 niveles numerados. Describa que configuración tiene el sistema para los runlevels del 1 al 5

Q

0. Runlevel 0: Detener sistema
1. Runlevel 1:
2. Runlevel 2:
3. Runlevel 3:
4. Runlevel 4:
5. Runlevel 5:

Sugerencia: El escritorio gnome tiene con un editor de runlevels