Trabajo Práctico Nº 3

Parte 1 - Kernel y Registry de Windows

1.- Describa la arquitectura básica de Windows. Para los principales componentes: ¿Qué archivos del FileSystem contienen su implementación?

2.- Respecto a la arquitectura de Windows, responda sintéticamente los siguientes puntos:

a. ¿Qué funciones se realizan en el componente "Executive" del kernel?
b. ¿Qué funciones se realizan en el componente "Kernel" del kernel?
c. ¿El sistema de ventanas (Windowing) se ejecuta en Kernel o User Mode?
d. ¿Qué es la HAL?

3.- ¿Qué es un microkernel? ¿Se considera que Windows es Microkernel? ¿Por qué?

4.- ¿Qué es la Registry? ¿A que hace referencia su contenido?

5.- La estructura de la Registry está dividida en cinco (5) secciones raíces:

a. ¿Cuáles son estas secciones?
b. ¿A que hace referencia la información incluida en cada una? Indique las principales secciones de cada una.
c. ¿Cuáles son links simbólicos? Indicar para cada una a que sección de la registry referencian.

6.- ¿Qué tipos de elementos puede contener la Registry?

7.- ¿Qué característica tiene la clave HKEY_LOCAL_MACHINE\Hardware?

8.- Suponga que usted desarrolla una aplicación la cual genera archivos con extensión "XXI". ¿Qué sección sería necesaria modificar para que, al hacer clic sobre el archivo de extensión XXI, se abra con nuestra aplicación?

9.- ¿Qué modificación debería realizar para que una aplicación, por ejemplo Acrobat Reader, se ejecute cuando su usuario inicia sesión?

10.- ¿Qué modificación debería realizar ahora para que la aplicación que indico en el punto anterior se ejecute para todos los futuros usuarios que se creen en el sistema?

11.- Analice la Registry e indique la ruta completa en donde se encuentra valor que indica cual es el wallpaper del usuario que esta logueado.

12.- Cada sección de la Registry se encuentra almacenada en diferentes archivos (llamados "hives") en el FileSystem.

a. ¿Cuál es la ruta completa en el FS que contiene ésta información?
b. Ingrese a su Registry e indique cuáles son estos archivos y donde se encuentran en el FileSystem.

c. Windows realiza backups de estos archivos. Indique en que directorio del File System son almacenados dichas copias.

13.- En el sistema operativo Linux, si necesitamos que un servicio se inicie al arrancar la computadora, el script que realiza esta tarea es colocado en la carpeta que corresponde al Run Level deseado (/etc. /rcX.d para el Run Level "X").

α. ¿Cuál es la ruta completa dentro de la Registry donde se indican los servicios que debe arrancar Windows cuando es iniciado?

β. ¿Es posible en Windows determinar el orden en que los servicios son ejecutados?

En caso afirmativo indique en que parte de la Registry se almacena esta información.

Ejercicios Guiados – Registry Tweak's

Aclaración: Los siguientes ejemplos fueron probados sobre Windows 7. Es posible que para versiones diferentes des SO alguna no funcione, o varíe.

14.- Supongamos que queremos hacer que aparezca un mensaje al usuario antes de que este inicie sesión. Esta situación puede ser útil en una empresa, por ejemplo, si queremos informar algo a todos los usuarios.

Para ello vamos a agregar los siguientes valores:

Sección:	HKEY_LOCAL_MACHINE\SOFTWARE
Clave:	\Microsoft\Windows NT\Current Version\Winlogon
Nombre:	LegalNoticeCaption
Tipo:	REG_SZ
Valor:	El título que desee para el cuadro del mensaje

Sección:	HKEY_LOCAL_MACHINE\SOFTWARE
Clave:	\Microsoft\Windows NT\Current Version\Winlogon
Nombre:	LegalNoticeText
Tipo:	REG_SZ
Valor:	El texto que desee para el cuadro del mensaje

Luego de realizar estas modificaciones, reinicie la computadora y corrobore que los cambios tuvieron efecto.

15.- Al insertar un CD-ROM Windows nos muestra una ventana en la que se puede elegir qué hacer con el tipo de cd insertado. Es posible deshabilitar esta opción para todos los cd´s. Para ello modificamos la registry de la siguiente manera:

Sección:	HKEY_LOCAL_MACHINE\SYSTEM
Clave:	\CurrentControlSet\Services\Cdrom
Nombre:	AutoRun
Tipo:	REG_DWORD
Valor:	0 deshabilitar el autorun y 1 para habilitarlo

Luego de realizar estas modificaciones, reinicie la computadora y corrobore que los cambios tuvieron efecto.

16.- Cambiando la información acerca del Procesador que se muestra en la solapa de "Propiedades del Sistema". Esta opción puede ser útil para hacerle creer a alguien que posee una CPU diferente a la real

Sección:	HKEY_LOCAL_MACHINE\HARDWARE
Clave:	Description\System\CentralProcessor\0
Nombre:	ProcessorNameString
Tipo:	REG_SZ
Valor:	Super CPU

17.- Utilice la herramienta **Process Monitor** para analizar la utilización de la Registry y responda a las siguientes preguntas:

- Cambie el wallpaper que posee el usuario ¿En que clave de la Registry se almacena dicha información?
- Intente abrir un archivo con extensión .txt con el programa notepad.exe, definiendo al mismo como programa por defecto para abrir este tipo de extensión de archivos ¿En qué clave se almacena dicha asociación?

Referencias:

- ✓ Registry de Windows: (2013) http://msdn.microsoft.com/en-us/library/ms724871(v=vs.85).aspx

- ✓ Ejemplos de registry tweaks : (2009)
  http://www.theeldergeek.com/registry_edits.htm
  http://www.elhacker.net/trucosxp.htm

- ✓ Utilizar la API de la Registry para guardar y recuperar información: (2009)
  http://support.microsoft.com/kb/q145679/

- ✓ Información del Registro de Windows para usuarios avanzados: (2009)
  http://support.microsoft.com/kb/256986/es

- ✓ Entendiendo la registry de Windows: (2009)
  http://www.fermu.com/content/view/12/26/lang,es/

- ✓ **Process Monitor**: Herramienta para monitorizar todo el movimiento de la registry en tiempo real. Permite ver que claves son accedidas, modificadas, y el resultado que obtiene la aplicación durante el acceso (por ejemplo, error). (2013)
  http://technet.microsoft.com/en-us/sysinternals/bb896645

Parte 2 - Sistema de archivos /proc

1. ¿Qué es el sistema de archivos /proc? ¿Qué tipo de información se almacena en él? ¿La información que contiene puede ser modificada?
2. ¿Cómo y cuando se crea? ¿Quién lo crea?
3. ¿De qué manera está organizada la información que en él se encuentra?
4. Indicar a qué archivo del /proc debe acceder para consultar información la información del equipo local acerca de:
   - **4.1.** Características de procesador
   - **4.2.** Filesystems configurados en el kernel
   - **4.3.** Dispositivos configurados en el kernel
   - **4.4.** Memoria del sistema
   - **4.5.** Carga promedio del sistema
   - **4.6.** Utilización de memoria
   - **4.7.** Módulos del kernel cargados (verifique si el módulo creado en la práctica 3 se encuentra cargado)
   - **4.8.** Tiempo desde que el sistema se encuentra funcionando (desde el último reinicio)
   - **4.9.** Versión del Kernel
5. Haga un ls –l del /proc. ¿Qué característica en particular nota acerca del tamaño de cada uno de los archivos allí almacenados? ¿Cuál es la razón?
6. Nombre al menos 2 utilitarios (comandos del bash) que hagan uso de la información que se encuentra almacenada en el /proc. Indique a qué sección del /proc acceden.
7. ¿Qué tipo de información refleja un directorio cuyo nombre es un número? ¿A qué hace referencia este número?
8. ¿Cuáles son los archivos mas relevantes que puede encontrar dentro de un directorio del /proc cuyo nombre es un número? Indique que información se encuentra en cada uno de los archivos mencionados.
9. ¿Qué información contiene el directorio /proc/sys? ¿A qué hace referencia cada directorio dentro de /proc/sys? ¿La información que contiene puede ser modificada? En caso afirmativo, indique cómo.
10. ¿Cual es la semántica del archivo /proc/sys/kernel/ctrl-alt-del? ¿Que provoca el siguiente comando?: `echo "1" > /proc/sys/kernel/ctrl-alt-del`
11. ¿Es persistente luego de un reinicio del sistema la información modificada sobre el /proc? Mencione dos maneras de hacer que el cambio sea persistente

12. ¿Qué es y para qué sirve **sysctl**? ¿Cuál es la función del comando sysctl? ¿Qué devuelve la ejecución del comando sysctl –a?
13. ¿A qué hace referencia el enlace simbólico "self" bajo el directorio /proc?
14. Accediendo al /proc, identifique la siguiente información para los procesos ssh, syslogd, cron, bash:
    - **14.1.** Argumentos pasados al proceso al momento de ejecutarlo
    - **14.2.** Utilización de la CPU por parte del proceso
    - **14.3.** Lista de las variables de entorno utilizadas por el proceso
15. ¿Qué tipo de información se puede encontrar en el archivo status del área de un proceso determinado en el /proc? Identifique la información del proceso bash.
16. El comando **ps** muestra información acerca de los procesos actuales en el sistema. Para hacer esto, obtiene información del directorio /proc. ¿De que archivos lee información para cumplir su función?
17. Cree un script que reciba como parámetro el nombre de un proceso y que a partir de éste informe el comando mediante el cual fue invocado, calculado a partir de archivos del directorio /proc.
18. Cree un script que reciba un PID como parámetro e informe si el proceso se encuentra corriendo en el sistema actualmente. En caso afirmativo, debe informar el nombre del proceso, así como también su estado.

Referencias:

- ✓ Páginas del man del /proc (man proc)
- ✓ http://www.linuxjournal.com/article/8381
- ✓ http://www.faqs.org/docs/kernel/x716.html
- ✓ http://www.linuxinsight.com/proc_filesystem.html
- ✓ http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/ch-proc.html
- ✓ http://www.tldp.org/LDP/sag/html/proc-fs.html

Parte 3 – *kobject* y sistema de archivos *sysfs*

LEA las referencias de esta parte de la práctica, y **LUEGO** resuelva:

kobject

1. ¿Qué es un *kobject* y cuál es su función?
2. ¿Qué es un *ktype*?
3. ¿Qué es un *kset*?
4. Cuando se crea un *kobject*, ¿qué dos funciones se deben llamar para que el mismo sea incluido en el *sysfs*? ¿Existe alguna alternativa más simple?
5. En una máquina con Linux y su código fuente (puede ser la utilizada en el trabajo práctico anterior, aunque no es un requerimiento), descomprima el archivo `unlp_kobject.tar.gz` que acompaña este trabajo práctico.
   a) Analice el archivo `unlp_kobject.c` y responda:
      I. ¿Qué hacen las funciones `unlp_show` y `unlp_store`?
      II. ¿Dónde está definida la macro `__ATTR`? ¿Qué hace?
   b) Modifique la variable `KDIR` del archivo `Makefile` para que apunte al directorio que contiene el código fuente del kernel en ejecución.
   Nota: En caso de estar utilizando Debian o alguna *distro* derivada, se recomienda dejar la variable como está e instalar los *headers* del kernel. Para instalar los headers del kernel en ejecución se debe correr como usuario `root` el siguiente comando:
   ```
   apt-get install linux-headers-$(uname -r)
   ```
   c) Dentro del directorio que contiene el código del modulo ejecute el siguiente comando:
   ```
   make
   ```
   d) Inserte el módulo recién compilado utilizando el comando correspondiente que aprendió en el trabajo práctico anterior.
   e) Deduzca en qué parte del sistema de archivos está el *kobject* que acaba de crear. Cuando encuentre el archivo, aplíquele varias veces el comando `cat`. ¿Qué ocurre?

Sysfs

1. Implemente un shellscript que detecte las unidades de disco removibles del sistema e imprima en pantalla sus nombres.
2. Vea el contenido del archivo /sys/power/state (de ser posible no use una máquina virtual).
   a) ¿Qué strings contiene?
   b) Si tiene archivos abiertos guárdelos, seleccione uno de esos strings y pruebe el comando:

1. `echo stringSeleccionado > /sys/power/state`

c) ¿Qué hace el comando anterior para cada uno de los strings? (la respuesta "el comando apaga la máquina" es incorrecta).

3. Implemente su propia versión del comando "lsmod" que muestra los módulos cargados, usando el subsistema module.

Muestre solamente los módulos cuyos kobjects tengan el atributo refcnt (es decir los módulos que pueden ser descargados con rmmod. Vea las referencias.).

4. Implemente "lsmod" usando /proc/modules.
5. Implemente lo que se conoce como "modo avión" en shellscript usando la clase rfkill. El "modo avión" consiste en apagar todos los dispositivos Wifi y Bluetooth.
6. Vamos a monitorear eventos relacionados con kobjects:

a) Copie el siguiente código en /usr/local/bin/mihotplug.sh:

```
#!/bin/sh

LOG=/tmp/mihotplug-${SEQNUM}.log

echo "***************** Evento $(date)
********************" >> $LOG

echo ------------------------- Argumentos ----------------
------- >> $LOG

echo "$@" >> $LOG

echo ------------------------- Entorno ------------------
-------->> $LOG

env >> $LOG
```

b) Ejecute lo siguiente para registrar el script como receptor de los eventos del kernel:

```
echo /usr/local/bin/mihotplug.sh > /sys/kernel/uevent_helper
```

c) Vea los archivos que se van generando en /tmp/ y analice que sucede cuando conecta y desconecta dispositivos USB, y cuando saca un CD de la lectora y pone otro (esta última prueba se puede hacer fácilmente en una máquina virtual).

d) Considere solamente los archivos ".log" que devuelvan entradas con DEVTYPE=usb_interface

¿El script mihotplug.sh recibe suficiente información del kernel como para insertar los módulos necesarios para usar los dispositivos que se conectan? Si sospecha que no ¿de donde sacaría la información faltante?.

Este mecanismo era el usado por hotplug para configurar el hardware antes de la existencia de udev. Udev utiliza un mecanismo similar pero no se

registra en /sys/kernel/uevent_helper sino que escucha los eventos del kernel con un socket.

Ayuda: Es suficiente con obtener el modalias de alguna forma.

Referencias

- ✓ <kernel_code>/Documentation/kobject.txt
- ✓ <kernel_code>/Documentation/sysfs-rules.txt
- ✓ kobjects and sysfs
- ✓ <kernel_code>/Documentation/ABI/stable/sysfs-module
- ✓ <kernel_code>/Documentation/ABI/stable/sysfs-class-rfkill
- ✓ <kernel_code>/Documentation/ABI/testing/sysfs-power
- ✓ http://www.markus-gattol.name/ws/udev.html