ARQUITECTURA Y SISTEMAS OPERATIVOS

Trabajo Práctico: Gestión de Servicios en los Sistemas Operativos

Objetivo y Escenario para la realización del trabajo práctico

Este trabajo práctico busca comparar y analizar los servicios activos por defecto en Windows y los daemons en Linux, así como su influencia en los logs de eventos de cada sistema operativo. Se utilizarán comandos nativos de cada sistema, PowerShell en Windows y asistentes de línea de comandos en Linux para realizar esta exploración.

Consigna:

1. Preparativos:

Entorno de trabajo:

- Una máquina virtual o física con Windows 10/11.
- Una máquina virtual o física con una distribución Linux (por ejemplo, Ubuntu o CentOS).

Herramientas necesarias:

- En Windows: PowerShell (preinstalado) y el Command Prompt.
- En Linux: Acceso a terminal con permisos de superusuario.

Configuración:

- Verifica que las máquinas tengan configurados los servicios básicos por defecto.
- Asegúrate de que ambos sistemas operativos tengan habilitados los logs de eventos.

2. Tareas:

Exploración en Windows

- Listar servicios activos por defecto:

Usar Get-Service en PowerShell

Get-Service | Where-Object {$_.Status -eq "Running"}

Usar el comando en Command Prompt:

sc query | findstr "RUNNING"

- Examinar parámetros de los servicios:

Obtener detalles de un servicio específico con PowerShell:

Get-Service -Name <NombreServicio> | Format-List *

Inspeccionar configuraciones con PowerShell:

Get-WmiObject Win32_Service | Select-Object Name, StartMode, State

Analizar logs de eventos relacionados:

Usar Get-EventLog en PowerShell:

Get-EventLog -LogName System -EntryType Information, Warning, Error

Exploración en Linux

Listar servicios activos por defecto:

Usar `systemctl` para listar servicios activos:

systemctl list-units --type=service --state=running

Filtrar servicios habilitados al inicio:

systemctl list-unit-files --type=service | grep enabled

Examinar parámetros de los *daemons*:

Inspeccionar un servicio específico:

systemctl show <nombre_servicio>

Ver detalles de configuración:

cat /etc/systemd/system/<nombre_servicio>.service

Analizar logs de eventos relacionados:

Usar `journalctl` para ver los logs:

journalctl -u <nombre_servicio>

3. Resultados esperados:

Listado completo y comparativo de servicios activos por defecto en Windows y daemons en Linux.

Descripción de los parámetros más relevantes en ambos sistemas (estado, inicio automático, usuario ejecutor, etc.).

Registro de eventos relacionados en los logs de sistema, identificando diferencias en la forma en que se registran los eventos en ambos sistemas operativos.

4. Preguntas de análisis:

A. **¿Qué similitudes y diferencias existen entre los servicios de Windows y los daemons de Linux en cuanto a funcionamiento y parámetros?**
B. **¿Cómo afecta la configuración de un servicio o daemon en los logs de eventos de cada sistema operativo?**
C. **¿Qué tipos de eventos generan los servicios en Windows frente a los daemons en Linux?**
D. **¿Cómo influyen los parámetros de inicio automático en el rendimiento general del sistema en ambos casos?**
E. **¿Qué desafíos surgen al administrar servicios en cada sistema operativo?**

5. Extensión opcional:

Automatización del análisis: Crea un script en PowerShell y otro en Bash que automatice las tareas de listado, inspección y extracción de logs.

Visualización de datos: Usa herramientas de visualización como Power BI (Windows) o Grafana (Linux) para representar gráficamente el impacto de los servicios en el sistema.

Integración con otros sistemas: Explora cómo los servicios o daemons pueden interactuar con sistemas de monitoreo externo como Nagios o Zabbix.

Resolución de la Práctica

A continuación se muestran ejemplos simulados de salidas de pantalla obtenidas al ejecutar los comandos indicados en el trabajo práctico. Estos ejemplos representan lo que se podría observar en un entorno real; las salidas pueden variar según la configuración y el estado del sistema.

---

1. Exploración en Windows

a) Listar servicios activos por defecto

Usando PowerShell:
Comando:

```
Get-Service | Where-Object {$_.Status -eq "Running"}
```

Salida simulada:

```
Status   Name               DisplayName
------   ----               -----------
Running  Audiosrv           Windows Audio
Running  BITS               Background Intelligent Transfer Service
Running  CryptSvc           Cryptographic Services
Running  Dhcp               DHCP Client
Running  EventLog           Windows Event Log
Running  LanmanServer       Server
Running  wuauserv           Windows Update
...
```

Usando Command Prompt:
Comando:

```
sc query | findstr "RUNNING"
```

Salida simulada:

```
SERVICE_NAME: Audiosrv
        STATE              : 4  RUNNING
SERVICE_NAME: BITS
        STATE              : 4  RUNNING
SERVICE_NAME: CryptSvc
        STATE              : 4  RUNNING
SERVICE_NAME: Dhcp
        STATE              : 4  RUNNING
SERVICE_NAME: EventLog
        STATE              : 4  RUNNING
SERVICE_NAME: LanmanServer
```

```
        STATE              : 4  RUNNING
SERVICE_NAME: wuauserv
        STATE              : 4  RUNNING
...
```

---

b) Examinar parámetros de un servicio

Con PowerShell (por ejemplo, para el servicio Windows Update "wuauserv"):
Comando:

```
Get-Service -Name wuauserv | Format-List *
```

Salida simulada:

```
Name             : wuauserv
DisplayName      : Windows Update
Status           : Running
DependentServices: {wuauserv_dependency1, wuauserv_dependency2}
ServiceType      : Win32OwnProcess, InteractiveProcess
StartType        : Automatic
ServiceName      : wuauserv
...
```

Inspección con WMI:
Comando:

```
Get-WmiObject Win32_Service | Select-Object Name, StartMode, State
```

Salida simulada:

```
Name                StartMode     State
----                ---------     -----
Audiosrv            Auto          Running
BITS                Auto          Running
CryptSvc            Auto          Running
Dhcp                Auto          Running
EventLog            Auto          Running
LanmanServer        Auto          Running
wuauserv            Auto          Running
...
```

---

c) Analizar logs de eventos

Con PowerShell:
Comando:

```
Get-EventLog -LogName System -EntryType Information, Warning, Error -
Newest 10
```

Salida simulada:

```
Index Time         EntryType   Source         EventID Message
```

```
----- ----          ---------   ------          ------- -------
10234 Feb 01 14:23 Information Service Control Manager 7036  El
servicio 'Windows Update' cambió de estado a 'Running'.
10233 Feb 01 14:20 Warning     Kernel-Power   41     El sistema se
reinició sin apagarse correctamente.
10232 Feb 01 14:18 Error       Disk           7      Error en la
detección de un dispositivo de almacenamiento.
...
```

---

2. Exploración en Linux

a) Listar servicios activos por defecto

Usando systemctl:
Comando:

```
systemctl list-units --type=service --state=running
```

Salida simulada:

```
  UNIT                            LOAD   ACTIVE SUB     DESCRIPTION
  accounts-daemon.service         loaded active running Accounts
Service
  apache2.service                 loaded active running The Apache
HTTP Server
  cron.service                    loaded active running Regular
background program processing daemon
  dbus.service                    loaded active running D-Bus System
Message Bus
  ssh.service                     loaded active running OpenBSD Secure
Shell server
  systemd-journald.service        loaded active running Journal
Service
  systemd-logind.service          loaded active running Login Service
  udev.service                    loaded active running udev Kernel
Device Manager
  ...
```

Filtrar servicios habilitados al inicio:
Comando:

```
systemctl list-unit-files --type=service | grep enabled
```

Salida simulada:

```
accounts-daemon.service        enabled
apache2.service                enabled
cron.service                   enabled
dbus.service                   enabled
ssh.service                    enabled
systemd-journald.service       enabled
systemd-logind.service         enabled
udev.service                   enabled
...
```

b) Examinar parámetros de un daemon

Inspeccionar un servicio específico (por ejemplo, ssh):
Comando:

```
systemctl show ssh.service
```

Salida simulada:

```
Id=ssh.service
Names=ssh.service
Description=OpenBSD Secure Shell server
LoadState=loaded
ActiveState=active
SubState=running
FragmentPath=/lib/systemd/system/ssh.service
UnitFileState=enabled
...
```

Ver detalles de configuración del servicio:
Comando:

```
cat /etc/systemd/system/ssh.service
```

Salida simulada (fragmento):

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target

[Service]
ExecStart=/usr/sbin/sshd -D
ExecReload=/bin/kill -HUP $MAINPID
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

c) Analizar logs de eventos relacionados

Usando journalctl para un servicio específico (por ejemplo, ssh):
Comando:

```
journalctl -u ssh.service --since "2025-01-31 00:00:00"
```

Salida simulada:

```
-- Logs begin at 2025-01-30 08:12:15, end at 2025-02-01 15:47:30. --
Feb 01 09:15:32 ubuntu systemd[1]: Starting OpenBSD Secure Shell
server...
```

```
Feb 01 09:15:33 ubuntu sshd[1023]: Server listening on 0.0.0.0 port
22.
Feb 01 09:15:33 ubuntu sshd[1023]: Server listening on :: port 22.
Feb 01 09:15:33 ubuntu systemd[1]: Started OpenBSD Secure Shell
server.
Feb 01 10:25:47 ubuntu sshd[1156]: Accepted password for user from
192.168.1.50 port 53421 ssh2
...
```

Resumen y Análisis

- **Listado de servicios/daemons:**
  Se observa que ambos sistemas listan servicios activos, aunque en Linux se distinguen además los "unit files" para ver qué servicios están habilitados al inicio.
- **Parámetros y configuración:**
  Mientras que en Windows se utiliza `Get-Service` y `Get-WmiObject` para obtener detalles (estado, modo de inicio, dependencias), en Linux se usa `systemctl show` y se inspecciona el archivo de configuración del servicio (por ejemplo, en `/etc/systemd/system/`).
- **Logs de eventos:**
  Windows utiliza el cmdlet `Get-EventLog` para extraer entradas de eventos del sistema, y Linux utiliza `journalctl` para obtener logs asociados a un servicio. La forma de presentación y el detalle de la información pueden diferir.

Estos ejemplos de salidas de pantalla ayudan a comprender cómo se gestionan y visualizan los servicios/daemons en cada sistema operativo, facilitando la comparación y el análisis de la influencia de estos servicios en los logs de eventos.

Nota: Las salidas aquí presentadas son simuladas y sirven únicamente para ilustrar el resultado esperado al ejecutar los comandos descritos en el trabajo práctico. En un entorno real, los resultados variarán según la configuración del sistema, los servicios instalados y el estado operativo del mismo.