Analisis de rendimiento de sistemas windows

notas de interés

- curso ofrecido por: https://segall.es (Miguel Angel)
- powershell basic cheat sheet

procesos

- un programa es una secuencia estática de instrucciones, mientras que un **proceso** es un contenedor para un conjunto de recursos utilizado para ejecutar un programa.
- PID = ID de proceso
  - al menos 1 hilo de ejecución, cada hilo tiene acceso al contenedor del proceso
  - espacio de dirección virtual privado
  - programa ejecutable
  - lista de handles abiertos (recursos de sistema)
  - contexto de seguridad (token de acceso)
- Ejemplos PowerShell listado de procesos:
  - `Get-Process` similar a `tasklist`
  - `Get-Process | OGV` redirección a programa OGV[1], permite filtrar la salida en modo GUI, reconoce las columnas, para manipularlas
  - `GPS | Select Name,VirtualMemorySize64 | OGV`
  - columnas:
    - Handles: número handles
    - NPM(K): non-paged memory
    - PM(K): pagelable memory
    - WS(K): working
    - VM(M)
    - CPU(s)
    - ID
- nombre, PID, identidad, ejecutable y parámetros, recursos
- `svchost` → DLLs y procesos de sistema, se asocian a un ejecutable.
- un hilo es la unidad básica a la que el sistema operativo asigna el tiempo del procesador
- handle (agarre) del objeto, es una entrada a una tabla con los recursos del objeto
- Monitor de recursos: `perfmon /res`
- `Get-Process | Get-Member | Out-Host -Paging` Todos los miembros de un objeto desde PS[2]

servicios

- procesos de windows que arrancan automaticamente
- menos servicios = arranque más rápido
  - de encriptación
  - de indexación
  - Windows Error Reporting
  - Windows Media Center
  - Distributed link shared client (o algo parecido) - cliente de seguimiento de vínculos distribuidos
  - updates de programas (Dell, Intel, Nero, Google, Skype, Adobe...)
  - Family Safety
  - Fax
- `tasklist /svc` o `sc.exe queryex state=all` o PS: `Get-Service` o `Get-WMIObject`

```
Win32_Service
```

Arranque

- `msconfig.exe`
- sysinternals: `autoruns`

windows update

- `Get-Service wuau*` → saber si está en ejecución
- `Get-Hotfix | Sort-Object InstalledOn | OGV`

variables de entorno

- `Get-Variable`
- `$ENV:<variable>`
- `Get-ChildItem env:` → lista todas las variables de `$env`

hardware

- obtener información del sistema:
  - `PS: GWMI Win32_PhysicalMemory`
  - `PS: GWMI Win32_LogicalDisk`
  - `PS: GWMI Win32_BIOS`
  - `PS: GWMI Win32_Processor`

Memoria

- memoria física
  - no es una constante! → hot-swap o memoria dinámica en máquina virtual
  - `PS: GWMI Win32_PhysicalMemory`
  - `PS: (gwmi Win32_ComputerSystem).TotalPhysicalMemory`
  - `PS: [System.Math]::Pow(2,32) / 1GB`
- memoria virtual
  - archivo de paginación: pagefile.sys
  - de 1 a 3 veces la memoria usada cuando crees que está a un alto rendimiento ¿?
  - estar atento al espacio libre en el disco donde esté ese fichero
  - `dir c:\ -Force -Include *.sys` → ?
  - variables de procesos vinculadas con la memoria
    - Private Bytes (PB): aproximación razonable de la cantidad de memoria que su ejecutable está utilizando
    - Working Set (WS): tamaño actual en bytes del conjunto de trabajo de un proceso
      - = Private Bytes no paginados + archivos mapeados en memoria
    - Virtual Bytes (VB): tamaño actual en bytes...
      - = WS + PB paginados + lista espera

discos

- defragmentación
    - `pagedefrag.exe` (sysinternals)
    - `contig.exe` (sysinternals)
    - `defrag.exe`
    - `JKDefrag` (open software)
    - conveniente realizar defragmentación
        - `perfmon /res` también se puede ver el proceso en discos

tareas

- `PS: Get-ScheduledTask ScheduledDefrag | fl *` → ??

servicios

- search:
    - `Get-Service *search*`
    - W10: no mira por defecto dentro de los archivos
- antivirus:
- programador de tareas

important

- `perfmon /rel` → historial de lo que ha pasado en el equipo (errores de aplicación, windows, varios, advertencias, información)
- `perfmon /res` → recursos (CPU, Memoria, Red, Disco)
- `autoruns` (sysinternals)

otros

- sysinternals https://docs.microsoft.com/ca-es/sysinternals/
    - herramientas administrativas de administración avanzada, diagnóstico y solución de problemas
    - intuitivas y fácil de usar
    - se empaquetan como una sola imagen ejecutable
    - no dejan rastro
    - ejemplos destacados:
        - Process Explorer
        - Autoruns
        - Process Monitor
        - VMMap (memoria virtual y física)
        - DebugView
        - ProcDump
        - Ps*
            - PsExec
            - PsFile
            - PsGetSid
            - PsInfo
            - ...

1)

Out-GridView

2)

PowerShell