Gestión de procesos críticos en Windows

Administración de Sistemas Operativos

Cristóbal Suárez Abad – 2º ASIR

Contenido

Introducción: 2

1) Muestra la lista de procesos en ejecución e identifica los principales procesos del sistema. (desde el PowerShell como administrador) 4

2) Realiza un filtrado de procesos por usuario, nombre o identificador, registrando los resultados más relevantes. 6

a) Filtrado por el nombre de usuario: 6

b) Filtrado por nombre del proceso: 6

c) Filtrado por ID del proceso: 6

3) Observa qué procesos consumen más CPU o memoria y describe posibles causas. .. 7

4) Seleccionar 3 procesos del sistema y documentar (services.exe, System, wininit.exe): 8

5) Vamos a filtrar procesos desde el powerShell: 9

6) Lanza un proceso manualmente, analiza su prioridad y cámbiala para comprobar el efecto en el rendimiento (Inicia la compresión de un archivo grande con 7-zip). Para ello cambia la prioridad a “Alta”. 11

7) Finaliza el proceso de forma controlada y observa la mejora en el rendimiento. 13

8) Lanza un proceso manualmente, analiza su prioridad y cámbiala (ejecutando notepad.exe desde PowerShell). Realiza cambios de prioridad y finalízalo documentando los efectos. 15

9) Documentar el ciclo de vida completo de un proceso. 17

10) ¿Qué es svchost.exe?, buscalo en el administrador de tareas. ¿Para que sirve? ¿Qué relación guarda con los virus? 18

Introducción:

El administrador de un servidor Windows detecta que el sistema se vuelve lento en ciertos momentos. Debes analizar los procesos activos, identificar los más críticos y experimentar con la creación, prioridad y terminación de un proceso para entender su ciclo de vida.

1. Muestra la lista de procesos en ejecución e identifica los principales procesos del sistema. (desde el PowerShell como administrador)

2. Realiza un filtrado de procesos por usuario, nombre o identificador, registrando los resultados más relevantes.

3. Observa qué procesos consumen más CPU o memoria y describe posibles causas.

*4. Seleccionar **3 procesos del sistema** y documentar (services.exe, System, wininit.exe):*

- *PID*
- *Usuario propietario*
- *Estado aproximado (activo, en espera)*
- *Memoria y CPU utilizada*

Verifica la existencia de estos 3 procesos usando el administrador de tareas. ¿pueden finalizarse? ¿qué consecuencias tendría?

5. *Vamos a filtrar procesos desde el powerShell:*

-
    - *lista todos los procesos de tu usuario (tenfras que usar IncludeUserName y las variables de entorno de powershell).*
    - *obtener todos los datos disponibles sobre los procesos que comienza por "Sear" y "MsM"*
    - *obtener todos los módulos cargados por tu navegador.*

6. Lanza un proceso manualmente, analiza su prioridad y cámbiala para comprobar el efecto en el rendimiento (Inicia la compresión de un archivo grande con 7-zip). Para ello cambia la prioridad a “Alta”

7. Finaliza el proceso de forma controlada y observa la mejora en el rendimiento.

8. Lanza un proceso manualmente, analiza su prioridad y cámbiala (ejecutando notepad.exe desde PowerShell). Realiza cambios de prioridad y finalízalo documentando los efectos.

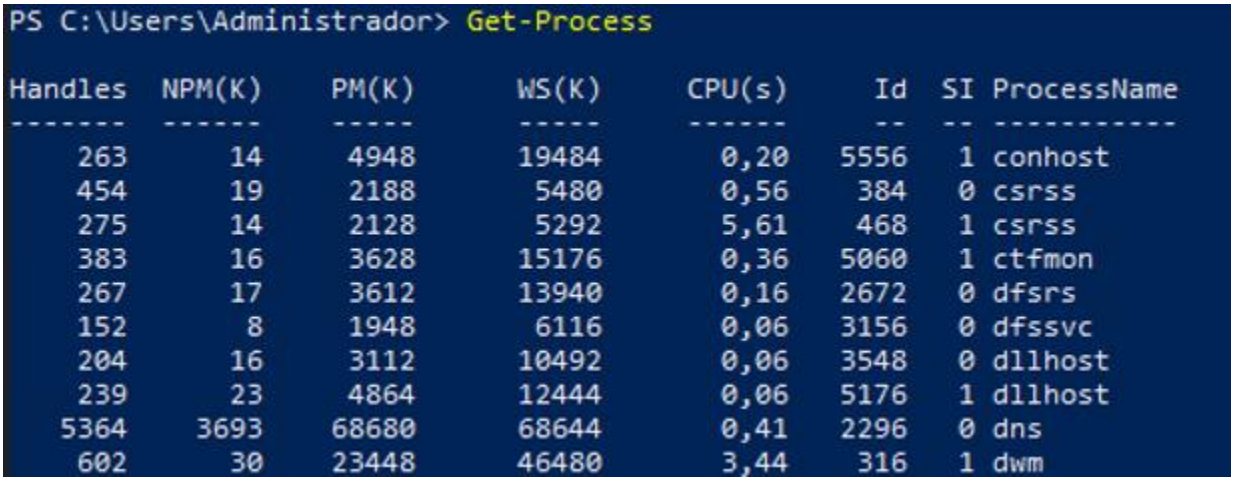
9. Documentar el ciclo de vida completo de un proceso

10. ¿Qué es svchost.exe?, buscalo en el administrador de tareas. ¿Para qué sirve? ¿Qué relación guarda con los virus?

1) Muestra la lista de procesos en ejecución e identifica los principales procesos del sistema. (desde el PowerShell como administrador)

Podemos usar:

Get-Process

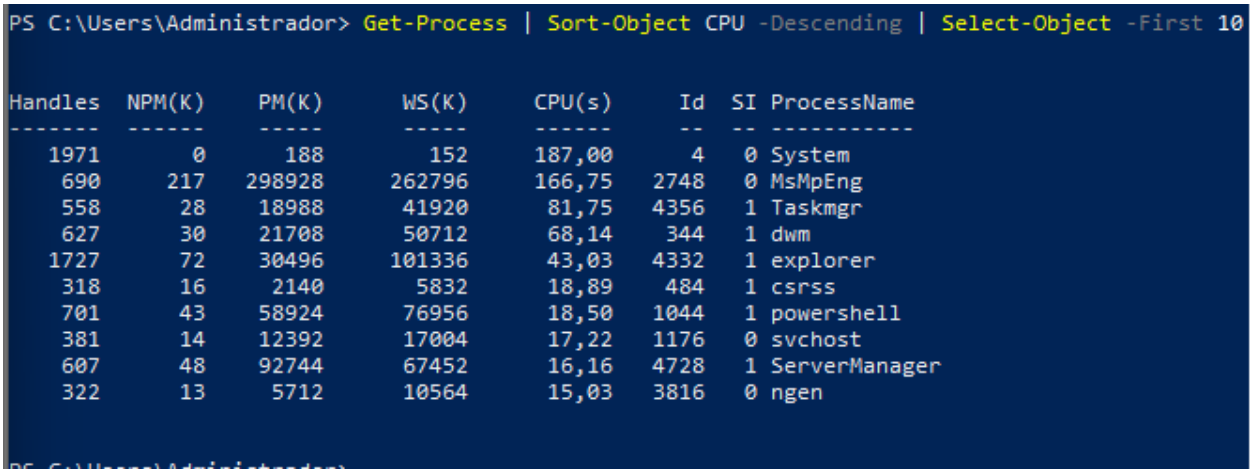


```
PS C:\Users\Administrador> Get-Process

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    263      14     4948      19484       0,20    5556   1 conhost
    454      19     2188       5480       0,56     384   0 csrss
    275      14     2128       5292       5,61     468   1 csrss
    383      16     3628      15176       0,36    5060   1 ctfmon
    267      17     3612      13940       0,16    2672   0 dfsrs
    152       8     1948       6116       0,06    3156   0 dfssvc
    204      16     3112      10492       0,06    3548   0 dllhost
    239      23     4864      12444       0,06    5176   1 dllhost
   5364    3693    68680      68644       0,41    2296   0 dns
    602      30    23448      46480       3,44     316   1 dwm
```

O el siguiente, el cual nos muestra los 10 procesos que más CPU consumen:

Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

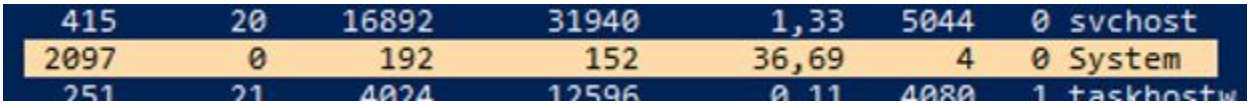


```
PS C:\Users\Administrador> Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
   1971       0      188        152     187,00       4   0 System
    690     217   298928     262796     166,75    2748   0 MsMpEng
    558      28    18988      41920      81,75    4356   1 Taskmgr
    627      30    21708      50712      68,14     344   1 dwm
   1727      72    30496     101336      43,03    4332   1 explorer
    318      16     2140       5832      18,89     484   1 csrss
    701      43    58924      76956      18,50    1044   1 powershell
    381      14    12392      17004      17,22    1176   0 svchost
    607      48    92744      67452      16,16    4728   1 ServerManager
    322      13     5712      10564      15,03    3816   0 ngen
```

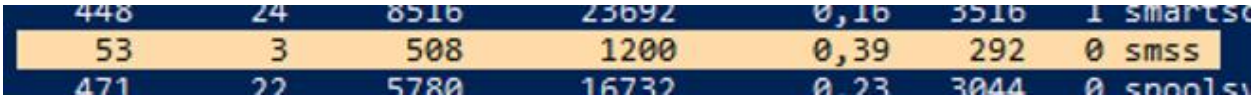
Los principales procesos son:

- System: Controla operaciones del kernel (no se puede detener).

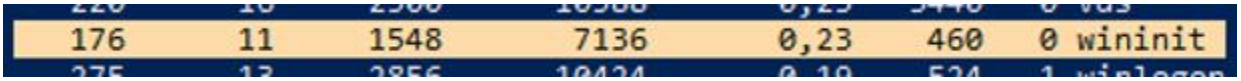


```
   415      20    16892      31940       1,33    5044   0 svchost
  2097       0      192        152      36,69       4   0 System
   251      21     4024      12596       0,11    4080   1 taskhostw
```

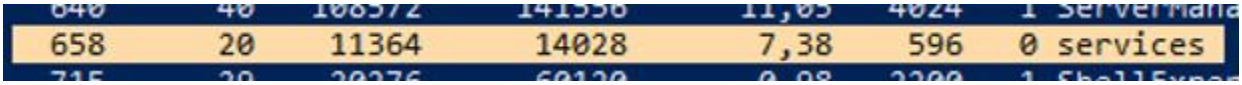
- smss: Gestor de sesiones del sistema.



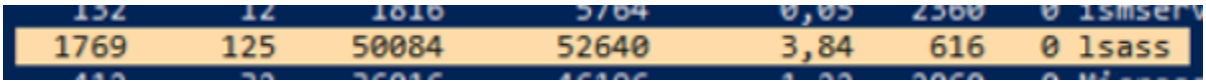
- wininit: Inicializa procesos del sistema durante el arranque.



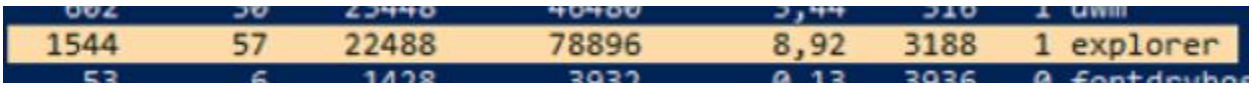
- services: Administra los servicios del sistema.



- lsass: Autenticación de usuarios (Local Security Authority).



- explorer: Interfaz gráfica del usuario (escritorio, menú inicio).



2) Realiza un filtrado de procesos por usuario, nombre o identificador, registrando los resultados más relevantes.

a) Filtrado por el nombre de usuario:

*Get-Process -IncludeUserName | Where-Object {$_.UserName -like "*Administrador*"}*

```
PS C:\Users\Administrador> Get-Process -IncludeUserName | Where-Object {$_.UserName -like "*Administrador*"}

Handles      WS(K)   CPU(s)     Id UserName               ProcessName
-------      -----   ------     -- --------               -----------
    263      16224     1,25   5556 ASIR_SUAREZ\Adminis... conhost
    384      15168     0,36   5060 ASIR_SUAREZ\Adminis... ctfmon
    230      12340     0,06   5176 ASIR_SUAREZ\Adminis... dllhost
   1467      43720     9,06   3188 ASIR_SUAREZ\Adminis... explorer
    837      67920     2,14   5544 ASIR_SUAREZ\Adminis... powershell
    493      39468     9,45   2724 ASIR_SUAREZ\Adminis... RuntimeBroker
    249      12816     0,23   3800 ASIR_SUAREZ\Adminis... RuntimeBroker
    250      18184     0,05   4524 ASIR_SUAREZ\Adminis... RuntimeBroker
   1162     152084    13,66   4104 ASIR_SUAREZ\Adminis... SearchUI
    621      82868    11,05   4024 ASIR_SUAREZ\Adminis... ServerManager
    801      66136     1,30   2200 ASIR_SUAREZ\Adminis... ShellExperienceHost
    476      25028     3,13   2144 ASIR_SUAREZ\Adminis... sihost
    426      23576     0,16   3516 ASIR_SUAREZ\Adminis... smartscreen
    407      30048     0,31   4092 ASIR_SUAREZ\Adminis... svchost
    293      14208     0,23   4420 ASIR_SUAREZ\Adminis... svchost
    247      12808     0,11   4080 ASIR_SUAREZ\Adminis... taskhostw
    275       6660     5,89   5008 ASIR_SUAREZ\Adminis... VBoxTray
```

b) Filtrado por nombre del proceso:

Get-Process | Where-Object {$_.ProcessName -like "*vbox*"}

```
PS C:\Users\Administrador> Get-Process | Where-Object {$_.ProcessName -like "*vbox*"}

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    158      10     2348       6756       0,22    1292   0 VBoxService
    273      13     2920       4948       5,89    5008   1 VBoxTray
```

c) Filtrado por ID del proceso:

Get-Process -Id 5544

```
PS C:\Users\Administrador> Get-Process -Id 5544

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    809      30    68540      45004       2,80    5544   1 powershell
```

3) Observa qué procesos consumen más CPU o memoria y describe posibles causas.

Procesos por consumo de CPU:

Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

```
PS C:\Users\Administrador> Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
   1848       0      192        152      38,41       4   0 System
    894     223   291932     235688      20,66    3104   0 MsMpEng
   1162      71    83392     152084      13,66    4104   1 SearchUI
    619      39   108356      47588      11,05    4024   1 ServerManager
    483      23    21296      39480       9,45    2724   1 RuntimeBroker
    679      75    21012      32240       9,19    2792   0 svchost
   1443      55    20324      25520       9,11    3188   1 explorer
    587      14     5324      12836       7,38     596   0 services
    272      15     2128       5292       6,41     468   1 csrss
    522      21    17476      31688       6,36    2348   0 svchost
```

Procesos por consumo de RAM:

Get-Process | Sort-Object WS -Descending | Select-Object -First 10

```
PS C:\Users\Administrador> Get-Process | Sort-Object WS -Descending | Select-Object -First 10

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    893     223   291932     249124      20,70    3104   0 MsMpEng
   1162      71    83392     152084      13,66    4104   1 SearchUI
      0       7     2180      70496       2,81      88   0 Registry
   5374    3688    68508      69252       0,48    2296   0 dns
    775      30    22592      65536       1,30    2200   1 ShellExperienceHost
   1700     123    52488      54976       4,31     616   0 lsass
    614      31    24668      52040       4,58     316   1 dwm
    619      39   108356      47440      11,05    4024   1 ServerManager
    480      32    35908      46332       1,22    2060   0 Microsoft.ActiveDirectory.WebServices
    790      30    68500      41572       3,09    5544   1 powershell
```

Causas del consumo:

El **alto uso de CPU del proceso "System"** indica actividad del kernel (drivers, disco o red).

MsMpEng.exe (Windows Defender) es **el mayor consumidor de CPU y memoria** → probablemente está ejecutando un análisis en segundo plano.

SearchUI.exe y **ServerManager.exe** también destacan, lo que es normal en servidores activos o recién configurados.

svchost.exe y **services.exe** son procesos contenedores de servicios críticos de Windows: su consumo puntual suele deberse a actualizaciones o ejecución de tareas programadas.

4) Seleccionar 3 procesos del sistema y documentar (services.exe, System, wininit.exe):

- PID
- Usuario propietario
- Estado aproximado (activo, en espera)
- Memoria y CPU utilizada

Verifica la existencia de estos 3 procesos usando el administrador de tareas. ¿pueden finalizarse? ¿qué consecuencias tendría?

- System: El Kernel se detendría y sufriría un pantallazo azul.

```
PS C:\Users\Administrador> Get-Process | Where-Object {$_.ProcessName -like "System*"}

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
   1856       0      192        144      45,64       4   0 System
```

- wininit: Provocaría un reinicio forzado.

```
PS C:\Users\Administrador> Get-Process | Where-Object {$_.ProcessName -like "wininit*"}

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    172      11     1392       7124       0,23     460   0 wininit
```

- services: El sistema perdería todos los servicios (red, sonido, etc.)

```
PS C:\Users\Administrador> Get-Process | Where-Object {$_.ProcessName -like "*services*"}

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    500      31    35924      47208       1,23   2060   0 Microsoft.ActiveDirectory.WebServices
    587      14     5428      12880       9,08    596   0 services
```

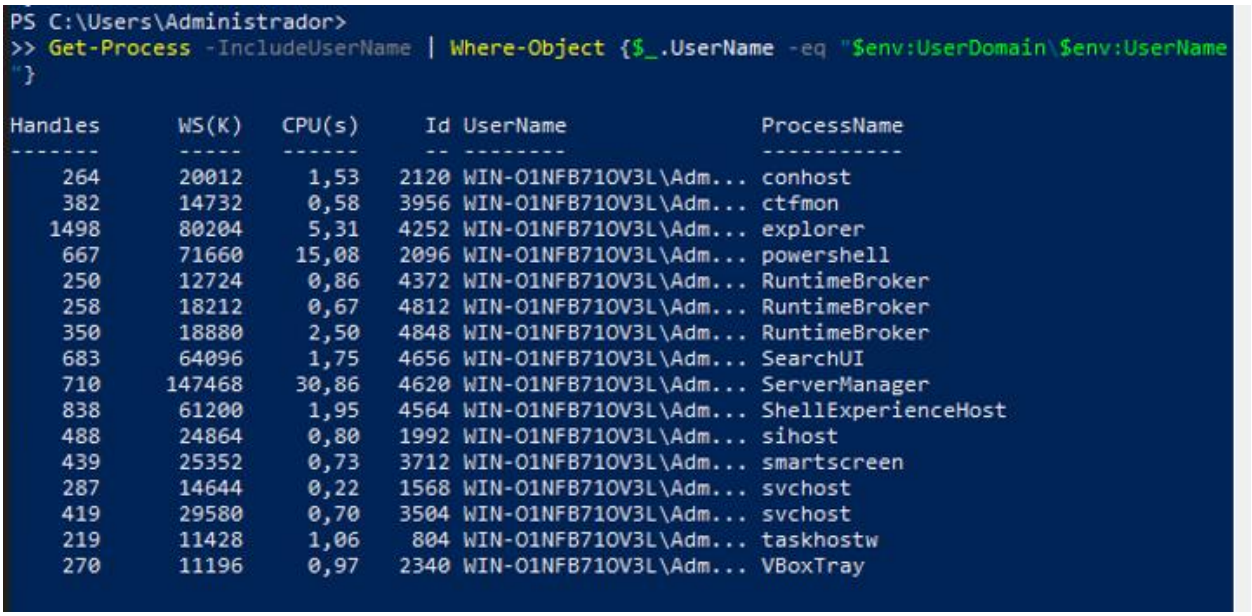
¿Pueden finalizarse?

No. Terminar System, services.exe o wininit.exe puede provocar pantallazo azul (BSOD), cierre del sistema o inestabilidad.

5) Vamos a filtrar procesos desde el powerShell:

- lista todos los procesos de tu usuario (tenfras que usar IncludeUserName y las variables de entorno de powershell).

Get-Process -IncludeUserName | Where-Object {$_.UserName -eq "$env:UserDomain\$env:UserName"}

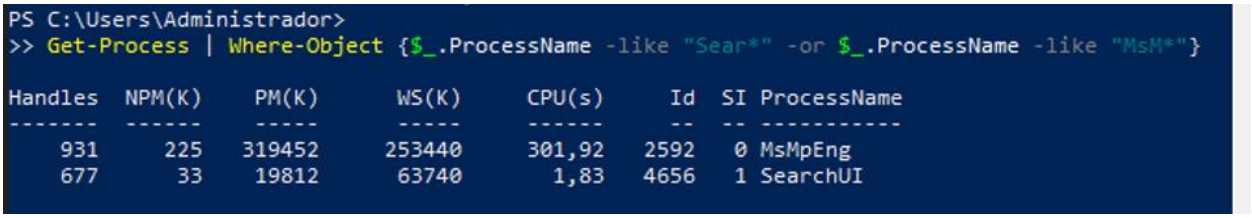


```
PS C:\Users\Administrador>
>> Get-Process -IncludeUserName | Where-Object {$_.UserName -eq "$env:UserDomain\$env:UserName
"}

Handles      WS(K)   CPU(s)     Id UserName               ProcessName
-------      -----   ------     -- --------               -----------
    264      20012     1,53   2120 WIN-O1NFB71OV3L\Adm... conhost
    382      14732     0,58   3956 WIN-O1NFB71OV3L\Adm... ctfmon
   1498      80204     5,31   4252 WIN-O1NFB71OV3L\Adm... explorer
    667      71660    15,08   2096 WIN-O1NFB71OV3L\Adm... powershell
    250      12724     0,86   4372 WIN-O1NFB71OV3L\Adm... RuntimeBroker
    258      18212     0,67   4812 WIN-O1NFB71OV3L\Adm... RuntimeBroker
    350      18880     2,50   4848 WIN-O1NFB71OV3L\Adm... RuntimeBroker
    683      64096     1,75   4656 WIN-O1NFB71OV3L\Adm... SearchUI
    710     147468    30,86   4620 WIN-O1NFB71OV3L\Adm... ServerManager
    838      61200     1,95   4564 WIN-O1NFB71OV3L\Adm... ShellExperienceHost
    488      24864     0,80   1992 WIN-O1NFB71OV3L\Adm... sihost
    439      25352     0,73   3712 WIN-O1NFB71OV3L\Adm... smartscreen
    287      14644     0,22   1568 WIN-O1NFB71OV3L\Adm... svchost
    419      29580     0,70   3504 WIN-O1NFB71OV3L\Adm... svchost
    219      11428     1,06    804 WIN-O1NFB71OV3L\Adm... taskhostw
    270      11196     0,97   2340 WIN-O1NFB71OV3L\Adm... VBoxTray
```

- obtener todos los datos disponibles sobre los procesos que comienza por "Sear" y "MsM"

Get-Process | Where-Object {$_.ProcessName -like "Sear*" -or $_.ProcessName -like "MsM*"}



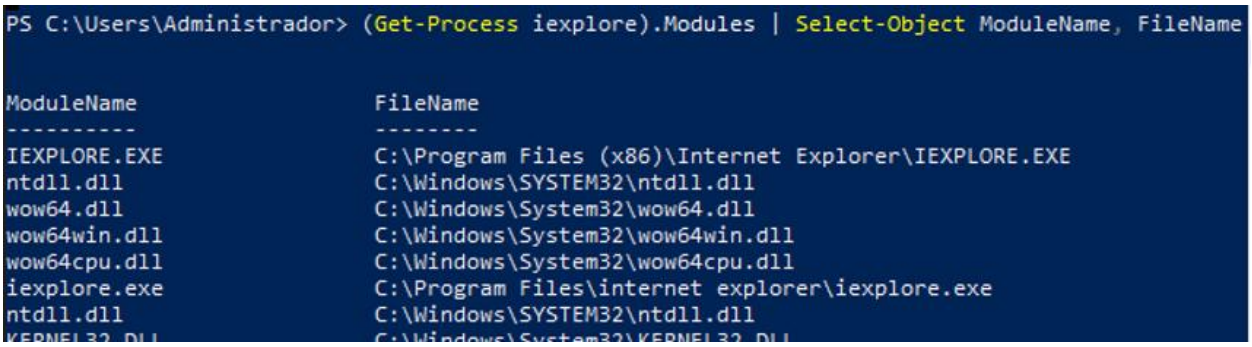
```
PS C:\Users\Administrador>
>> Get-Process | Where-Object {$_.ProcessName -like "Sear*" -or $_.ProcessName -like "MsM*"}

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    931     225   319452     253440     301,92    2592   0 MsMpEng
    677      33    19812      63740       1,83    4656   1 SearchUI
```

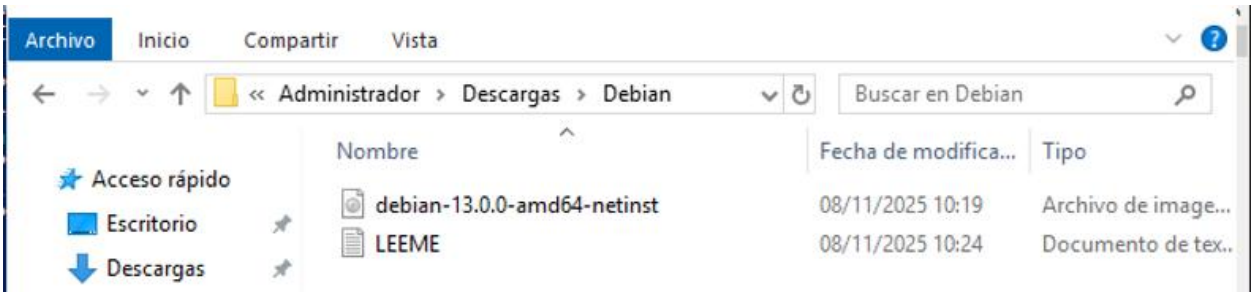
- Obtener todos los módulos cargados por tu navegador.

(Get-Process iexplore).Modules | Select-Object ModuleName, FileName

(Get-Process chrome).Modules | Select-Object ModuleName, FileName

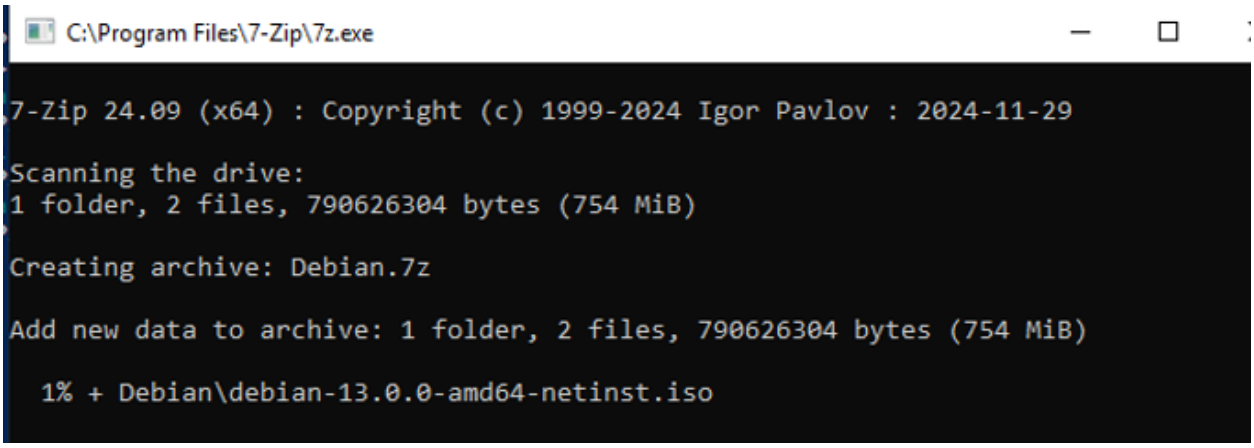


6) Lanza un proceso manualmente, analiza su prioridad y cámbiala para comprobar el efecto en el rendimiento (Inicia la compresión de un archivo grande con 7-zip). Para ello cambia la prioridad a “Alta”.



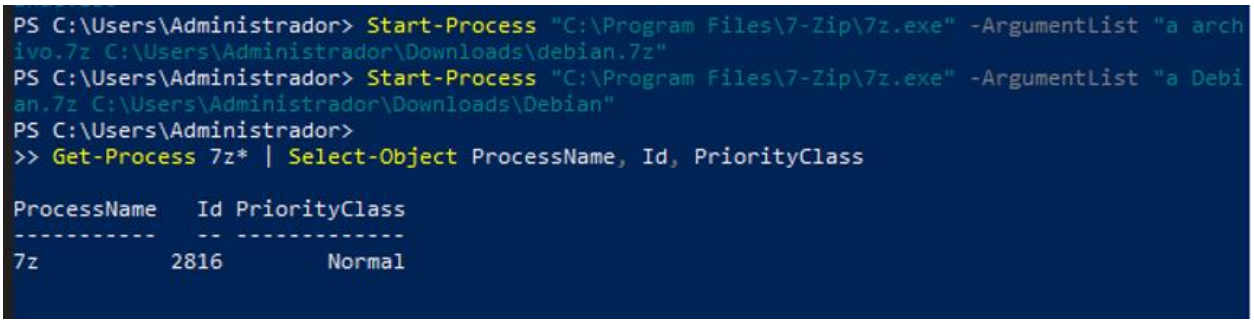
- Ejecuta la compresión de un archivo grande con 7-Zip:

Start-Process "C:\Program Files\7-Zip\7z.exe" -ArgumentList "a Debian.7z C:\Users\Administrador\Downloads\Debian"



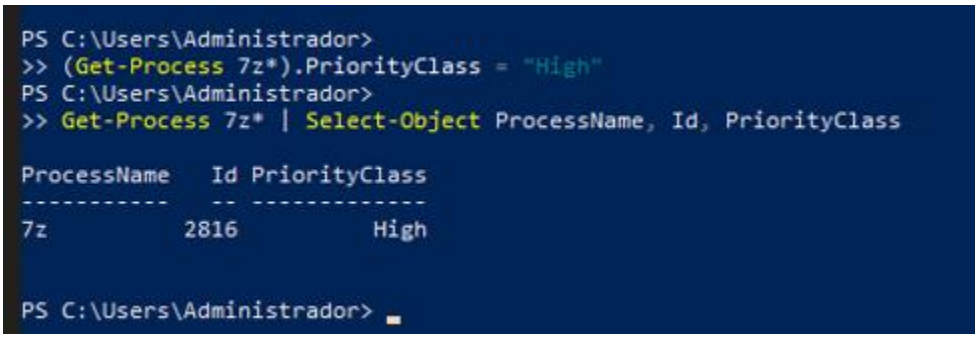
Ver prioridad:

Get-Process 7z* | Select-Object ProcessName, Id, PriorityClass



Cambiar prioridad a “Alta”:

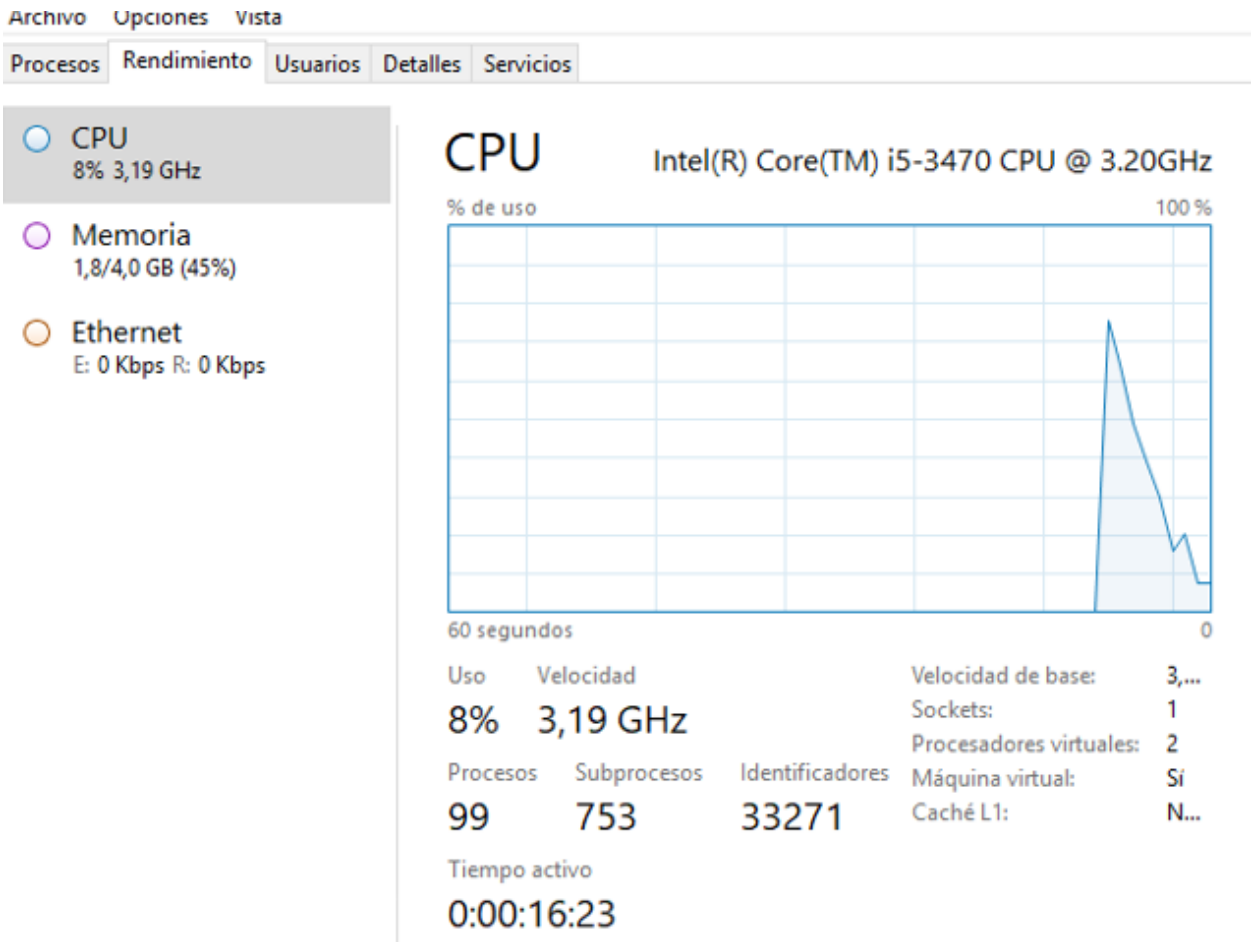
(Get-Process 7z*).PriorityClass = "High"



Efecto: el proceso tendrá más acceso a CPU, puede hacer que el sistema se vuelva menos fluido si es muy exigente.

7) Finaliza el proceso de forma controlada y observa la mejora en el rendimiento.

Antes de parar:



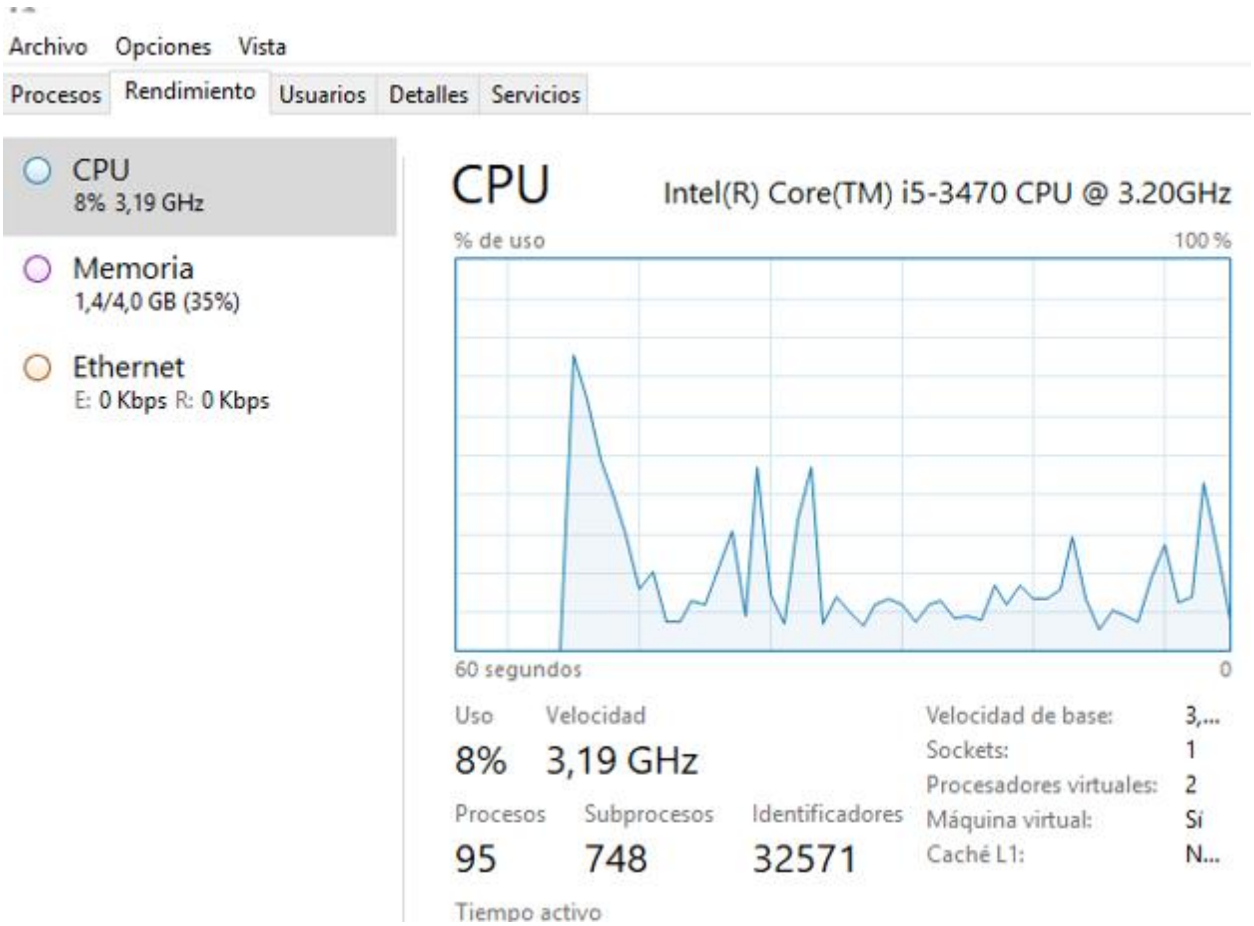
Paramos:

Stop-Process -Name "7z" -Confirm

```
PS C:\Users\Administrador>
>> Stop-Process -Name "7z" -Confirm

Confirmar
¿Está seguro de que desea realizar esta acción?
Se está realizando la operación "Stop-Process" en el destino "7z (2816)".
[S] Sí  [O] Sí a todo  [N] No  [T] No a todo  [U] Suspender  [?] Ayuda
(el valor predeterminado es "S"):S
PS C:\Users\Administrador>
```

Después de parar:

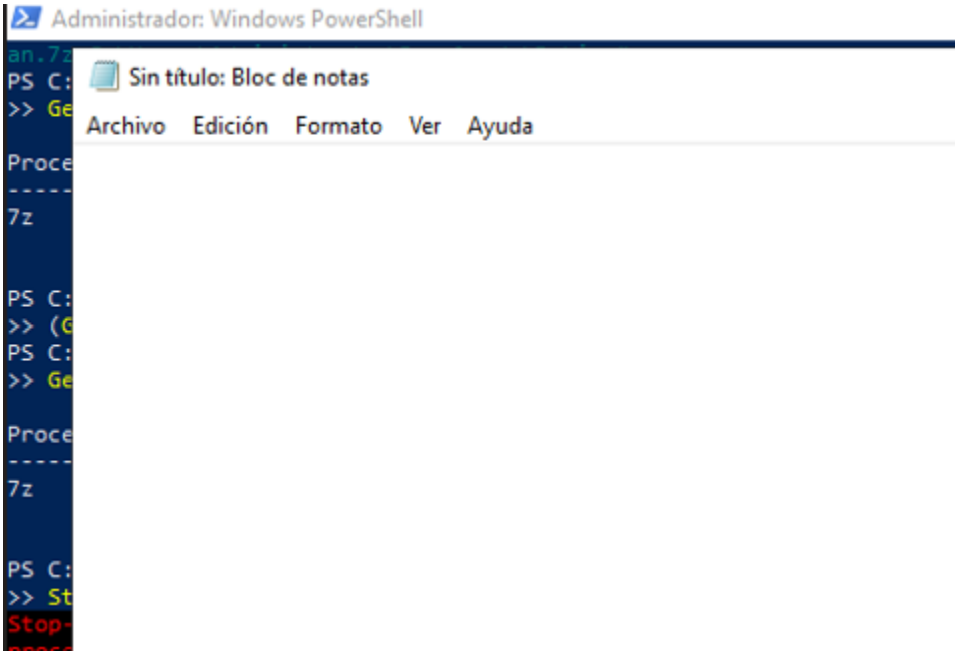


Observaciones: Se reduce el consume de CPU y RAM.

8) Lanza un proceso manualmente, analiza su prioridad y cámbiala (ejecutando notepad.exe desde PowerShell). Realiza cambios de prioridad y finalízalo documentando los efectos.

Start-Process notepad.exe

```
PS C:\Users\Administrador> Start-Process notepad.exe
```



Ver prioridad:

Get-Process notepad | Select-Object Id, PriorityClass

```
PS C:\Users\Administrador> Get-Process notepad | Select-Object Id, PriorityClass

  Id PriorityClass
  -- -------------
3008        Normal
```

Cambiarla:

(Get-Process notepad).PriorityClass = "High"

```
PS C:\Users\Administrador> (Get-Process notepad).PriorityClass = "High"
PS C:\Users\Administrador> Get-Process notepad | Select-Object Id, PriorityClass

  Id PriorityClass
  -- -------------
3008          High
```

Finalizarlo:

Stop-Process -Name notepad -Force

```
PS C:\Users\Administrador> Stop-Process -Name notepad -Force
```

Efectos: con prioridad alta apenas se notan cambios, pero si el sistema está saturado, el Bloc de notas tendrá prioridad sobre otros procesos más lentos.

9) Documentar el ciclo de vida completo de un proceso.

Creación: se lanza mediante Start-Process o por el sistema.

Inicialización: se asignan recursos (PID, memoria, hilos).

Ejecución: el proceso realiza sus tareas.

Espera o suspensión: puede detenerse temporalmente esperando recursos.

Finalización: termina normalmente o se detiene manualmente (Stop-Process o al cerrar el programa).

Liberación: el sistema libera memoria, desasigna PID y recursos.

10) ¿Qué es svchost.exe?, buscalo en el administrador de tareas. ¿Para que sirve? ¿Qué relación guarda con los virus?

Nombre completo: Service Host Process

Función: carga y gestiona servicios de Windows (.dll) agrupados por tipo (red, sistema, etc.).

Ubicación legítima: C:\Windows\System32\svchost.exe

Relación con virus: muchos malware se disfrazan con este nombre, pero si se encuentran fuera de System32, pueden ser maliciosos.

Para verlo en PowerShell usamos:

Get-Process svchost | Select-Object Id, ProcessName, Path