

Unidad 4 - Tarea 7

Gestión de Procesos

Introducción

En esta tarea trabajaremos la gestión de procesos básica en el Sistema Operativo Windows tanto desde interfaz gráfica como desde línea de comandos CMD y PowerShell.

Utiliza capturas para demostrar la ejecución de los ejercicios.

Equipo recomendado

Una computadora con Windows 10 instalado.

Parte 1: Uso de la interfaz gráfica

Ejercicio 1. Cuestiones (4 puntos)

1. ¿Cuál es el atajo de teclado para acceder al administrador de tareas directamente?
2. ¿Cuál es el comando utilizado para abrir el administrador de tareas?
3. Explica **brevemente cada pestaña** del administrador de tareas
4. Abrir una pestaña del navegador e indicar qué usuarios tienen permisos sobre ella (obtener dicha información desde el administrador de tareas).
5. Abrir la calculadora de Windows y mostrar desde el administrador de tareas (*Detalles*): su PID, su estado, la CPU y la Memoria que consume. ¿Cuál es su prioridad?
6. Explicar cada una de las operaciones que se pueden realizar sobre cada proceso desde la pestaña *Detalles*. Nombra los estados que intervienen en la operación.
7. Ordenar los procesos del sistema por uso de Memoria y por su PID (dos capturas).
8. Mostrar cual es el proceso que más CPU consume y el que más Memoria.
9. Abrir un bloc de notas y cerrarlo desde el administrador de tareas.
10. ¿Cual es la diferencia entre un servicio y un proceso? ¿En qué estados ves que puede estar un servicio?

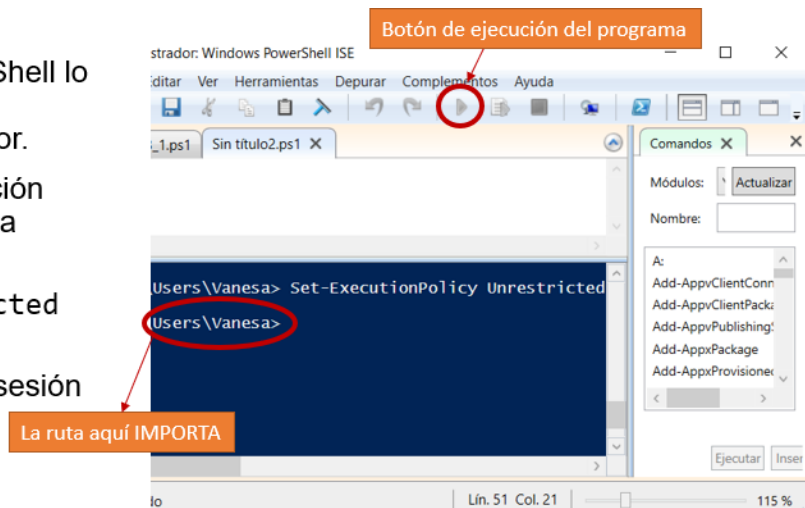
Parte 2: Uso de comandos

Ejercicio 1. Comandos en CMD (4 puntos)

1. Muestra la ayuda del comando tasklist
2. Muestra los procesos activos de tu usuario en formato tabla y después en formato csv.
3. Muestra los procesos activos cuyo uso de memoria sea mayor que 10 MB.
4. Abrir el bloc de notas y mostrar todos los módulos que utiliza al ejecutarse (archivos .dll)
5. Mostrar todos los procesos activos cuyo nombre empiece por "Ac"
6. Muestra la ayuda del comando taskkill
7. Termina aquellos procesos cuyo nombre comience por "chrome".
8. Termina aquellos procesos cuyo nombre comience por "Ac" y que consuman de 10MB a 15MB
9. Cierra el explorador de Windows y a continuación vuélvelo a iniciar. (**Cuidado, guardar todo antes de hacer esta operación**).
10. Cierra las tareas de Windows cuyo estado sea "no responde".

Ejercicio 2 Comandos en PowerShell (2 puntos)

- Para escribir programas en PowerShell lo mejor es abrir la consola **Windows PowerShell ISE** como administrador.
- **Importante:** dar permisos de ejecución cambiando la política de ejecución a Unrestricted.
`Set-ExecutionPolicy Unrestricted`
- Se **aconseja** volver a ponerla a Restricted una vez terminada la sesión de trabajo.



Escribir un programa llamado "ejercicio2.ps1" (ELEGID UNA RUTA DE GUARDADO ADECUADA, no system32) **debidamente documentado** que realice:

- En vuestra carpeta de trabajo (la elegida antes), crear un directorio llamado *procesos* (comando New-Item con opción -Force para sobrescribir en cada ejecución).
- Entrar en el directorio (comando Set-Location)
- Mediante una única orden crear un archivo para todos los procesos que comiencen por **a** (usar la opción -Force para sobrescribir en New-Item). Nombre de archivo: nombre del proceso. Contenido archivo: nombre, identificador y tiempo de CPU del proceso.
- Mostrar por pantalla el contenido de cada fichero en carpeta *procesos*.
- MOSTRAR: captura del programa y del resultado de ejecución.
 - ★ Ayuda del comando New-Item:
<https://learn.microsoft.com/en-gb/powershell/module/microsoft.powershell.management/new-item?view=powershell-5.1>
 - ★ Ayuda: ejemplo de usar pipe para trabajar con el resultado del Get-Process. Recuerda que el % es el alias de un bucle foreach donde \$_ representa al objeto proceso que estamos visitando dentro del bucle. En este ejemplo estamos mostrando por pantalla:
Get-Process | %{Write-Host -Nombre (\$_.Name) -CPU (\$_.CPU)}
 - ★ Ayuda: ejemplo para acceder a elementos de un objeto \$_ siendo \$_ el resultado de Get-Process:
(\$_ | Select-Object Name, CPU, id)
 - ★ Ayuda: para obtener todos los ficheros del directorio actual usamos **Get-ChildItem** y para mostrar el contenido de un fichero usamos **Get-Content**

Importante: ENTREGAR EL ARCHIVO PS1 EN LA TAREA para poder probarlo, en otro caso, NO SE CORREGIRÁ