

Práctica 1

Procesos y servicios

Objetivos

- Comprender la información mostrada por el *Administrador de tareas* acerca de los procesos y aplicaciones que se encuentran en ejecución en el sistema en un momento dado.
- Comprender el concepto de proceso de sistema y conocer los procesos de sistema fundamentales que se ejecutan normalmente en una plataforma Windows.
- Comprender la diferencia entre proceso y programa.
- Entender cómo la CPU se reparte entre los múltiples procesos que se encuentran en ejecución en cada momento.
- Entender cómo los procesos interactúan con el usuario a través de ventanas.
- Entender la diferencia entre una ventana primaria y secundaria.
- Comprender el concepto de servicio.
- Aprender a utilizar la herramienta Servicios, para llevar a cabo la gestión básica de los servicios del sistema.

Desarrollo de la práctica

1 Procesos

Un proceso puede definirse como un programa en ejecución. Los sistemas operativos multitarea, como Windows, permiten la ejecución simultánea de múltiples procesos. En estos sistemas, la CPU se va repartiendo entre los diferentes procesos, de modo que todos ellos se vayan ejecutando en “tiempo compartido”. Para observar y llevar el control de los procesos que se encuentran en ejecución en el sistema, Windows proporciona una herramienta denominada *Administrador de tareas*. Ahora, utilizando esta herramienta, vamos a observar algunos procesos muy importantes que están presentes en todo sistema Windows 2000.

Procesos de sistema

Durante el proceso de arranque del computador, el sistema operativo, una vez cargado, pone en ejecución un conjunto de procesos que realizan funciones vitales dentro del

sistema. Estos procesos reciben el nombre de *procesos de sistema*. Vamos a observar estos procesos con el *Administrador de tareas*.

H Asegúrate de que no tienes ninguna aplicación abierta en el sistema. Ahora abre el *Administrador de tareas* y elige la ficha *Procesos*. En este momento observarás todos los procesos que se están ejecutando actualmente en el sistema.

Como en este momento no tienes ninguna aplicación abierta, todos los procesos que observas son procesos de sistema.

Para cada proceso se muestran varios campos de información. Lo normal es que los campos mostrados sean los siguientes¹:

- **Nombre de imagen:** Es el nombre del programa ejecutable desde el que se cargó el proceso.
- **Nombre de usuario:** Es la cuenta de usuario utilizada por el proceso en su ejecución. Dicha cuenta establece el entorno de seguridad del proceso.
- **CPU:** Indica el porcentaje de CPU consumida por un proceso.
- **Uso de memoria:** Es la cantidad de memoria utilizada por el proceso.

Otro campo que puede visualizarse mediante el *Administrador de tareas* es el **PID**. Se trata de un número entero que identifica al proceso dentro del sistema. A cada proceso se le asigna un número diferente. PID es el acrónimo de *Process Identifier*.

H Si el *Administrador de tareas* no muestra el PID, reconfigúralo para que lo muestre utilizando la opción *Seleccionar columnas* de su menú *Ver*.

Ahora vamos a reconocer algunos de estos procesos de sistema. Comenzaremos por el *Proceso inactivo del sistema*. Este proceso es el que se ejecuta cuando la CPU no tiene trabajo útil que llevar a cabo.

H Indica a continuación el identificador de este proceso y el porcentaje de CPU utilizado por él:

–Pregunta 1–

PID:

Porcentaje de uso de la CPU:

H ¿Qué conclusión sacas del porcentaje de CPU utilizado por este proceso? Escríbela a continuación.

–Pregunta 2–

¹ Puede que en tu sistema los campos que aparezcan no sean exactamente estos, ya que los campos mostrados son configurables.

Otros procesos críticos del sistema son el SMSS, el CSRSS y WINLOGON. El SMSS, conocido como *Session Manager*, es el que controla toda una sesión de trabajo, desde que el sistema se inicia hasta que se apaga. El cometido del CSRSS es algo más complejo de explicar, así que no entraremos en él, pero es un proceso fundamental que está siempre en ejecución. Finalmente, WINLOGON es el que controla las sesiones de usuario. Así por ejemplo, este proceso controla la ventana en la que introducimos nuestro nombre de usuario y clave para comenzar una sesión en el sistema.

H Asegúrate de que has observado estos tres procesos en el *Administrador de tareas*. Quédate con sus nombres porque siempre estarán presentes en toda sesión de trabajo.

Vamos a ver ahora otro proceso fundamental que es el *explorer.exe*². Este proceso controla la interfaz gráfica con el usuario, proporcionando así una de las funciones principales del sistema operativo Windows. Vamos a eliminar este proceso a ver que ocurre.

H Pulsa con el botón derecho sobre el proceso *explorer.exe*. Elige la opción *Terminar proceso*. ¿Qué ocurre? Observarás que desaparecen todos los iconos del *Escritorio*, así como la *Barra de tareas* y el menú *Inicio*. Al eliminar este proceso hemos cortado prácticamente toda posibilidad de comunicación con el usuario.

Una opción para continuar trabajando con el sistema es eliminar la presente sesión de usuario y poner en marcha una nueva sesión. Esto podemos hacerlo sin problemas porque quien controla las sesiones de usuario es el proceso WINLOGON, que sigue activo. Este proceso se activa con la pulsación de Ctrl-Alt-Supr.

H Pulsa Ctrl-Alt-Supr. WINLOGON se activa mostrando la ventana *Seguridad de Windows*, que es la que usas para lanzar el *Administrador de tareas*, pero que también te permite cerrar la sesión de usuario. Elige *Cerrar sesión*. Ahora comienza como siempre una nueva sesión. Observa que todo vuelve a estar como estaba. WINLOGON ha dirigido todo este proceso. Al crearse una nueva sesión de usuario se ha cargado de nuevo el *explorer.exe* y por tanto se ha habilitado una nueva interfaz con el usuario para atender a nuestras peticiones.

H Abre el *Administrador de tareas* y comprueba que el *explorer.exe* está de nuevo en ejecución.

Hasta aquí hemos visto los siguientes procesos de sistema: proceso inactivo, SMSS, CSRSS, WINLOGON y *explorer.exe*. Como puedes observar en el *Administrador de tareas*, hay otros procesos de sistema. Más adelante veremos alguno de ellos.

Proceso frente a programa

Un programa, en su versión ejecutable, es un conjunto de instrucciones máquina y datos almacenados en un fichero. En Windows los programas ejecutables se nombran con la extensión “.exe”. A la versión ejecutable de un programa se le llama también la *imagen binaria* del programa. Un ejemplo de programa ejecutable es el *Bloc de notas*. Veamos ahora cuál es su imagen binaria.

H Abre el menú *Inicio*, *Todos los programas*, *Accesorios*. Entonces observarás la opción *Bloc de notas*. Pincha sobre ella con el botón derecho y elige la opción

² No confundir con el Internet Explorer: el Explorer y el Internet Explorer son cosas diferentes

Propiedades. En la ficha *General* observas en el campo *Tipo de archivo* que el elemento del menú *Bloc de notas* es en realidad un acceso directo. En la ficha *Acceso directo* puedes ver el programa al que hace referencia dicho acceso. Se trata de *notepad.exe*. Es decir, este es el programa (imagen binaria) que se cargará en memoria y se ejecutará cuando pulsemos sobre la opción *Bloc de notas*.

Tenemos entonces una imagen binaria de un programa, denominada *notepad.exe*, que se encuentra almacenada en algún lugar del sistema de disco del ordenador. Ahora vamos a poner esa imagen binaria en ejecución, convirtiéndola por tanto en un proceso.

H Abre el *Administrador de tareas*, ficha *Procesos*. Ahora vas a ordenar los procesos por el nombre de su imagen binaria. Pulsa sobre *Nombre de imagen*. Observarás que los procesos quedan ordenados por nombre. Ejecuta el *Bloc de notas* y observa cómo aparece el nombre de su imagen (*notepad.exe*) en el *Administrador de tareas*.

En este momento el programa *notepad.exe* se ha convertido en un proceso porque lo hemos puesto en ejecución en el sistema. Ahora veremos cómo a partir de un único programa podemos generar tantos procesos como queramos, sin más que ejecutar el programa múltiples veces.

H Anota a continuación el PID del proceso *notepad.exe* que acabas de poner en ejecución.

–Pregunta 3–

PID =

Vuelve a ejecutar el *Bloc de notas*. Observarás que un nuevo proceso *notepad.exe* ha sido puesto en ejecución. Anota el PID de este nuevo proceso.

–Pregunta 4–

PID =

A partir de un único programa hemos generado dos procesos diferentes. El sistema operativo los diferencia porque ha asignado un PID distinto a cada uno de ellos. Podrías seguir generando tantos procesos *notepad.exe* como quisieses.

Este último experimento habrá servido para dejar bien clara la diferencia entre programa y proceso.

Uso de la CPU por parte de los procesos

Es habitual que los procesadores actuales dispongan de más de un núcleo (*core*) de procesamiento. En concreto, los procesadores de los ordenadores del Laboratorio disponen de dos núcleos. Cada núcleo representa una CPU completa y, por tanto, nuestros ordenadores cuentan con dos CPUs. El número de núcleos activos en un sistema puede observarse mediante el *Administrador de tareas*.

H Abre el *Administrador de tareas*. Elige la ficha *Rendimiento*. En la zona de esta ficha identificada como *Historial de uso de la CPU* se muestran tantos recuadros como núcleos activos hay en el sistema. En este momento debes observar dos recuadros, que corresponden a los dos núcleos del procesador.

Para comprender la planificación de procesos en la CPU es mejor empezar experimentando con un sistema que tenga una sola CPU. Esto podemos conseguirlo haciendo que nuestro sistema funcione con un solo núcleo activo. Para ello, tenemos

que modificar un fichero del sistema operativo en el que se especifican diversos aspectos relativos a la carga e inicialización del sistema operativo. Se trata del fichero `boot. ini`. En nuestro sistema, este fichero se encuentra en la carpeta raíz de la unidad C: . De momento no lo ves, porque es un fichero oculto del sistema.

H Para poder manipular este archivo debes configurar el explorador de Windows para que permita ver archivos protegidos del sistema. Para ello debes abrir una ventana de exploración de archivos y en el menú *Herramientas*, elegir *Opciones de carpeta*. Posteriormente elige la ficha *Ver* y en ella deberás configurar adecuadamente dos opciones para que se muestren los archivos ocultos del sistema.

H Una vez que observes `boot. ini` en la carpeta raíz de la unidad C: debes eliminar el atributo *Sólo lectura*, para que se pueda modificar.

H Abre el fichero con el *Bloc de notas*.

Este fichero contiene dos secciones: *boot loader* y *operating systems*. Cada línea de la sección *operating systems* corresponde a un sistema operativo instalado en el sistema. En nuestro caso observarás una línea correspondiente al Windows Server 2003 instalado en nuestro sistema. En la parte final de esta línea se indican las opciones de arranque. Estas se separan mediante un espacio en blanco y van precedidas del carácter '/'. Para indicar el número de núcleos con los que va a trabajar el sistema se utiliza la opción '/numproc='.

H Añade la opción '/numproc=1' a `boot. ini`. Recuerda que debe estar separada de las otras opciones mediante un espacio. Usando esta opción ordenamos al sistema que funcione con un solo núcleo.

H Arranca de nuevo el sistema. Abre el *Administrador de tareas*, ficha *Rendimiento*. Entonces debes observar que en la zona *Historial de uso de la CPU* solo hay un recuadro, que corresponderá al único núcleo activo.

Ahora realizaremos una serie de experimentos con esta configuración del sistema y luego dejaremos las cosas como estaban inicialmente. Para hacer las pruebas de planificación vamos a utilizar tres programas, que están disponibles en la carpeta de la asignatura. Se trata de los programas `prog1-1.exe`, `prog1-2.exe` y `prog1-3.exe`.

H Copia estos programas, así como sus correspondientes ficheros fuente (`prog1-1.c`, `prog1-2.c` y `prog1-3.c`) en el escritorio de tu ordenador.

Empezaremos las pruebas con `prog1-1.exe`. Se trata de un programa que ejecuta dos bucles `for` anidados. El bucle interior no ejecuta ninguna sentencia. El bucle exterior sólo ejecuta el bucle interior. Cada bucle incrementa una variable de tipo entero desde 0 hasta 200000000. Esto asegura que el programa tarda mucho tiempo en ejecutarse.

H Abre `prog1-1.c` con el *Bloc de notas* y asegúrate de que comprendes su código. Observarás que es un programa que no hace E/S y que por tanto utiliza la CPU de forma intensiva.

H Ahora vas a observar la ejecución de este programa con el *Administrador de tareas*. Abre el *Administrador de tareas*. Observa que en este momento la CPU está siendo ocupada, casi al cien por cien, por el *Proceso inactivo*. Ejecuta `prog1-1.exe`. Observarás que se abre una ventana en la que no hay ningún tipo de actividad. Esto es debido a que el proceso no realiza ninguna operación de E/S. ¿Qué ocurre con el

uso de la CPU? Indica a continuación el porcentaje de CPU que utilizan en este momento el *Proceso inactivo* y *prog1- 1. exe*.

-Pregunta 5-

% de CPU proceso inactivo:

% de CPU proceso prog1-1.exe:

H Conmuta a la ficha *Rendimiento* del Administrador de tareas y observa que el uso de la CPU se encuentra al 100%.

H Retorna a la ficha *Procesos*. Entonces pulsando con el botón derecho del ratón sobre el proceso *prog1- 1. exe*, termina su ejecución, comprobando cómo la CPU pasa de nuevo a ser ocupada por el *Proceso inactivo*.

Ahora vamos a comprobar cómo la CPU se reparte entre los procesos en ejecución. Para ello vamos a lanzar varias ejecuciones del programa *prog1- 1. exe*.

H Abre el *Administrador de tareas*. Pulsa sobre *Nombre de imagen* para ordenar los procesos por nombre. Ahora vas a poner varias imágenes del programa *prog1- 1. exe* en ejecución. A continuación anotarás el porcentaje de CPU que se asigna a cada proceso correspondiente a este programa cuando hay una, dos, tres y cuatro imágenes de él en ejecución. Es posible que tras lanzar cada imagen tengas que esperar un poco de tiempo hasta que se establezca la planificación.

-Pregunta 6-

Una imagen en ejecución. % de CPU asociado a la imagen:

Dos imágenes en ejecución. % de CPU asociado a cada imagen:

Tres imágenes en ejecución. % de CPU asociado a cada imagen:

Cuatro imágenes en ejecución. % de CPU asociado a cada imagen:

H Termina la ejecución de todos estos procesos utilizando el botón derecho del ratón y la opción *Terminar proceso*.

H ¿Qué mecanismo utiliza el sistema operativo para que un programa como *prog1- 1. exe* no monopolice totalmente la CPU cuando se ejecuta?

-Pregunta 7-

El programa *prog1- 1. exe* no es un programa de estructura computacional típica, ya que no hace operaciones de E/S. Ahora vamos a continuar las pruebas con el programa *prog2- 2. exe*, que tiene una estructura computacional más común, es decir, computa y hace E/S alternativamente.

H Abre *prog1- 2. c* con el *Bloc de notas* y analiza su código.

La parte de este programa que usa intensivamente la CPU está formada por dos bucles *for* anidados. El bucle interior incrementa la variable *j* de 0 a 200000000. El bucle exterior incrementa la variable *i* hasta que ésta iguale a la variable *iter*.

Este programa hace también E/S, ya que el usuario debe introducir por consola el valor de la variable *iter*. La E/S es realizada por las funciones *printf()* y *scanf_s()*, que pertenecen a la librería estándar del lenguaje C. *printf()* envía a la consola la cadena "Numero de iteraciones: " y *scanf_s()* recoge el número que el usuario

introduce por teclado, almacenándolo en la variable `iter`. La clave está precisamente en `scanf_s()`. Cuando esta función se ejecuta, el proceso queda bloqueado esperando la entrada de datos del usuario. Mientras esto ocurre el proceso no consume CPU.

Tanto los bucles `for` anidados como las funciones de E/S se encuentran dentro de un bucle `while`, que se ejecuta infinitamente, por lo que podremos repetir el proceso de E/S y computación indefinidamente. Mientras se produce la espera en la función `scanf_s()` se puede pulsar Ctrl-C, lo que rompe la ejecución del programa.

H Abre el *Administrador de tareas* y elige su ficha *Procesos*. Ejecuta `prog1-2.exe`. En este momento observarás que el proceso no consume CPU, ¿por qué?

–Pregunta 8–

H Introduce 10 iteraciones. ¿Qué ocurre con el consumo de CPU y por qué?

–Pregunta 9–

H Elige la ficha *Rendimiento* en el *Administrador de tareas* y observa el historial de uso de la CPU mientras el programa ejecuta 5, 10, 15 y 20 iteraciones.

Terminaremos las pruebas de planificación utilizando el programa `prog1-3.exe`.

H Abre `prog1-3.c` con el *Bloc de notas* y analiza su código.

Este programa también genera un uso intensivo de la CPU mediante dos bucles `for` anidados. Para observar la evolución de la computación, en el bucle exterior se envía un mensaje a la pantalla indicando el número de la iteración.

Antes y después de la ejecución de los bucles anidados se toman tiempos, utilizando la función `time()`, que pertenece a la librería estándar del lenguaje C. El programa finalmente imprime la diferencia de los tiempos capturados. Dicha diferencia es el tiempo que tardan los bucles anidados en ejecutarse.

H Ejecuta `prog1-3.exe` y anota a continuación el tiempo que tarda en ejecutarse.

–Pregunta 10–

H Lanza ahora dos ejecuciones simultáneas de `prog1-3.exe` y anota a continuación el tiempo que tardan en ejecutarse.

–Pregunta 11–

H Finalmente lanza cuatro ejecuciones simultáneas de `prog1-3.exe` y anota el tiempo que tardan en ejecutarse.

–Pregunta 12–

H Son coherentes los resultados que has obtenido en los experimentos anteriores. Explica por qué.

–Pregunta 13–

Finalmente modificaremos de nuevo el fichero `boot. ini` para que el sistema funcione con todos su núcleos activos, que es la situación habitual.

H Abre el fichero `boot. ini` con el *Bloc de notas*. Elimina en él la opción `'/numproc=1'`. Asegúrate de no dejar espacios en blanco al final de la línea correspondiente. Salva el fichero. Ahora arranca de nuevo el sistema. Abre el *Administrador de tareas* y usando la ficha *Rendimiento* comprueba que los dos núcleos vuelven a estar activos en el sistema. Para ello debes observar que en la zona *Historial de uso de la CPU* hay dos recuadros, correspondiendo cada uno de ellos a un núcleo del sistema.

Haremos ahora una última prueba de planificación con los dos núcleos activos.

H Abre el *Administrador de tareas*, ficha *Rendimiento*. Ejecuta el programa `prog1- 1. exe`. Observarás que el uso de la CPU no sube hasta el 100%, sino sólo hasta el 50%, ¿por qué? Escribe a continuación tu respuesta. Si tienes dudas pregúntale al profesor.

–Pregunta 14–

H Finalmente vuelve a marcar el archivo `boot. ini` como de solo lectura y configura el explorador de Windows para que los archivos ocultos del sistema no se muestren. Comprueba que has realizado esto correctamente, observando que el fichero `boot. ini` ya no se muestra.

Interfaz de los procesos con el usuario: procesos GUI

Una gran parte de los procesos que se ejecutan en un computador necesitan intercambiar información con los usuarios. Por ejemplo, un programa como el *Bloc de notas* muestra al usuario la información almacenada en un documento de texto y recibe comandos del usuario a través de un conjunto de menús. Así se observa que para realizar el intercambio de información entre procesos y usuarios se utilizan unos objetos, denominados *ventanas*, que se muestran sobre el escritorio del sistema. Según su forma de interacción con el usuario existen dos tipos de programas Windows: los programas de tipo GUI (Graphical User Interface) y los de tipo CUI (Console User Interface). Nos centraremos ahora en los programas GUI.

Los programas GUI utilizan ventanas gráficas para llevar a cabo su proceso de entrada/salida con el usuario. Estas ventanas usan múltiples tipos de elementos para interaccionar con el usuario, tales como *menús*, *barras de herramientas*, *cuadros de diálogo*, *paletas de selección*, etc. Un ejemplo de programa GUI es el WordPad, que es un editor de texto, distribuido junto con el sistema operativo, y que es bastante más potente que el *Bloc de notas*. Vamos a ver algunos conceptos sobre ventanas utilizando el WordPad como ejemplo.

H Abre el WordPad, al que llegarás mediante los menús *Inicio*, *Programas*, y *Accesorios*. Al abrir el programa se muestra sobre el escritorio una ventana, que es sobre la que se llevará a cabo el principal proceso de interacción con el programa. Esta ventana recibe el nombre de *ventana primaria*.

La parte superior de la ventana primaria se encuentra delimitada por la barra de título, de color azul, en la que se observan los botones *Minimizar*, *Restaurar* y *Cerrar* (a la derecha) y un icono indicativo del programa que abrió la ventana y el título de ésta (a la izquierda).

H Escribe a continuación el título de la ventana primaria abierta por el WordPad.

–Pregunta 15–

Las ventanas primarias tienen una propiedad muy importante: cuando un proceso abre una ventana primaria, se inserta un botón en la *Barra de tareas* de la interfaz de usuario para manejar dicha ventana. El botón se marca con el mismo icono y título que el que aparece en la barra de título de la ventana primaria.

H Observa el botón de la barra de tareas correspondiente al WordPad y comprueba que se marca con el mismo icono y título utilizados en la ventana primaria correspondiente.

Tras presentar el concepto de ventana primaria, podremos seguir explorando la información proporcionada por el *Administrador de tareas*.

H Abre el *Administrador de tareas*. Como has abierto el WordPad, en la ficha *Procesos* observarás que el proceso *wordpad.exe* se encuentra en ejecución. Ahora pasa a la ficha *Aplicaciones*. Observarás una entrada en esta ficha. Escribe a continuación su nombre

–Pregunta 16–

¿Coinciden exactamente tus repuestas a las preguntas 15 y 16? Si no es así, puede que hayas cometido algún error. Pregúntale a tu profesor.

La conclusión es que lo que se muestra en la ficha *Aplicaciones del Administrador de tareas* y los botones insertados en la *Barra de tareas* de la interfaz de usuario son la misma cosa: ventanas primarias abiertas por los procesos en ejecución.

Con objeto de seguir analizando la relación entre procesos y ventanas primarias, vamos a poner otro programa en ejecución, el *Internet explorer*.

H Abre el *Internet explorer*. Utilizando las fichas *Procesos* y *Aplicaciones* del *Administrador de tareas* contesta:

–Pregunta 17–

Nombre del proceso que se pone en ejecución:

Nombre de la ventana primaria que se muestra:

H Observa que el nombre y el icono de título asignados a la nueva ventana abierta coinciden en la ficha *Aplicaciones* y en el botón insertado en la *Barra de tareas*.

Ahora vamos a ver cómo se puede conocer el proceso que corresponde a una ventana primaria determinada.

H Sitúate en la ficha *Aplicaciones* del *Administrador de tareas*. Pulsa con el botón derecho del ratón sobre la entrada correspondiente al *WordPad*. Elige la opción *Ir al proceso*. Esto te indica el proceso poseedor de la ventana. Vuelve a la ficha *Aplicaciones* y repite la operación con la entrada correspondiente al *Internet explorer*.

Hasta ahora hemos vistos cómo un proceso, como el *WordPad*, utiliza una ventana primaria para comunicarse con el usuario. Otros procesos, más complejos, pueden utilizar varias ventanas primarias, con objeto de proporcionar una mayor flexibilidad en su comunicación con el usuario. Veamos algún ejemplo de esto.

H Abre el *Administrador de tareas*. Elige la ficha *Procesos*. Abre el *Internet explorer*. Observarás cómo aparece el proceso *I EXPLORER. EXE* en el *Administrador de tareas*. Si ahora conmutas a la ficha *Aplicaciones*, observarás el nombre de la ventana primaria abierta por el *I EXPLORER. EXE*. Navega en el *Internet explorer* y observa cómo va cambiando el nombre de la ventana.

Con objeto de agilizar la navegación, el *Internet explorer* te permite manejar tantas ventanas primarias como desees. Ahora vas a abrir más ventanas:

H Mantén abierta la ficha *Aplicaciones* del *Administrador de tareas*. En el *Internet explorer*, abre el menú *Archivo* y elige la opción *Nuevo* y después *Ventana*. El navegador abre una nueva ventana primaria. Observa cómo aparece una nueva entrada correspondiente a la nueva ventana en el *Administrador de tareas*, así como en la *Barra de tareas*. Vuelve a repetir la misma operación para tener tres ventanas. Ahora puedes realizar alguna operación de navegación en cada una de ellas. Conmuta a la ficha *Procesos*. Ordena los procesos por el nombre de su imagen. Observa que hay un único proceso *I EXPLORER. EXE*. Este proceso es el que maneja las tres ventanas abiertas. Pulsa con el botón derecho del ratón sobre *I EXPLORER. EXE* y termina el proceso. Observa cómo se destruyen las tres ventanas correspondientes a *I EXPLORER. EXE*.

H Un resultado totalmente idéntico al anterior, desde el punto de vista del usuario, se obtiene si abres tres veces el *Internet explorer*. Hazlo. ¿Cuál es la diferencia entre ejecutar el proceso tres veces y abrir tres ventanas? Usa la ficha procesos para observar la diferencia. Escribe a continuación tu respuesta:

–Pregunta 18–

H Desde el punto de vista del uso de los recursos del computador, ¿qué estrategia te parece más apropiada, ejecutar varias veces el mismo programa o abrir varias ventanas desde un solo proceso? ¿Por qué?

–Pregunta 19–

Vamos a plantear otro ejercicio simple acerca de los procesos y las ventanas primarias manejadas por éstos.

H Abre el *Administrador de tareas*, ficha *Aplicaciones*. Ahora vas a realizar algunas operaciones típicas de navegación en el ordenador, que darán lugar a la apertura de ventanas primarias. Observa en el *Administrador de tareas* cómo se van registrando las ventanas abiertas. Abre *Mi PC* y vete a la unidad C: : obtienes así una ventana primaria. Ahora, Abre de nuevo *Mi PC*: obtienes otra ventana primaria. Finalmente, mediante *Menú de inicio*, opción *Buscar*, se abre otra ventana primaria. ¿A qué proceso o procesos pertenecen estas ventanas? (Averígualo usando el botón derecho del ratón en la ficha *Aplicaciones* del *Administrador de tareas*.)

–Pregunta 20–

Proceso propietario de la ventana C:l:

Proceso propietario de la ventana *Mi PC*:

Proceso propietario de la ventana *Resultado de la búsqueda*:

Como no podía ser de otra manera, todas estas ventanas pertenecen al proceso que gestiona la interfaz de usuario del sistema. Este proceso permite controlar múltiples aspectos del funcionamiento del sistema, utilizando múltiples ventanas primarias para intercambiar información con el usuario.

Hasta ahora sólo hemos hablado de ventanas primarias. Sin embargo, a partir de éstas se pueden abrir otras ventanas que reciben el nombre de *ventanas secundarias*. Mediante estas ventanas el usuario puede especificar parámetros u opciones del programa correspondiente. Una diferencia fundamental entre las ventanas primarias y secundarias es que a las primarias siempre se les asigna un botón en la *Barra de tareas*, mientras que a las secundarias, no. Veamos a continuación un ejemplo de uso de ventanas secundarias.

H Abre de nuevo el *WordPad*. Como ya hemos visto anteriormente se abre una ventana primaria mediante la que interaccionamos con este programa. Recuerda que

se inserta un botón en la *Barra de tareas* para dicha ventana. Despliega el menú *Formato* y elige la opción *Fuente*. Al ejecutar esta opción se abre una ventana secundaria. Observa cómo no se asigna ningún botón en la *Barra de tareas* para esta ventana, por ser secundaria. Una vez realizadas las operaciones necesarias con la ventana secundaria, ésta es cerrada por el usuario, retornándose el control a la ventana primaria. Las ventanas secundarias tienen, al igual que las primarias, un nombre en la barra de título, pero no tienen icono en dicha barra. Localiza cuatro ventanas secundarias en el WordPad y escribe a continuación sus nombres:

–Pregunta 21–

H Abre la ayuda del WordPad. ¿En qué tipo de ventana se muestra? Contesta a continuación:

–Pregunta 22–

Interfaz de los procesos con el usuario: procesos CUI

Los procesos de este tipo realizan la comunicación con el usuario en una venta de tipo texto. En este tipo de ventana no hay elementos gráficos. Todo el proceso de entrada/salida se lleva a cabo mediante cadenas de texto que se muestran sobre la ventana. Los programas CUI son muy apropiados para implementar comandos del sistema que llevan a cabo una determinada operación, muestran resultados sobre la operación realizada y terminan. Así el tiempo en que se encuentran en ejecución es mínimo. Al igual que los procesos GUI, los procesos CUI pueden ser ejecutados desde la interfaz de comandos del sistema (*explorer.exe*) mediante el comando ejecutar del *Menú de inicio*. Sin embargo, debido a su forma de operar y a que se ejecutan muy rápidamente, no es conveniente ejecutarlos desde el *explorer*, porque la ventana de texto que crean para mostrar los resultados desaparece casi instantáneamente. Vamos a probar esto usando el programa *ping.exe*. Se trata de un programa de tipo CUI que nos permite determinar si un equipo conectado a una *intranet* está “vivo”.

H Ahora vas a ejecutar *ping.exe* desde el *explorer*. Para ello, elige la opción *Ejecutar* del menú *Inicio* e introduce *ping.exe*. Observarás una especie de destello que corresponde a la apertura y cierre (casi instantáneos) de la ventana primaria correspondiente al proceso.

Debido a esto, para ejecutar programas CUI, se suele utilizar una interfaz de comandos especializada para dicho tipo de programas. Llamaremos a esta interfaz la *Interfaz de comandos textual* (ya que hace su E/S en una ventana de tipo texto.) Con seguridad ya habrás ejecutado múltiples veces esta interfaz, que se encuentra implementada mediante el programa *CMD.EXE*.

H Ejecuta mediante la opción *Ejecutar* del *Menú de inicio* el programa *CMD.EXE*. Observa en la ficha *Procesos* del *Administrador de tareas* que el proceso *CMD.EXE* se encuentra en ejecución. Observa en la ficha *Aplicaciones* que hay una ventana primaria asociada a *CMD.EXE*. Esta es la ventana de tipo texto en la que *CMD.EXE* realizará la interacción con el usuario. Ahora desde esta ventana podemos ejecutar programas CUI que realizarán su E/S textual en esta misma ventana. Vamos a ejecutar *ping.exe* para determinar si la máquina “*pin.epsig.uniovi.es*” se encuentra “viva”. Para ello ejecuta

```
C:\...>ping pin.epsig.uniovi.es
```

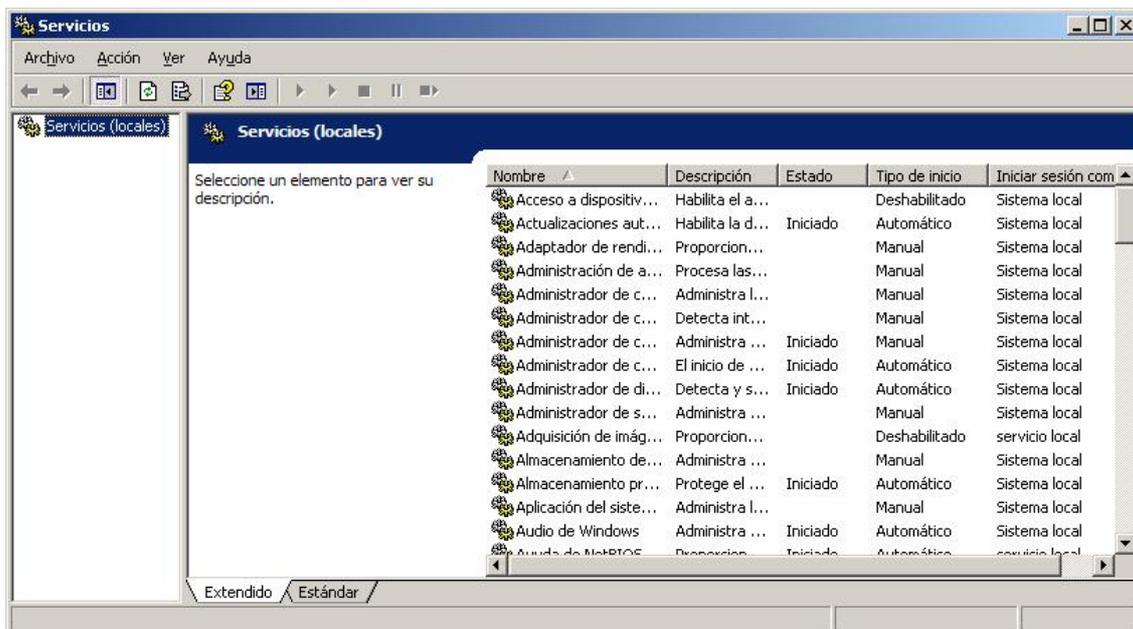
Ahora el programa `ping.exe` realizará su proceso de E/S en la ventana de la interfaz de comandos textual, mostrándonos si la máquina responde o no.

2 Servicios

En la sección anterior hemos trabajado con programas que interaccionan con el usuario a través de ventanas. En todo sistema operativo existen otro tipo de procesos que llevan a cabo funciones fundamentales en el sistema, pero que no requieren ningún tipo de interacción con el usuario. Estos programas suelen arrancarse automáticamente durante la inicialización del sistema y, en el entorno Windows, reciben el nombre de *servicios*. Como los procesos que implementan servicios no interaccionan con el usuario, no utilizan ventanas. Debido a ello, estos procesos pasan desapercibidos al usuario, sin embargo, se encuentran permanentemente en ejecución: se suele decir que se ejecutan en segundo plano (*background*).

En Windows, los servicios se implementan mediante procesos. Un proceso puede contener uno o varios servicios. Para controlar los servicios, se utiliza una herramienta de administración que se encuentra accesible desde el menú de *Herramientas administrativas*. Vamos a abrir entonces la herramienta de administración de servicios.

H Abre menú *Inicio* → *Herramienta administrativas* y selecciona *Servicios*. Se abrirá una ventana como la que se muestra a continuación:



La ventana abierta muestra en forma de tabla los servicios disponibles en el sistema. En el campo *Nombre* se indica el nombre del servicio; en el campo *Descripción* se comenta de forma concisa el cometido del servicio; el campo *Estado* indica si el servicio se encuentra o no en funcionamiento; y el campo *Tipo de inicio* determina la forma de puesta en marcha del servicio. El campo *Iniciar sesión como* será comentado en sesiones de prácticas futuras.

A modo de ejemplo, para ver cómo se gestionan los servicios, trabajaremos ahora con un servicio concreto, el servicio *Messenger*³. Primero vamos a obtener información general acerca de este servicio.

H Busca en la ventana *Servicios* el servicio *Messenger*. Pulsa sobre él con el botón derecho del ratón y elige *Propiedades*. En la ficha *General* se muestran las propiedades básicas del servicio. Así, se indica el nombre original del servicio, *Messenger*, una breve descripción del cometido del servicio, la ruta y el fichero ejecutable en el que se encuentra el servicio, el tipo de inicio y el estado en el que está el servicio.

H Indica a continuación el nombre del fichero ejecutable que contiene el servicio *Messenger*, su tipo de inicio, y el estado en el que se encuentra.

–Pregunta 23–

Fichero ejecutable:

Tipo de inicio:

Estado del servicio:

H Abre el *Administrador de tareas* y elige la ficha *Procesos*. Comprueba que el proceso que contiene el servicio *Messenger* se encuentra en ejecución.

Aunque *Messenger* no esté ahora en funcionamiento, *svchost.exe* alberga otros muchos servicios que sí están operativos en este momento, y esa es la razón por la que *svchost.exe* está en ejecución. De hecho, *svchost.exe* (que quiere decir *service host*) es el proceso que se utiliza para albergar la mayor parte de los servicios del sistema operativo Windows.

Si todos los servicios implementados por un determinado proceso se encuentran desactivados, no es necesario tener el proceso que implementa dichos servicios en ejecución. Por ejemplo, el proceso *SPOOLSV.EXE* implementa un único servicio: el denominado *spooler* o *Cola de impresión*. Por consiguiente, si detenemos el servicio *Cola de impresión*, el proceso *SPOOLSV.EXE* será sacado de ejecución porque ya no es necesario. Vamos a probar esto.

H Abre la ventana *Servicios* y localiza en ella el servicio *Cola de impresión*. Mantén la ventana *Servicios* abierta. Abre el *Administrador de tareas*, ficha *Procesos*. Ordena los procesos por el nombre de su imagen y localiza *SPOOLSV.EXE*. Pulsa con el botón derecho sobre el servicio *Cola de impresión* y elige la opción *Detener*. Según lo comentado anteriormente, deberás observar en la ficha *Procesos* cómo el proceso *SPOOLSV.EXE* es sacado de ejecución. Ahora haz la operación contraria, es decir, inicia el servicio *Cola de impresión* y observa cómo *SPOOLSV.EXE* vuelve a ser puesto en ejecución.

El servicio *Cola de impresión* es el que gestiona el envío de los trabajos de impresión a las impresoras. Si este servicio no está activo no se puede imprimir. No obstante, si en un determinado sistema no es necesario imprimir, se podría deshabilitar este servicio. Esto ahorraría aproximadamente 3 MB de memoria, que es lo que ocupa este proceso. **Un usuario avanzado puede ahorrar una estimable cantidad de recursos en su sistema, iniciando nada más que los servicios que sean estrictamente necesarios.** No

³ No confundir con el MSN Messenger, que es una herramienta de comunicación a través de Internet.

obstante, hoy en día, los ordenadores suelen andar bastante sobrados de memoria, así que, quizás, sea mejor “no complicarse la vida”.

Volveremos ahora sobre el servicio *Messenger*. Utilizaremos este servicio a modo ejemplo para comprobar cómo los servicios hacen trabajo útil en el sistema. Ahora mismo el tipo de inicio de este servicio debe ser *Deshabilitado*. Para poder utilizar este servicio debemos cambiar su tipo de inicio.

H Pasa el servicio a *Manual* y pulsa *Aceptar*. Esto hará que el servicio pueda ser iniciado o parado por el usuario a voluntad. Entonces pulsa con el botón derecho del ratón sobre este servicio e inícialo. Ahora vamos a comprobar que el servicio funciona.

El objetivo del servicio *Messenger* es recibir mensajes provenientes de otras máquinas conectadas a la red y presentarlos en la pantalla. Este servicio es normalmente utilizado por los administradores de redes de ordenadores para enviar mensajes a los usuarios.

Para enviar un mensaje desde una estación de trabajo a otra u otras estaciones, se utiliza el comando NET SEND, que normalmente se ejecuta desde la interfaz de comandos textual. El comando va seguido, primero, del nombre de la estación a la que se desea enviar el mensaje y, después, del mensaje en sí mismo. Así para enviar el mensaje “hola” al equipo ATC120 se debe ejecutar el siguiente comando:

```
NET SEND ATC120 hola
```

H Arranca el otro equipo disponible en tu mesa de trabajo. Una vez arrancado, abre la interfaz de comandos textual (cmd. exe) y envía mediante el comando NET SEND el mensaje “Hola” al equipo en el que estabas trabajando previamente. El mensaje es presentado en una ventana en el equipo receptor. Observa el título de la ventana: *Messenger Service*. Dicho título indica al usuario que la ventana ha sido abierta por el servicio *Messenger*. Cierra la ventana *Messenger Service*. Observa también que en la estación emisora del mensaje se indica que el mensaje ha sido enviado.

H Desactiva el servicio *Messenger* (en el ordenador en el que has hecho toda la práctica) utilizando el botón *Detener* de su ventana de propiedades. Ahora intenta enviar desde la otra máquina el mensaje “Hola2”. Observarás que el mensaje no puede ser enviado. En la máquina emisora se indica que ha producido un error al enviar el mensaje.

H Restaura de nuevo el *Tipo de inicio* del servicio *Mensajero* al estado *Deshabilitado*, con objeto de que quede en su estado original.

El servicio *Messenger* no es excesivamente relevante, pero sí hay otros servicios que hacen funciones esenciales para el correcto funcionamiento del sistema. Una parte de los procesos que observas en la ficha *Procesos* del *Administrador de tareas* son los procesos que implementan los servicios del sistema. No se encuentran en ejecución por capricho. Deben estar ahí para que el sistema funcione.

3 Ejercicios adicionales

E Determina cuatro servicios que sean proporcionados por el proceso *svchost.exe*.

–Pregunta 24–

E En este ejercicio se propone probar el funcionamiento de otro servicio del sistema. Se trata del servicio *Programador de tareas*.

El sistema operativo Windows, al igual que otros sistemas operativos, permite la ejecución de tareas temporizadas, es decir, tareas que se ponen en ejecución en un determinado momento del tiempo. Para programar las tareas que se desean ejecutar temporizadamente, Windows proporciona una utilidad denominada *Tareas programadas* a la que se accede a través del *Panel de control*. El servicio *Programador de tareas* es el encargado de gestionar las tareas temporizadas, poniéndolas en ejecución en el instante o instantes planificados. Este servicio basa su funcionamiento en el reloj del sistema.

Determina cuál es el proceso asociado al servicio *Programador de tareas*. Indícalo a continuación:

–Pregunta 25–

Comprueba que este servicio se encuentra iniciado, que es su estado normal. En el caso de que estuviese detenido, inícialo.

Ahora vas a programar una tarea para que se ejecute a una determinada hora. Luego Comprobarás que la tarea se ejecuta al alcanzarse la hora estipulada. Todo funcionará correctamente porque el servicio *Programador de tareas* se encuentra activo. Después desactivaremos este servicio, reprogramaremos la tarea anterior para que se ejecute un poco más tarde, y observaremos que la tarea no se pone en ejecución a la hora programada. Este experimento servirá para comprobar que la ejecución de tareas temporizadas depende de que el servicio *Programador de tareas* se encuentre activo.

Mediante la utilidad *Tareas programadas* se puede programar la ejecución temporizada de cualquier programa. Elegiremos, a modo de ejemplo, el *Bloc de notas*. Imagínate que queremos activar un aviso en el ordenador en una determinada fecha y hora. Podemos programar el *Bloc de notas* para que se ejecute en ese instante y nos muestre un texto con el aviso. Lo primero que tenemos que hacer es escribir el texto del aviso y almacenarlo en una carpeta.

Escribe, con el *Bloc de notas*, el texto que desees en un fichero denominado *avi so. txt* y almacénalo en la carpeta *C: \Temp* de tu sistema. Si esta carpeta no existe, créala previamente.

Ahora debes tener claro cuál es el nombre y la ruta del programa que desees ejecutar temporizadamente. En este caso, el programa es *NOTEPAD.EXE* (que es el ejecutable correspondiente al *Bloc de notas*). Para buscar la ruta, puedes utilizar el comando *Buscar* del menú *Inicio*. Anota la ruta a continuación:

-Pregunta 26-

Ahora ya estamos en condiciones de programar la ejecución temporizada del *Bloc de notas*.

Partiendo del *Panel de control* abre *Tareas programadas*. Para programar una nueva tarea, elige *Archivo, Nuevo, Tarea programada*. Dale un nombre cualquiera a la entrada que se genera, por ejemplo, **Avi so- i mportante**. Una vez creada la nueva tarea hay que programarla. Pulsa sobre ella: se abrirá una venta que te permitirá programar la tarea. Elige la ficha *Tarea*. En ella hay que rellenar los campos *Ejecutar* e *Iniciar en*.

En el campo *Ejecutar* hay que escribir el nombre del programa que se desea ejecutar temporizadamente junto con su ruta. Además, si el programa a ejecutar requiere algún parámetro, hay que indicarlo. En nuestro caso el parámetro es el fichero a abrir por el NOTEPAD.EXE, que es **aviso.txt**. Teniendo en cuenta estas indicaciones, rellena este campo.

Para saber el cometido del campo *Iniciar en*, coge la interrogación de la esquina superior derecha de la ventana y pulsa sobre dicho campo. Después rellénalo de la forma apropiada teniendo en cuenta la ayuda obtenida.

En el campo *Ejecutar como* se indica el usuario en cuyo ámbito se ejecutará la tarea, que por defecto es el usuario que crea la tarea. Sin embargo, hay que completar este campo con la clave de acceso del usuario. Pulsa sobre el botón *Establecer contraseña* para introducir dicha clave.

Ahora hay que establecer la planificación temporal para la ejecución de la tarea. Para ello elige la ficha *Programación*. Vamos a hacer que la tarea se ejecute sólo una vez. En el campo *Programar tarea* selecciona *Solo una vez*, y elige como hora de inicio un par de minutos más que la hora actual del computador (que puedes ver a la derecha de la barra de tareas).

Una vez introducida toda la información anterior, la tarea estará lista para ejecutarse. Cuando el reloj del sistema alcance la hora de inicio de la tarea, ésta se ejecutará.

Espera hasta que se ejecute la tarea para comprobar que has realizado correctamente la programación de la misma. Ten en cuenta que la tarea se ejecuta porque el servicio *Programador de tareas* está activo.

Finalmente comprobaremos que al desactivar el servicio *Programar tareas*, las tareas programadas no se ejecutan. Abre la ventana *Servicios* y detén el servicio *Programador de tareas*. Ahora reprograma la tarea **Avi so- i mportante** para que se ejecute dos minutos después de la hora que tenga el reloj del sistema. Espera hasta que sea esa hora y comprueba que la tarea no se ejecuta. Finalmente, vuelve a dejar el servicio *Programador de tareas* en el estado *Iniciado* y elimina la tarea *Aviso-importante* de la carpeta *Tareas programadas*.