

Sistemas operativos: Gestión
de entrada/salida.

TEMA 18

ABACUS NT

Oposiciones 2021

Índice

- 1. Introducción**
- 2. Dispositivos de entrada/salida**
 - 2.1. Dispositivos de bloques**
 - 2.2. Dispositivos de caracteres**
 - 2.3. Dispositivos de uso exclusivo**
 - 2.4. Dispositivos compartidos**
 - 2.5. Dispositivos virtuales**
- 3. Sistemas operativos: Gestión de E/S**
 - 3.1. Gestión software**
 - 3.1.1. Manejador de interrupciones**
 - 3.1.2. Controladores de dispositivos**
 - 3.1.3. Software de E/S independiente del dispositivo**
 - 3.1.4. Software de E/S de la capa de usuario**
 - 3.2. Gestión del Hardware: controladora**
 - 3.2.1. Canales de E/S. Conexión. Tipos de canales.**
 - 3.3. Técnicas de gestión de E/S**
 - 3.3.1. E/S Programada**
 - 3.3.2. E/S por Interrupciones**
 - 3.3.3. Acceso Directo a Memoria (DMA)**
- 4. Subsistema de E/S**
 - 4.1. Cachés**
 - 4.2. Buffering**
 - 4.3. Spooling**
- 5. Gestión de la entrada/salida en sistemas operativos actuales**
 - 5.1. GNU/Linux**
 - 5.2. Windows**
- 6. Conclusión**
 - 6.1. Relación del tema con el currículo y transposición didáctica**
- 7. Bibliografía**

1. Introducción

Una de las principales funciones de un sistema operativo es la de actuar como máquina virtual o máquina extendida. En el tema actual veremos como consigue el sistema operativo que los procesos no tengan que preocuparse por las características particulares de los periféricos, proporcionando una interfaz única para cada tipo de dispositivo y aislando al programador de peculiaridades particulares.

Esta tarea es bastante compleja y supone para el sistema operativo gran parte de su código, ya que los periféricos presentan diversas discrepancias en cuanto a:

- **Velocidad de transferencia**
- **Unidad de transferencia**
- **Complejidad del controlador del dispositivo**
- Tratamiento de las **condiciones de error**.

El sistema operativo deberá solventar todas estas diferencias proporcionando una interfaz común.

2. Dispositivos de entrada/salida

Se denominan dispositivos de E/S o periféricos de comunicación a los dispositivos a través de los cuales la CPU (unidad central de proceso) se comunica con el mundo exterior, esta comunicación se realiza a través de los **controladores hardware**.

- Los periféricos se interconectan a la unidad de E/S mediante el uso de una **interfaz** con especificaciones de acceso **estandarizadas** tanto a nivel de forma, como de voltaje y de protocolos de comunicación.
- Cada periférico dispone de un **circuito controlador** (o **controladora hardware**) que gestiona toda la lógica del mismo.
- Las peculiaridades de cada dispositivo son gestionadas de forma transparente por el sistema operativo mediante un **software controlador (driver)** que es normalmente provisto por el propio fabricante.

Los periféricos pueden ser clasificados en dispositivos de **Entrada**, de **Salida**, o de Entrada y Salida; estos últimos son de tipo **almacenamiento** o **comunicación**.

A su vez, los periféricos se pueden clasificar en dispositivos de bloques y dispositivos de caracteres dependiendo de cómo manejen la información. Así mismo, el sistema operativo los va a clasificar según la exclusividad de su uso a la hora de asignar recursos a los procesos.

2.1. Dispositivos de bloques

Se caracterizan por que la información se transfiere en bloques de tamaño fijo. Cada bloque tiene su propia dirección, y se puede leer o escribir en ellos de forma independiente de los demás en cualquier momento. Un ejemplo de dispositivo de bloque son los discos.

2.2. Dispositivos de caracteres

Se caracterizan por que la información se transfiere como un flujo de caracteres, sin estructura de bloques. No existen direcciones ni tampoco se pueden realizar operaciones de búsqueda. Un ejemplo de dispositivo de carácter es la impresora.

2.3. Dispositivos de uso exclusivo

No pueden ser compartidos por varios procesos. El sistema operativo debe encargarse de asignarlo y de liberarlo cada vez que sea solicitado, cuidando de que no se produzcan interbloqueos.

2.4. Dispositivos compartidos

Pueden ser compartidos concurrentemente por varios procesos. El sistema operativo deberá velar por su buen uso y evitar los problemas que esto pueda plantear.

2.5. Dispositivos virtuales

Esta técnica muy utilizada en multiprogramación permite tratar dispositivos exclusivos como si fuesen compartidos mediante una técnica denominada spool.

3. Sistemas operativos: Gestión de E/S

3.1. Gestión software

Con objeto de ocultar las características inherentes a los dispositivos de entrada /salida que analizamos anteriormente, la comunicación con los controladores físicos (hardware) por parte de los programas (software) se realiza a través del SO, utilizando un servicio o módulo (Software de E/S), el cual ofrece una interfaz uniforme e independiente del dispositivo.

Según el autor **Andrew S. Tanenbaum**, en su libro Sistemas operativos modernos, el funcionamiento del software de E/S normalmente se adapta al siguiente modelo por capas:

3.1.1. Manejador de interrupciones

Esta capa se encarga del tratamiento de las interrupciones. Así, cuando se recibe una interrupción, el **núcleo** del SO toma el control, y a continuación notifica al **controlador software** correspondiente sobre la misma, **desbloqueándolo** mediante una primitiva del SO.

3.1.2. Controladores de dispositivos

También conocidos como manejadores o drivers, son un **código específico** que permite controlar un dispositivo de E/S concreto mediante su controlador hardware. Estos normalmente son **implementados** para un SO en particular. Así, su funcionamiento consiste en aceptar **solicitudes abstractas** por parte del software independiente del dispositivo, y ejecutar éstas, utilizando para ello uno o más **registros de dispositivos**.

Este nivel tiene las siguientes **funciones**:

- **Planificación y control**
- Generación de las **instrucciones** concretas para interactuar con el hardware

3.1.3. Software de E/S independiente del dispositivo

El objetivo principal de esta capa es proporcionar una **interfaz uniforme** a nivel de usuario, por ejemplo, en **Unix** todos los dispositivos de E/S se presentan como ficheros especiales. Adicionalmente, en esta capa tiene las siguientes responsabilidades normalmente:

- **Asignación de espacio** de direcciones en los dispositivos por bloque
- Proporcionar un **tamaño de bloque** independiente del dispositivo
- **Nombrar** los dispositivos para su identificación.
- **Asignación y liberación** de los dispositivos de uso exclusivo
- **Proteger** los dispositivos.
- Informar de los **errores**
- Gestionar el uso del **buffer**.

3.1.4. Software de E/S de la capa de usuario

Aunque la mayoría del software de E/S se encuentra dentro del SO, una pequeña parte de él consta de **bibliotecas ligadas entre sí** con los programas del usuario. Las **llamadas al sistema**, incluyendo las llamadas al sistema de E/S, se realizan comúnmente mediante procedimientos de biblioteca.

3.2. Gestión del Hardware: controladora

Un controlador o controladora de dispositivo es el componente hardware del ordenador responsable del control de uno o más dispositivos externos y del intercambio de datos entre dichos periféricos, la memoria principal y los registros de la CPU. Puede venir implementado en el propio dispositivo, integrado en la placa base o en una tarjeta controladora que se inserta en un zócalo de expansión.

Los controladores poseen una interfaz interna para su conexión con la CPU y la memoria principal, y una interfaz externa para su conexión con el dispositivo externo:

Interfaz interna. Se realiza a través de un cable y un conector que normalmente cumple alguna norma estándar, como los interfaces de disco SATA y M.2. Se encarga de convertir el flujo de bits en serie transmitidos desde el dispositivo en un bloque de bytes. Además, contiene la lógica específica para cada periférico que controla.

Interfaz externa. Se realiza utilizando las líneas del bus del sistema:

- **Bus de datos:** El bus de datos permite el intercambio de datos entre la CPU y el resto de unidades.
- **Bus de direcciones:** El bus de direcciones es un canal del microprocesador totalmente independiente del bus de datos donde se establece la dirección de memoria del dato en tránsito.
- **Bus de control:** El bus de control gobierna el uso y acceso a las líneas de datos y de direcciones. Como estas líneas están compartidas por todos los componentes, tiene que proveerse de determinados mecanismos que controlen su utilización. Las señales de control transmiten tanto órdenes como información de sincronización, evitando que haya colisión de información en el sistema.
- **Buses multiplexados:** Algunos diseños utilizan líneas eléctricas multiplexadas para el bus de direcciones y el bus de datos. Esto significa que un mismo conjunto de líneas eléctricas se comportan unas veces como bus de direcciones y otras veces como bus de datos, pero nunca al mismo tiempo. Una línea de control permite discernir cuál de las dos funciones está activa.

Registros de la controladora

La controladora propiamente dicha consta de un conjunto de registros dedicados denominados puertos de E/S, que se dividen en las siguientes categorías funcionales:

- **Registros de datos.** Conectados a las líneas de datos del bus del sistema, almacenan los datos hasta que la CPU o el dispositivo destino esté preparado para recibirlos.
- **Registros de control.** Conectados a las líneas de control del bus del sistema, se encargan de transmitir órdenes entre la CPU y los dispositivos de E/S.
- **Registros de estado.** Conectados a las líneas de datos del bus del sistema, se utilizan para indicar a la CPU el estado del dispositivo de E/S.

Interconexión CPU/Memoria

La conexión de la controladora con la CPU y la memoria se puede realizar de dos maneras:

Por mapeo de memoria del sistema. Existe un único bus entre la CPU, la memoria y el sistema de E/S con líneas de control comunes. Las controladoras forman parte del espacio normal de direcciones de la memoria y cada dispositivo tiene asignado un rango de direcciones. Son accesibles mediante instrucciones de acceso a memoria.

A través de puertos de entrada/salida. La CPU ve el subsistema de E/S como posiciones de memoria en un mapa de direcciones distinto al de la memoria principal. Una posición del mapa de E/S se llama puerto, que físicamente es una dirección de memoria dentro del controlador. Los puertos son accesibles mediante instrucciones de acceso a E/S, que

activan la señal 1/0 en el bus de control para indicar que la dirección especificada es de E/S.

3.2.1. Canales de E/S. Conexión. Tipos de canales.

Un canal de E/S es un **microprocesador** especializado en ejecutar instrucciones de E/S almacenadas en la memoria principal, liberando a la CPU de esta tarea. El canal de E/S controla la transferencia de datos e informa a la CPU al terminar.

Cuando la CPU debe realizar una operación de E/S, envía una instrucción especial que indica la dirección donde se encuentra el programa a ejecutar, en el que se especifica lo siguiente:

- Los periféricos que intervienen en la operación de E/S.
- La zona de memoria y las prioridades utilizadas en la operación.
- Las acciones a efectuar si se producen ciertas condiciones de error durante la operación.

El sistema de E/S se comunica con la CPU y con la memoria principal mediante dos buses independientes. La memoria se comunica con la CPU y con el canal por el bus del sistema, mientras que los dispositivos de E/S se comunican con el canal a través de un bus de E/S.

Hay dos tipos comunes de canales de E/S:

- **Canal selector.** Controla múltiples dispositivos de alta velocidad. En cualquier instante de tiempo está dedicado a la transferencia de datos con uno sólo de estos dispositivos. El canal sustituye a la CPU en la supervisión del funcionamiento de los controladores de E/S.
- **Canal multiplexor.** Puede controlar de forma simultánea operaciones de E/S con múltiples dispositivos. Un multiplexor orientado a la transferencia de bytes transmite caracteres hacia o desde los dispositivos de carácter, mientras que un multiplexor orientado a la transferencia de bloques alterna bloques de datos de este tipo de dispositivos.

3.3. Técnicas de gestión de E/S

3.3.1. E/S Programada

El funcionamiento se basa en la transferencia **síncrona** de datos entre la CPU y el controlador:

- Cuando la CPU ejecuta una instrucción de E/S en un proceso, envía una orden al controlador de dispositivo correspondiente y espera hasta que finalice la operación de E/S.
- El controlador lleva a cabo la operación requerida, y una vez finalizada activa los bits apropiados en su registro de estado. La CPU debe comprobar periódicamente mediante un bucle el estado del controlador, hasta que esté preparado para la siguiente operación.

- La CPU es la responsable de transferir los datos entre la memoria principal y el controlador de dispositivo a través del bus de datos en las operaciones de E/S.

Como consecuencia, la CPU debe tener control directo sobre la operación de E/S, para lo cual debe contar con instrucciones de E/S para el control de los dispositivos, la comprobación de su estado y la transferencia de los datos entre la memoria y los periféricos.

Los inconvenientes de este tipo de tratamiento son los siguientes:

- Degradación del rendimiento del sistema por la pérdida de tiempo en el bucle de espera.
- Atención exclusiva a un periférico durante el bucle de espera en detrimento de los demás.

La E/S programada es simple, pero tiene la desventaja de ocupar la CPU tiempo completo hasta que se completen todas las operaciones de E/S. Si el tiempo para "imprimir" un carácter es muy corto (debido a que todo lo que hace la impresora es copiar el nuevo carácter a un búfer interno), entonces está bien usar ocupado en espera. Además, en un sistema incrustado o embebido, donde la CPU no tiene nada más que hacer, ocupado en espera es razonable. Sin embargo, en sistemas más complejos en donde la CPU tiene otros trabajos que realizar, ocupado en espera es ineficiente. Se necesita un mejor método de E/S.

3.3.2. E/S por Interrupciones

El funcionamiento se basa en la transferencia asíncrona de datos entre la CPU y el controlador:

- Cuando la CPU ejecuta una instrucción de E/S en un proceso, envía una orden al controlador de dispositivo correspondiente y continúa ejecutando el proceso, si no precisa esperar al final de la operación. En otro caso, se suspende y se continúa con otro proceso.
- El controlador lleva a cabo la operación requerida, y una vez finalizada activa la línea de petición de interrupción del bus de control, que fuerza en la CPU una interrupción en el proceso que está ejecutando. La CPU guarda el contexto de trabajo, realiza la transferencia de datos y reanuda la ejecución del proceso en el punto donde fue interrumpido.
- La CPU es la responsable de transferir los datos entre la memoria principal y el controlador de dispositivo a través del bus de datos en las operaciones de E/S.

El tratamiento de las interrupciones se realiza de la siguiente manera:

- Después de cada instrucción la CPU verifica la línea de interrupción. Si se encuentra activa indica que se ha producido una interrupción. Las IRQ (Interrupt ReQuest) son líneas que llegan al controlador de interrupciones, un componente hardware que habilita o inhibe líneas de interrupción, y establece prioridades entre las distintas interrupciones habilitadas.
- Una CPU sin controlador de interrupciones integrado suele tener una única línea de interrupción llamada INT, que es activada por el controlador de interrupciones cuando tiene

una interrupción que servir. Al activarse esta línea, el procesador completa la ejecución de la instrucción en curso y guarda el estado del proceso en la pila.

Después la CPU consulta los registros del controlador de interrupciones para averiguar qué interrupción debe atender. A partir del número de IRQ busca en el vector de interrupciones qué rutina debe llamar para atender la petición del dispositivo asociado a dicha IRQ.

El vector de interrupciones es un vector que contiene el valor que apunta a la dirección en memoria de la rutina servidora de interrupción. Esta rutina se encarga de realizar la transferencia de datos solicitada entre la memoria principal y el controlador de dispositivo.

- Una vez finalizada la rutina servidora de interrupción, el procesador restaura el estado del proceso interrumpido y vuelve al punto anterior a la interrupción.

3.3.3. Acceso Directo a Memoria (DMA)

La E/S por programa y la E/S por interrupciones requieren la participación activa de la CPU para transferir datos entre la memoria y el controlador. El acceso directo a memoria (DMA) es una técnica que permite la transferencia directa de información sin intervención de la CPU.

Requiere un módulo hardware conectado al bus del sistema, el controlador de DMA, que consta de una serie de registros que le permiten transferir datos desde o hacia memoria:

- Registro de datos. Almacena las palabras de datos a enviar o recibir.
- Registro de dirección. Se utiliza para almacenar la dirección de la siguiente palabra que se va a enviar o recibir, y se incrementa de forma automática después de cada transferencia.
- Registro contador de palabras. Almacena el número de palabras que quedan por enviar o recibir, y se decrementa automáticamente después de cada transferencia.

El funcionamiento del DMA es el siguiente:

- Cuando la CPU desea leer o escribir un bloque de datos emite una orden al controlador de DMA, indicándole si la operación es de lectura o escritura, la dirección del dispositivo, la posición inicial de memoria de donde hay que leer o donde hay que escribir, y el número de palabras de datos que se tienen que leer o escribir.
- A partir de este momento la CPU continúa realizando otra tarea. Cuando el controlador de DMA está preparado para transmitir o recibir datos, activa la línea de petición de DMA a la CPU. Ésta renuncia al control de los buses de datos y direcciones y activa la línea de reconocimiento de DMA parando su actividad, aunque no se trata de una interrupción.
- El controlador de DMA transfiere directamente palabra a palabra el bloque completo de datos entre el dispositivo y la memoria. Cuando la transferencia finaliza, el controlador de DMA envía una señal de interrupción a la CPU. Por tanto, la CPU únicamente participa al comienzo y al final de la transferencia.

Existen diferentes tipos de acceso al bus por parte del controlador de DMA:

- **Por ráfagas.** Cuando toma el control del bus, no lo libera hasta finalizar la transmisión. Se consigue la mayor velocidad de transferencia, pero se tiene inactiva la CPU.
- **Por robo de ciclos.** Toma el bus durante un ciclo enviando una palabra cada vez. Reduce al máximo la velocidad de transferencia y la interferencia del DMA sobre la CPU.
- **DMA transparente.** Se aprovechan los ciclos en los que la CPU no utiliza el bus. No existe interferencia entre el DMA y la CPU, pero no se obtiene una velocidad muy elevada.
- **Por demanda.** Comienza la transferencia, devolviendo el control a la CPU cuando no tiene más datos disponibles. Cuando dispone de nuevos datos, retoma el control del bus.
- **Dato a dato.** Cada vez que el dispositivo solicita una transferencia, se envía un único dato y se devuelve el control a la CPU. El proceso acaba cuando se ha transferido todo el bloque.

El mecanismo de DMA se puede configurar de varias maneras. En la Figura se muestran algunas alternativas. En el primer ejemplo, todos los módulos comparten el mismo bus de sistema. El módulo de DMA, actuando como un procesador subordinado, usa E/S programada para intercambiar datos entre la memoria y un módulo de E/S a través del módulo de DMA. Esta configuración, aunque pueda ser económica, es claramente ineficiente: al igual que ocurre con la E/S programada controlada por el procesador, cada transferencia de una palabra consume dos ciclos de bus (petición de transferencia seguida por la transferencia).

El número de ciclos de bus requeridos puede recortarse sustancialmente integrando el DMA y las funciones de E/S. Como indica la Figura b, esto significa que hay un camino entre el módulo de DMA y uno o más módulos de E/S que no incluye el bus del sistema. La lógica del DMA puede ser realmente parte de un módulo de E/S, o puede ser un módulo separado que controla uno o más módulos de E/S. Este concepto se puede llevar un paso más allá conectando los módulos de E/S al módulo DMA utilizando un bus de E/S

Esto reduce a sólo uno el número de interfaces de E/S en el módulo de DMA y proporciona una configuración fácilmente expansible. En todos estos casos (Figuras b y c), el módulo de DMA utiliza el bus del sistema, que comparte con el procesador y la memoria principal, sólo para intercambiar datos con la memoria y señales de control con el procesador. El intercambio de datos entre los módulos de DMA y de E/S tiene lugar fuera del bus del sistema

4. Subsistema de E/S

Con objeto de mejorar la eficiencia en las operaciones de lectura y escritura, los SSOO suelen aplicar diferentes técnicas:

4.1. Cachés

La caché o buffer de memoria es una colección de **bloques** que pertenecen desde el **punto de vista lógico al disco**, pero que se mantiene en la memoria por razones de rendimiento.

4.2. Buffering

Se denomina buffers de E/S al espacio de memoria principal que se reserva para el **almacenamiento intermedio** de datos procedentes o con destino a los periféricos.

Con esta técnica se **emula la lectura y escritura** en los dispositivos, consiguiéndose compensar las distintas **velocidades** de los dispositivos e incrementar el **rendimiento** del sistema.

El autor William Stallings, en su libro Sistemas operativos, describe los siguientes tipos:

- **Buffer simple:** En este caso, cuando un dispositivo, o un proceso van a realizar una operación de E/S, el SO le asigna un buffer.
- **Buffer doble:** Se trata de una mejora del sistema anterior, en este caso se asignan dos búferes para las operaciones de E/S.
- **Buffer circular:** En este caso, se asigna a cada proceso un conjunto de búferes (buffer circular).

4.3. Spooling

Esta técnica es utilizada sobre todo en sistemas **multiprogramación** para la gestión de dispositivos de **uso exclusivo**. Con ella se consigue que dos o más procesos **crean** que están **utilizando el dispositivo a la vez**. Para llevarla a cabo, el sistema operativo crea un directorio especial denominado **directorio de spooling** y un proceso denominado **demonio**, de tal forma que cuando un proceso necesite, por ejemplo, imprimir, generará un archivo de impresión y lo llevará al directorio de spooling, allí permanecerá hasta que el demonio de impresión los transfiera a la impresora.

5. Gestión de la entrada/salida en sistemas operativos actuales

5.1. GNU/Linux

En GNU/Linux casi todos los dispositivos de E/S se representan como **ficheros especiales**. Una ventaja adicional de esto es que las reglas usuales de **protección** de archivos se aplican de manera automática a los dispositivos de E/S. Estos ficheros especiales, a su vez, se van a dividir en **ficheros especiales de bloque**, y ficheros **especiales de carácter**.

Cada dispositivo de E/S tiene asociado un archivo especial en el directorio / dev gestionado por el sistema de archivos, que se lee y se escribe igual que el resto de archivos. Para

acceder a un dispositivo, basta con realizar peticiones de lectura o escritura a su archivo especial asociado.

Los archivos especiales se dividen en:

- **Archivos de bloque.** Están formados por una secuencia de bloques numerados, y se caracterizan por que es posible acceder a cada bloque por su dirección.
- **Archivos de carácter.** Se utilizan para dispositivos que producen o reciben flujos de caracteres, como teclados, impresoras, ratones, tarjetas de red, etc.

Cada archivo especial tiene asociado un manejador de dispositivo, el cual tiene un número principal que sirve para identificarlo. Si un manejador de dispositivo puede manipular varios dispositivos, cada dispositivo tiene un número secundario que le identifica. Por tanto, cada pareja de números principal y secundario identifica a un determinado dispositivo.

La E/S se implementa mediante una colección de manejadores de dispositivo, uno por cada tipo de dispositivo. Su función consiste en aislar al resto del sistema de las peculiaridades del hardware. Cuando un proceso de usuario accede a un archivo especial, el sistema determina los números principal y secundario, que son utilizados como parámetros para acceder a las rutinas de manejo del dispositivo correspondiente.

Los manejadores de dispositivo pueden estar incluidos estáticamente dentro del núcleo o incluidos en módulos que pueden ser cargados dentro del núcleo en tiempo de ejecución.

La entrada/salida puede realizarse de dos maneras:

- Utilizando DMA. Es el método más rápido, pero no permite que los procesos suspendidos por la operación de E/S puedan ser intercambiados. Además, el dispositivo se paraliza junto al proceso mientras dure la transferencia, quedando inasequible para otros procesos.
- Utilizando los buffers del sistema. Se pueden usar dos clases de buffers:

Caché de disco. Adecuada para dispositivos de bloque. Se trata de un conjunto fijo de bloques recientemente leídos que se guardan en memoria para aumentar el rendimiento. Si un bloque no está en la caché de disco, se lee del disco hacia la caché y de ahí a la memoria principal, y viceversa en el caso de escritura. Para cargar un bloque en una caché totalmente ocupada se plantea una situación muy parecida a la paginación, y se resuelve con algoritmos similares. La transferencia de datos entre la caché de disco y el espacio de usuario del proceso se produce mediante DMA. Ante la posibilidad de un fallo del sistema, es necesario que los bloques críticos que hayan sido modificados permanezcan el menor tiempo posible en la caché y se transfieran al disco, ya sea a intervalos regulares o justo después de modificarse la caché.

Cola de caracteres. Adecuada para dispositivos de carácter. El dispositivo de E/S escribe en una cola de caracteres de la que lee el proceso, o también, el proceso escribe y el dispositivo lee de ella. De esta manera, las colas de caracteres sólo pueden ser leídas una vez. A medida que se lee cada carácter, éste es destruido. Este mecanismo es distinto al de la caché de buffers, donde se puede leer varias veces.

5.2. Windows

En el caso de Windows, la gestión de los dispositivos se realiza a través de la **manipulación de objetos**, así para cada dispositivo se carga un objeto driver, y al menos, un objeto dispositivo, los cuales pueden crearse y cargarse cuando se conecta el dispositivo, lo que permite que se puedan conectar **dispositivos en caliente**.

El administrador de E/S de Windows está diseñado para proporcionar **un marco de trabajo extensivo y flexible**, que permita manejar una amplia variedad de dispositivos, y la instalación de controladores (plug-and-play). Para facilitar la tarea de codificación de drivers, Microsoft ha definido el **modelo WDM**, Windows Driver Model.

El gestor plug and play permite que los dispositivos compatibles sean inmediatamente reconocidos una vez conectados. Para ello, el gestor pide al dispositivo que se identifique, y con esta información carga en memoria el manejador de dispositivo apropiado.

La entrada/salida de dispositivos se maneja a través de las APIs del subsistema Win32, el cual se relaciona con el hardware a través de la capa de abstracción de hardware (HAL). Esta capa es una interfaz entre el hardware y el resto del sistema operativo que se caracteriza por lo siguiente:

- Está implementada como una biblioteca de enlace dinámico (dll) y es responsable de proteger y aislar el resto del sistema de las especificaciones del hardware.
- Presenta los dispositivos al sistema operativo de manera homogénea a través de un conjunto de funciones bien definidas. Estas funciones son llamadas tanto desde el sistema operativo como desde los propios manejadores de dispositivo, permitiendo a éstos adaptarse a distintas arquitecturas de E/S sin tener que ser modificados en gran medida.
- Si Windows es portado a una nueva arquitectura de procesador el HAL debe ser reescrito, pero el resto del sistema simplemente debe ser recompilado.
- También suministra la interfaz para el multiprocesamiento simétrico, transformando cada procesador físico en un procesador virtual idéntico al resto de procesadores.
- Al HAL sólo pueden acceder componentes del Executive de Windows y nunca los programas en modo usuario. El HAL también intenta ser el único software dentro del sistema que se comunica con el hardware, aunque existe un pequeño número de llamadas de los manejadores de dispositivo y del núcleo que interactúan directamente con el hardware.

6. Conclusión

A modo de síntesis, se podría indicar que una de las funciones más importantes de un sistema operativo es gestionar las operaciones de E/S, permitiendo de esta forma que los **programadores** de aplicaciones de usuario focalicen sus fuerzas en otros aspectos de programación más abstractos.

6.1. Relación del tema con el currículo

Este tema es aplicado en el aula en los módulos profesionales siguientes, con las atribuciones docentes indicadas (PES/SAI):

Grado Medio

- Sistemas operativos monopuesto (SMR) (PES/SAI)

Grado Superior

- Sistemas informáticos (DAM / DAW) (PES/SAI)
- Implantación de sistemas operativos (ASIR) (PES/SAI)

7. Bibliografía

- De Anasagasti, Miguel. "Fundamentos de la Computadora" 9^aed 2004 Edt. Paraninfo
- Patterson D.A. y Hennessy JL. "Estructura y diseño de computadoras: la interfaz hardware/software" 4^a Ed. (2005) Edt McGraw-Hill
- Prieto A, Lloris A, Torres JC. "Introducción a la Informática" 4^aed. (2006) Edt. McGraw-Hill
- Stallings W. "Organización y Arquitectura de Computadoras" (2006) 5^a Ed. Edt. Prentice-Hall
- Ramos A, Ramos MJ y Viñas S "Montaje y Mantenimiento de Equipos" (2012). Edt McGraw-Hill
- Jiménez Cembreras, Isabel M^a "Sistemas Informáticos" 2^aEd (2018) Edt. Garceta
- Informáticos" 2^aEd (2018) Edit. Garceta