



**Universidad
Europea de Madrid**

LAUREATE INTERNATIONAL UNIVERSITIES

FUNDAMENTOS DE COMPUTADORES

ENTRADA/SALIDA

© Todos los derechos de propiedad intelectual de esta obra pertenecen en exclusiva a la Universidad Europea de Madrid, S.L.U. Queda terminantemente prohibida la reproducción, puesta a disposición del público y en general cualquier otra forma de explotación de toda o parte de la misma.

La utilización no autorizada de esta obra, así como los perjuicios ocasionados en los derechos de propiedad intelectual e industrial de la Universidad Europea de Madrid, S.L.U., darán lugar al ejercicio de las acciones que legalmente le correspondan y, en su caso, a las responsabilidades que de dicho ejercicio se deriven.

Índice

Presentación	4
Introducción	5
El controlador I	7
El controlador II	8
Técnicas aplicadas en operaciones de E/S	10
Mapa de memoria	12
E/S controlada por programa	14
E/S controlada por interrupciones	15
Funcionamiento de la E/S por interrupción	15
DMA: acceso directo a memoria I	16
DMA: acceso directo a memoria II	17
El controlador de DMA	20
Procesadores de E/S I	21
Procesadores de E/S II	22
Resumen	24

Presentación

En este tema veremos el último, aunque no por ello el menos importante, componente en que Von Neumann dividió su computador. Nada más y nada menos que la permitirá la comunicación *hombre-máquina* o *máquina-máquina*. Este componente es el controlador de entrada/salida.

Para ello, estudiaremos:

- La definición de controlador, para qué sirve y cuáles son sus funciones.
- Técnicas aplicadas en operaciones de E/S. Cuáles son las técnicas con las que un programa puede comunicarse con un periférico. Recordemos que aunque el proceso de comunicación es *procesador-periférico*, en realidad es *programa-periférico* pues el procesador simplemente ejecuta programas, siendo el programa el que tiene las necesidades de comunicación, no el procesador.
- Mapa de memoria común frente a independiente. Se refiere a cómo se ven los periféricos, cuál es la dirección para comunicarse con ellos.
- Ejemplo de controlador 1: E/S controlada por programa.
- Ejemplo de controlador 2: E/S controlada por interrupciones.
- Ejemplo de controlador 3: Acceso directo a memoria (DMA).
- Ejemplo de controlador 4: Procesadores de E/S .



Introducción

A grandes rasgos, entrada/salida es todo componente que Von Neumann no ha descrito en su estructura básica de un computador. Es decir, cualquier componente que no sea memoria, unidad de control o buses, es de entrada/salida (a partir de ahora E/S). Por tanto, la E/S es un componente que permite intercambiar información entre la computadora y el mundo exterior.

Como curiosidad, cabe destacar que los periféricos aumentan muy poco su velocidad con respecto al aumento anual que sufren los microprocesadores y los computadores en general. Esta diferencia de velocidades, en la mayor cantidad de los casos, se debe a que los dispositivos de E/S suelen tener componentes móviles, o elementos físicos, que no permiten aumentar el movimiento de datos tan rápido como sería deseable.



Debido a esta diferencia de velocidades y tecnologías, con los dispositivos de E/S se pretende que los tiempos de espera por parte del microprocesador sean los menores posibles, lo que se consigue solapando, lo más posible, los tiempos de trabajo de la E/S y la unidad de control.

La mayor parte de los dispositivos de E/S, son vistos por el microprocesador como una memoria, una memoria especial que sirve como intercambio de información entre el periférico y el procesador. Como los periféricos manejan información muy lentamente, es frecuente que la unidad de E/S disponga de una memoria auxiliar donde guardar la información del periférico.

Por ejemplo, si comparamos el tiempo que tarda una impresora en escribir un carácter con el tiempo que el procesador determina el carácter a enviar, comprobaremos que la impresora tarda mucho en escribirlo. Por tanto, parece coherente que la impresora tenga una memoria intermedia en la que pueda almacenar una palabra, línea o incluso página, para que el procesador no este enviando carácter a carácter.

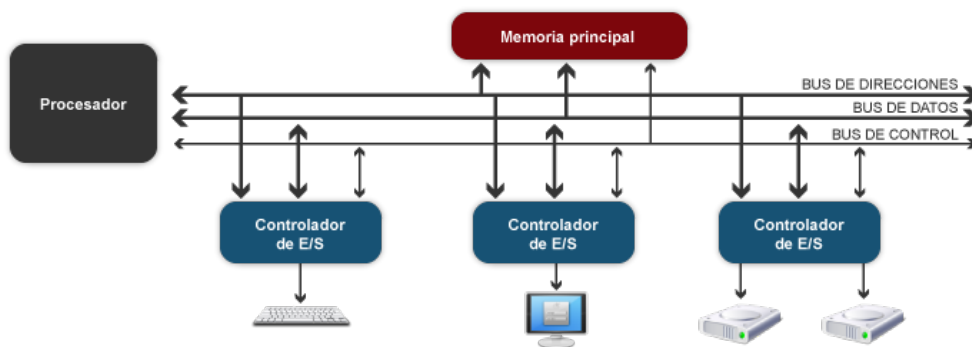
El controlador I

Debido a la gran cantidad y variedad de periféricos existentes hoy en día, cada uno con un funcionamiento diferente, surge la necesidad de colocar un dispositivo hardware entre el computador y los periféricos. Este componente se conoce como controlador y su función principal es descargar al procesador de la tarea de controlar el trasvase de información entre el periférico y la memoria.

Al igual que con cualquier otro componente del computador, la comunicación entre microprocesador y periféricos se realiza a través de los buses.

Para que un programa pueda comunicarse con ellos, los controladores deben realizar funciones de control y sincronización de la transferencia de datos desde el procesador y la memoria principal a los periféricos, de establecimiento del camino de comunicación entre los recursos internos y los periféricos, de detección y gestión de errores y de almacenamiento temporal de datos.

Tal como puedes comprobar en la siguiente imagen, un computador no tiene por qué tener un único controlador genérico, sino que puede tener varios controladores para periféricos de diferentes familias.



El controlador II

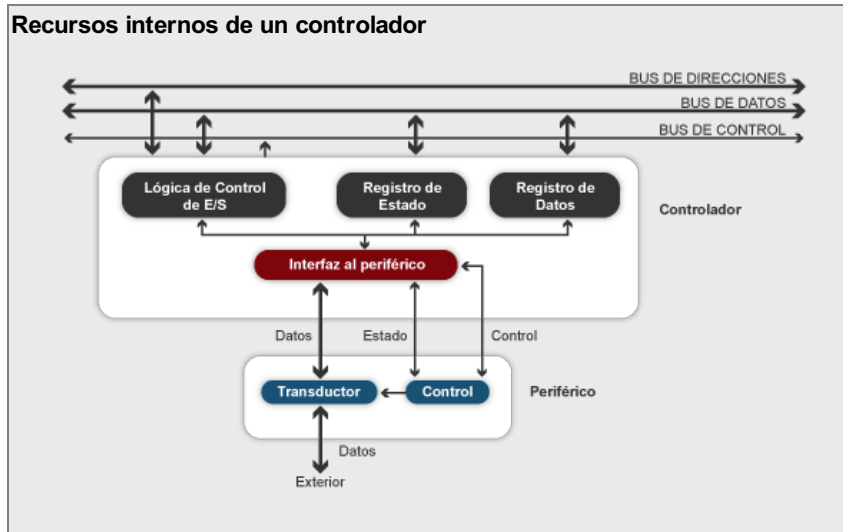
Para realizar estas funciones, el controlador dispone internamente de una serie de recursos entre los que podemos destacar los siguientes.

Lógica de control	Se encarga de activar y controlar las funciones que lleva a cabo ese controlador.
Registro de estado	Almacena información sobre cómo se ha realizado el trasvase de información y sobre los posibles fallos en caso de haberse producido alguno.
Registro de datos	Almacena la información que se maneja en cada transferencia elemental.
Interfaz al periférico	Soporta el intercambio de datos con el transductor del periférico, así como la generación y detección de señales de control y estado.

Fijémonos en que esta estructura es muy parecida al de la interfaz que permitía unir dos buses diferentes, ya que esa función es parte del cometido de un controlador de E/S: ajustar velocidades de reloj, sincronizar señales, ajustar tensiones de funcionamiento, detectar errores, priorizar el uso, etc.

El controlador de un módulo de E/S permite que el procesador vea a una amplia gama de dispositivos de forma simplificada. Ante el gran espectro de posibilidades que se pueden dar según los diferentes dispositivos, el módulo debe ocultar los detalles de temporización, formatos y electrónica de los dispositivos externos, para que el procesador los maneje todos, de manera similar, independientemente del tipo y protocolos que usen internamente.

En función de la bibliografía utilizada, podemos encontrar que a los controladores de E/S se los denomina como canales de E/S o procesadores de E/S. En cualquier caso, no importa como se denomine al dispositivo encargado de hacer de puente entre los buses del sistema y los periféricos, pero por norma general, controlador es el más extendido, dejando el término de canal a los microcomputadores y procesador a los canales que usan los supercomputadores para las comunicaciones internas.



Técnicas aplicadas en operaciones de E/S

Recordemos que la E/S es lenta. Por ese motivo, una técnica muy extendida es la de solapar las operaciones de entrada/salida con otras actividades, de manera que la unidad de control se puede liberar de ciertas tareas de poca importancia.

En función del grado de implicación que tenga la unidad de control con las operaciones de E/S, según delegue más trabajo en el controlador o menos, existen cuatro técnicas fundamentales para realizar las operaciones de E/S.

<p>E/S controladas por programa</p>	<p>La unidad de control controla íntegramente la operación de E/S, impidiendo cualquier tipo de solapamiento en el trabajo de ambos componentes.</p> <p>Ejemplo: un módem conectado a un puerto serie.</p>
<p>E/S controladas por interrupción</p>	<p>Parte del proceso de E/S corre a cargo del controlador pudiendo trabajar en paralelo al procesador durante dicho periodo. Este método permite que la UCP, una vez que ha indicado la operación al controlador, continúe con su actividad normal y solo dedique tiempos de atención cada vez que el periférico lo requiera.</p> <p>Ejemplo: el teclado o el ratón.</p>

<p>Acceso directo a memoria (DMA)</p>	<p>La mayor parte de la operación de E/S la soporta el controlador, existiendo una simultaneidad alta. Los controladores tienen cierto grado de inteligencia para controlar íntegramente la transferencia de información entre periféricos y memoria principal.</p> <p>Ejemplo: un disco duro externo.</p>
<p>Procesador de E/S especializado</p>	<p>Todo el proceso de E/S corre a cargo del controlador, que alcanza tal grado de complejidad que actúa como otro procesador dedicado. El paralelismo entre el procesador y la operación de E/S es total.</p> <p>Ejemplo: una tarjeta de red o una tarjeta gráfica.</p>



Mapa de memoria

Los controladores de E/S consiguen que todos los dispositivos tengan una interfaz de alto nivel común, por lo que el programador ve esos dispositivos como direcciones de memoria, ficheros, etc., que permiten la comunicación con ellos a través de operaciones de entrada/salida.

Por tanto, los dispositivos tienen asignado un grupo de direcciones de memoria a las que accedemos para poder comunicar datos a esos dispositivos. Según el formato de estas direcciones de memoria, vamos a tener dos procedimientos:


- Entrada/salida con mapa de memoria común.
- Entrada/salida con mapa de memoria independiente.

E/S con mapa de memoria común

Las direcciones de memoria y E/S tienen el mismo formato. Lo único que se hace es reservar un grupo de direcciones de memoria para entrada/salida. Así, si en un computador de 32 bits en el bus de direcciones, que podría llegar a direccionar 4Gb de memoria, se reserva un grupo de estas direcciones para entrada/salida, se reducirá en esa cantidad el tamaño de la memoria direccionable por el computador.

Por tanto, con este tipo de direccionamiento, solo hay un mapa de memoria para direccionar los dispositivos de E/S, los datos y las instrucciones de los programas.

La ventaja de este sistema, es que el procesador utiliza las mismas instrucciones a nivel de ensamblador para comunicarse con la memoria que con la E/S.



1/3 

E/S con mapa de memoria independiente

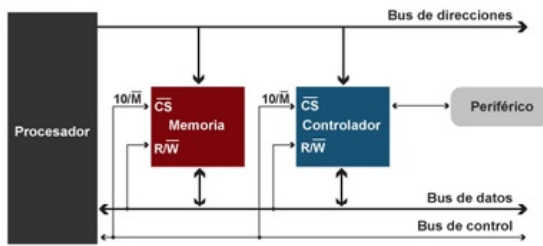
Es el caso contrario al anterior. Es decir, existe una dirección N en memoria principal y la misma dirección en E/S, de manera que el microprocesador tiene que diferenciar en el momento de solicitar un dato por el bus, si esa solicitud va dirigida a memoria o a E/S.

Para acceder a ambos mapas de memoria, el procesador utiliza el bus de direcciones y, mediante una señal de control IO/M (del bus de control), se selecciona el mapa de memoria principal o el de E/S:

- 1 para acceder al mapa de E/S
- 0 se accede al mapa de memoria principal.

 2/3 

Cuando se maneja un mapa de E/S independiente, la unidad de control dispone de una serie de instrucciones para escribir y leer las posiciones del mapa de E/S. Estas instrucciones generan señales especiales en el bus de control.



Hay procesadores, como los de la familia Intel que manejan un mapa de memoria independiente y microprocesadores como los de la familia Alpha que tienen un mapa de memoria común.

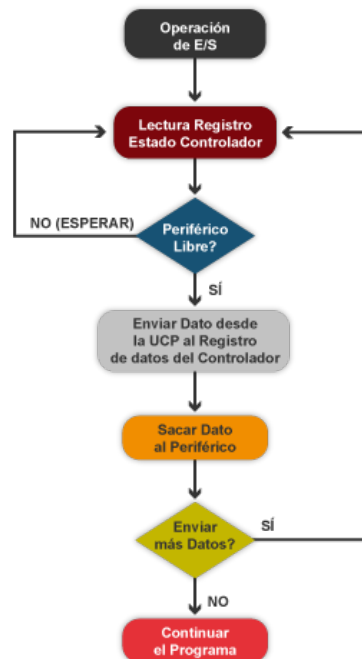
 [Funcionamiento: E/S con mapa de memoria independiente](#)

E/S controlada por programa

Es el controlador con menor inteligencia. Por tanto, la unidad de control controla íntegramente la operación de E/S, impidiendo cualquier tipo de solapamiento en el trabajo de ambos componentes.

El procesador controla todo el proceso de la operación de E/S, enviando al controlador las diferentes órdenes para su correcta ejecución.

Por tanto, es un programa (el programa de usuario) el encargado de generar las instrucciones que soportan las operaciones de E/S, teniendo que controlar todas las posibles ocurrencias y errores que se pueden producir durante el acceso al dispositivo. La unidad de control interpreta dichas instrucciones y envía al controlador las órdenes de control, lectura y escritura.



Para poder controlar un dispositivo por este método, el software (nuestra aplicación) tiene que implementar el algoritmo mostrado en la figura y que siga los siguientes pasos:

- La Unidad Central de Procesos inicia la operación.
- Explora el estado del periférico.
- Gobierna la transferencia.
- Detecta posibles errores.
- Finaliza el proceso.

La E/S controlada por programa tiene un grave inconveniente: consume demasiado tiempo de CPU.

E/S controlada por interrupciones

La S/E por interrupción, permite solucionar en parte el problema de la E/S por programa, que es el uso de Unidad de Control.

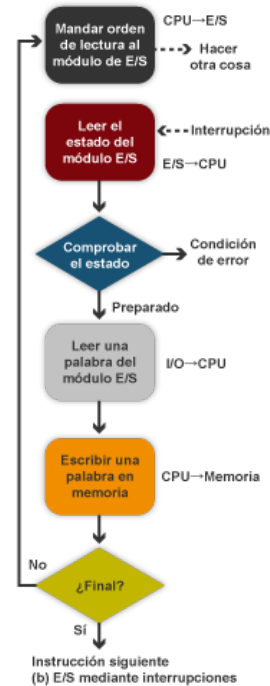
Funcionamiento de la E/S por interrupción

 [Descripción de una interrupción](#)
Documentos

Con la ayuda de las interrupciones, el proceso puede desligarse de llevar un control exhaustivo sobre el proceso de E/S.

El mecanismo para ello es:

- Iniciar el proceso de E/S.
- Dar una colección parcial de órdenes y de datos que debe realizar el dispositivo.
- Desligarse del control, con la esperanza de que el propio controlador de E/S generará una interrupción para indicar un evento.



Este evento puede ser de dos tipos:

- Los datos pendientes a procesar se han terminado, por lo que se requieren más datos.
- El dispositivo de E/S ha generado un error, por lo que necesita nuevas órdenes de control.

En ambos casos, la interrupción generada conlleva una atención por parte del procesador, que actuará en consecuencia.

Con este sistema se consigue un solapamiento de la E/S con otras actividades que el procesador puede realizar, necesitándose la atención del mismo solo durante los tiempos de interrupción. Liberándose así sustancialmente el consumo de procesador durante la tarea de E/S.

DMA: acceso directo a memoria I

Cuando el periférico genera muchas atenciones la comunicación por interrupciones es óptima, pero en cada atención se transmite una cantidad de datos reducida. Esto ocurre, por ejemplo, con el teclado o el ratón, los cuales generan una interrupción cada vez que se pulsa una tecla o se mueve el ratón.

Cuando la cantidad de datos a mover es muy grande, como por ejemplo un escáner o una impresora, la cantidad de interrupciones generadas sería muy elevada, por lo que la cantidad de veces que el microprocesador es interrumpido lo sería también. Hay que entender que atender una interrupción no es una tarea simple, sino que conlleva tiempo en el *backup* y *restore* del estado actual del microprocesador.

Para los dispositivos en los que existe una gran cantidad de movimiento de datos, la técnica más utilizada es el acceso directo a memoria (DMA), ya que, con este modo, se utiliza el propio controlador del periférico para realizar la transferencia de información entre la memoria principal y el periférico, sin que la intervención del procesador sea necesaria. Es decir, el movimiento de datos se realiza de forma transparente para el procesador, siendo directamente el periférico de E/S el que escribe-lee los datos en memoria.



DMA: acceso directo a memoria II

Este método implica muchas mejoras en la arquitectura de la memoria o los buses, ya que el acceso directo a memoria por parte del dispositivo de E/S podría entorpecer el normal funcionamiento del microprocesador si los buses o incluso la memoria estuvieran ocupados dando servicio a otros periféricos provocando que el microprocesador tuviera que esperar. Es importante evitar esta situación por todos los medios, ya que no ganaríamos nada contra otros métodos como el de *controlado por programa*.

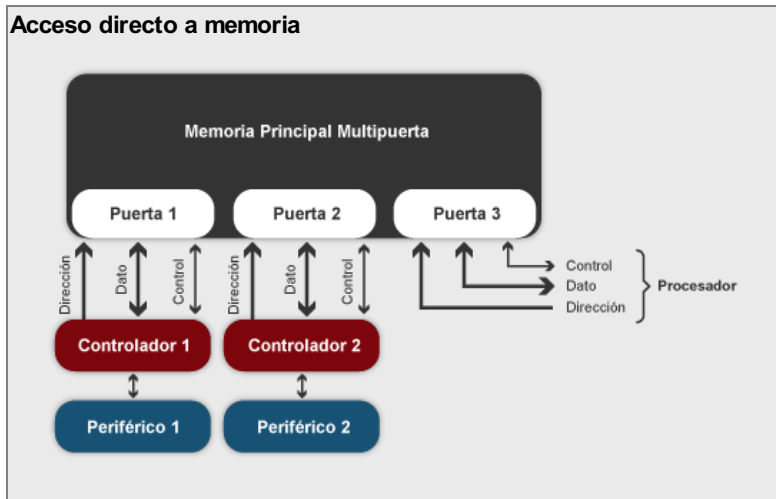
Para solucionar el problema de la memoria, una de las técnicas más utilizadas radica en el empleo de una memoria principal multipuerta que posee varias puertas de acceso, una de las cuales se dedica al microprocesador y el resto a los controladores de los periféricos.

 Acceso directo a memoria
Gráfico

Seguiría pendiente el problema de acceso a los buses. En este caso, la opción de la duplicidad sería una posible alternativa, pero complicaría en exceso toda la arquitectura. Además, hay otras opciones más baratas y simples, como las siguientes.

Por ráfagas	El controlador de DMA no libera los buses hasta que no acaba la transferencia de un bloque completo de información. La ventaja de este sistema radica en una elevada velocidad de transferencia de información. El inconveniente es que el procesador permanece inactivo durante largos periodos de tiempo.
-------------	---

<p>Por robo de ciclo</p>	<p>Cuando el controlador de DMA dispone de datos para transferir, toma el control de los buses durante un ciclo de procesador, en el que transmite una palabra.</p> <p>Luego cede los buses a la unidad de control, que continua con su actividad normal hasta que nuevamente el controlador de DMA roba otro ciclo de unidad de control y transmite otra palabra.</p> <p>Con este procedimiento, los buses se reparten a lo largo del tiempo entre el controlador de DMA y el procesador. Así la unidad de control mantiene su actividad, aunque a menor velocidad, evitando la inactividad por periodos largos de tiempo.</p>
<p>DMA transparente</p>	<p>Es similar al modelo anterior, pero el controlador de DMA roba los buses a la unidad de control durante los ciclos en los que ella no los utiliza.</p> <p>Si, durante un ciclo, la unidad de control se dedica a decodificar o interpretar el código de una instrucción, no usa los buses y el controlador de DMA aprovecha la situación para transferir una palabra.</p> <p>Con este sistema no se rebaja la velocidad de procesamiento de la unidad de control y se llevan a cabo operaciones de E/S. El principal inconveniente es la reducida velocidad que se alcanza en la transferencia de información.</p>
<p>Por demanda</p>	<p>Cuando el Controlador de DMA dispone de datos a transferir, solicita el control de los buses y los mantiene en su poder hasta que finaliza el envío de toda la información.</p>



El controlador de DMA

Los controladores de DMA se encargan de controlar los buses de direcciones, de datos y de control para poder realizar los accesos a la memoria principal.



Pero, ¿Cómo funcionan los controladores de DMA?

- El microprocesador inicia la operación de E/S indicando la tarea a realizar y asignando un bloque de memoria para que sea utilizado por el controlador de DMA, que se desentiende del resto de la operación.
- El controlador inicia la operación de E/S según el modo de acceso a los buses y a la memoria que soporte.
 - El controlador solicita al procesador la liberación de los buses.
 - Cuando esto ocurre, el controlador de DMA toma el control de los mismos.
 - Utiliza el bus de direcciones para enviar las direcciones de memoria a las que se quiere acceder.
 - Por el bus de control se indica si la operación es de lectura o de escritura.
 - Por el bus de datos se trasfiere la información.
 - Repite estos puntos tantas veces como necesite hasta que todos los datos están transmitidos.
- Terminada la operación, genera una interrupción para informar al procesador que la operación de E/S está terminada.
- En cualquier punto de la operación se puede generar un error. En tal caso, el controlador de DMA también genera una interrupción, pero para informar de errores.

Cabe destacar que los controladores de DMA también usan interrupciones, ya que al menos generan una interrupción para indicar el fin de operación e incluso, en otros casos, pueden generar más de una interrupción para avisar de problemas y otras circunstancias.

Procesadores de E/S I

Solo queda por ver un grupo de controladores de entrada/salida: los procesadores de E/S.

Estos controladores alcanzan tal grado de complejidad que se transforman en verdaderos procesadores capaces de interpretar y ejecutar un pequeño repertorio de instrucciones para manipular las operaciones de E/S.

Cuando hay que realizar un programa que atienda las E/S, el procesador principal encarga al procesador de E/S (PE/S) que lo desarrolle, trabajando ambos en paralelo. El PE/S solo se comunica con el procesador principal al terminar el programa de E/S mediante la petición de interrupción.

Los procesadores de E/S son utilizados en circunstancias muy concretas ya que, aunque son los más eficientes, también son los más complejos y caros.

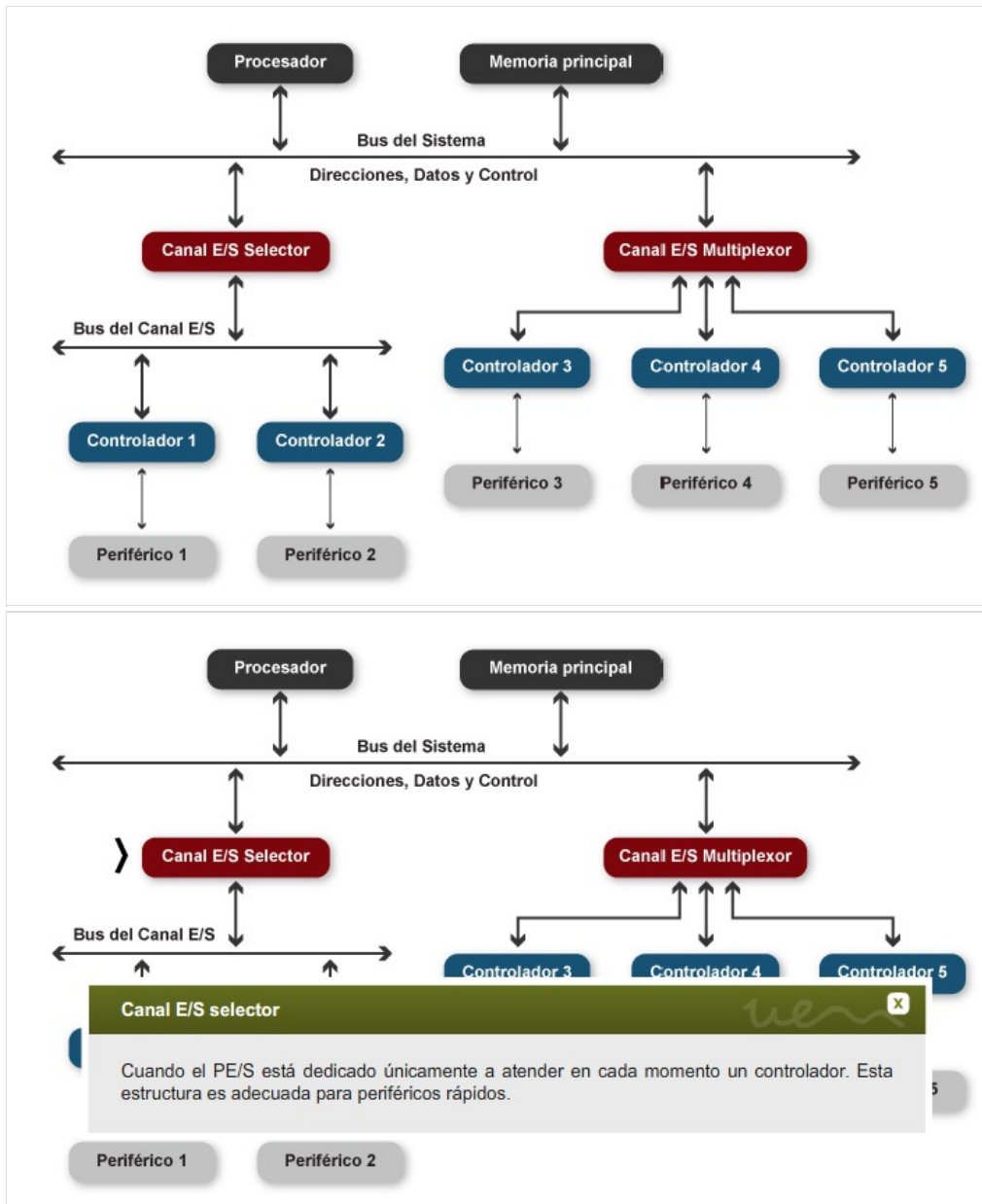
Fundamentalmente, se emplean cuando los datos a tratar por el controlador, procedentes del periférico o del propio procesador, son infinitos y, por tanto, se necesita procesamiento e inteligencia local para manejarlos sin tener que estar pidiendo ayuda constantemente al procesador principal del ordenador.

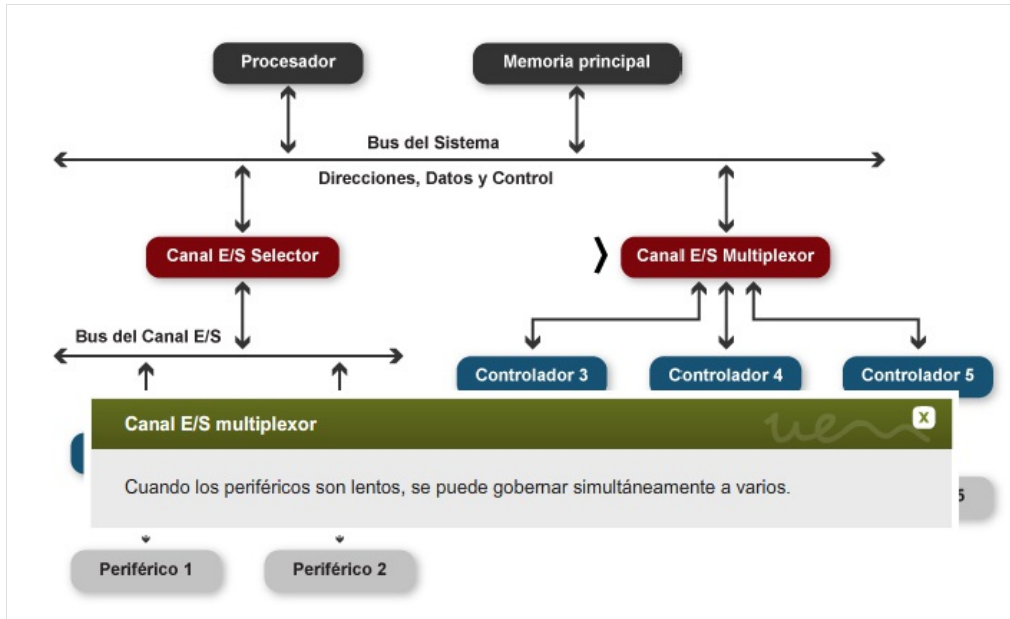
Un ejemplo muy característico es la tarjeta de red o el propio controlador de vídeo del ordenador (tarjeta gráfica). En el primer caso, por citar un ejemplo, la cantidad de datos que llega a la tarjeta es infinita. Constantemente llegan datos, pero solo una porción de ellos es para el procesador, la mayor cantidad de los datos que viajan por la red corresponden a otras comunicaciones presentes en la red, o señales de *alive* (acuse de vida). Es importante que estas señales sean procesadas directamente por la tarjeta de red sin que el procesador sea molestado. Solo se informa al procesador en caso de existir datos procedentes para nuestro ordenador.



Procesadores de E/S II

Como puedes comprobar en la figura, los PE/S admiten dos comportamientos diferentes según la manera de gobernar a los controladores de periféricos. Haz clic sobre las cajas granate en el siguiente gráfico para ampliar la información sobre dichos comportamientos.





Resumen

En este tema hemos visto cuáles son las técnicas que se utilizan para la comunicación de un computador actual con el mundo exterior. Una comunicación que puede ser:

- Hacia el humano por medio del teclado, ratón, impresoras, etc.
- Contra otras máquinas como la tarjeta de red, etc.

Se han desglosado los cuatro modos principales de entrada salida:

- E/S controlada por programa.
- E/S controlada por interrupciones.
- Acceso directo a memoria (DMA).
- Procesadores de E/S.

Y se han comparado las ventajas e inconvenientes de cada uno de los métodos.

Además, hemos analizado el hardware que necesita cada uno de ellos y las mejoras que, en la estructura clásica de un computador, necesitan para ser implementados.