

Control del sistema de E/S: Sistemas Operativos

I/O system control: Operating System

Angélica Carrión^a

Abstract:

The purpose of this work is to introduce a computer system into the Input / Output (I / O) system. The I / O system is an interface that has a computer with the outside and its main objective is to facilitate all I / O operations between peripherals and memory or in turn the processor registers. To manage I / O operations, hardware is essential and complemented with the help of software. The hardware system and the Operating System (OS) of a computer system are intended to meet the demands of users in the execution of specific tasks.

Keywords:

I/O system, Direct Memory Access, Operating System

Resumen:

El presente trabajo tiene como finalidad introducir en el sistema de Entrada/Salida (E/S) de un sistema de cómputo. El sistema de E/S es una interfaz que posee una computadora con el exterior y tiene como principal objetivo el facilitar todas las operaciones de E/S entre los periféricos y la memoria o a su vez los registros del procesador. Para gestionar las operaciones de E/S es fundamental un hardware y complementar con la ayuda de un software. El sistema hardware y el Sistema Operativo (SO) de un sistema de cómputo tienen la finalidad de satisfacer las demandas de los usuarios en la ejecución de tareas específicas.

Palabras Clave:

Sistema E/S, Acceso Directo a Memoria, Sistema Operativo

Introducción

El sistema operativo (SO) es un programa que nos permite administrar el hardware de la computadora, además nos proporciona las bases para ser usado en los programas de aplicación y cumple una gran funcionalidad al ser el intermediario entre el usuario y el hardware. Todas las tareas pueden ser tratadas de diferentes formas, permitir que algunos sistemas operativos sean diseñados de manera práctica, otros de manera eficiente y por otros que cumplen ambas funcionalidades.

Los sistemas de Entrada/Salida (E/S) son una interfaz que posee una computadora con el exterior y tiene como principal objetivo el facilitar todas las operaciones de E/S entre los periféricos y la memoria o a su vez los registros del procesador, donde para poder gestionar las operaciones de E/S es fundamental un hardware y complementar con la ayuda de un software. Debido a la variedad de los periféricos, es obligatorio tener un

hardware y software específicos para cada uno, por eso se ha llevado la normalización de conexión de los periféricos y la computadora por medio de módulos de E/S o también conocidos como controladores de E/S. Esto nos permite tener una conexión entre los módulos de E/S y los periféricos.

A continuación, se dará un estudio sobre el sistema de E/S, en lo cual se verán los principios de hardware y software de E/S, repaso de interrupciones, E/S por asignación de memoria y por último sobre los controladores de dispositivos.

Desarrollo

Principios del hardware de E/S

Distintas personas ven el hardware de E/S de diferentes maneras. Los ingenieros eléctricos lo ven en términos de chips, cables, fuentes de poder, motores y todos los demás componentes físicos que constituyen el hardware.

^a Autor de Correspondencia, Universidad Técnica Estatal de Quevedo, Facultad de Ciencias de la Ingeniería, Email: angelicangon.carrion@uteq.edu.ec

Los programadores ven la interfaz que se presenta al software: los comandos que acepta el hardware, las funciones que lleva a cabo y los errores que se pueden reportar [1].

Dispositivos de E/S

Los dispositivos de E/S se pueden dividir básicamente en dos categorías: dispositivos de bloque y dispositivos de carácter. Un dispositivo de bloque almacena información en bloques de tamaño específico, cada uno con su propia dirección. La propiedad esencial de un dispositivo de bloque es que hace posible leer o escribir en cada bloque de manera independiente de los demás. Los discos duros, CD-ROMs y memorias USBs son dispositivos de bloque comunes. El otro tipo de dispositivo de E/S es el dispositivo de carácter. Un dispositivo de carácter envía o acepta un flujo de caracteres, sin importar la estructura del bloque. No es direccionable y no tiene ninguna operación de búsqueda. Las impresoras, las interfaces de red, los ratones (para señalar), y la mayoría de los demás dispositivos que no son parecidos al disco se pueden considerar como dispositivos de carácter.

Los dispositivos de E/S cubren un amplio rango de velocidades (ver Figura 1), lo cual impone una presión considerable en el software para obtener un buen desempeño sobre muchas órdenes de magnitud [1].

Dispositivo	Velocidad de transferencia de datos
Teclado	10 bytes/seg
Ratón	100 bytes/seg
Módem de 56K	7 KB/seg
Escáner	400 KB/seg
Cámara de video digital	3.5 MB/seg
802.11g inalámbrico	6.75 MB/seg
CD-ROM de 52X	7.8 MB/seg
Fast Ethernet	12.5 MB/seg
Tarjeta Compact Flash	40 MB/seg
FireWire (IEEE 1394)	50 MB/seg
USB 2.0	60 MB/seg
Red SONET OC-12	78 MB/seg
Disco SCSI Ultra 2	80 MB/seg
Gigabit Ethernet	125 MB/seg
Unidad de disco SATA	300 MB/seg
Cinta de Ultrium	320 MB/seg
Bus PCI	528 MB/seg

Figura 1. Velocidades de transferencia de datos comunes de algunos dispositivos

Los dispositivos externos dedicados a la E/S en una computadora se pueden agrupar, a grandes rasgos, en tres categorías:

- Legibles al Usuario: Adecuados para la comunicación con el usuario de la computadora.
- Legibles para la máquina: Adecuados para la comunicación con equipamiento electrónico.

- Comunicación: Adecuados para la comunicación con dispositivos remotos [2].

E/S por asignación de memoria

Cada controlador tiene unos cuantos registros que se utilizan para comunicarse con la CPU. Al escribir en ellos, el SO puede hacer que el dispositivo envíe o acepte datos, se encienda o se apague, o realice cualquier otra acción. Al leer de estos registros, el SO puede conocer el estado del dispositivo, si está preparado o no para aceptar un nuevo comando, y continúa de esa manera.

Repaso de las interrupciones

En un sistema de computadora personal común, la estructura de las interrupciones es como se muestra en la Figura 2. A nivel de hardware, las interrupciones funcionan de la siguiente manera. Cuando un dispositivo de E/S ha terminado el trabajo que se le asignó, produce una interrupción (suponiendo que el SO haya habilitado las interrupciones). Para ello, impone una señal en una línea de bus que se le haya asignado. Esta señal es detectada por el chip controlador de interrupciones en la tarjeta principal, que después decide lo que debe hacer.

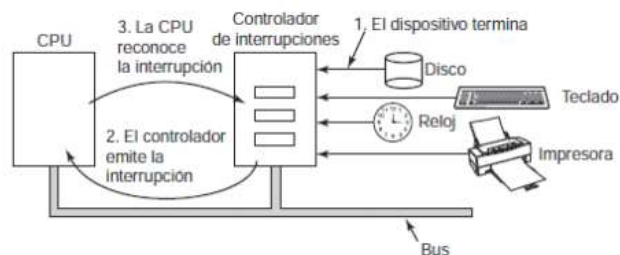


Figura 2. Ocurrencia de interrupciones en el SO.

Controladores de dispositivos

Por lo general, las unidades de E/S consisten en un componente mecánico y un componente electrónico. Normalmente es posible separar las dos porciones para proveer un diseño más modular y general. El componente electrónico se llama controlador de dispositivo o adaptador. El trabajo del controlador es convertir el flujo de bits serial en un bloque de bytes y realizar cualquier corrección de errores necesaria [1].

El controlador para un monitor también funciona como un dispositivo de bits en serie a un nivel igual de bajo. Lee los bytes que contienen los caracteres a mostrar de la memoria y genera las señales utilizadas para modular el haz del CRT para hacer que escriba en la pantalla. Cada controlador posee registros que se utilizan para comunicarse con la CPU donde pueden ser parte del espacio normal de las direcciones de la memoria de E/S mapeada a la memoria. Esto puede utilizar un espacio de

direcciones especial para la E/S, asignando a cada controlador una parte de él [3].

El SO realiza la E/S al escribir los comandos en los registros de los controladores; los parámetros de los comandos también se cargan a su vez los registros de los controladores, y al aceptar el comando, la CPU puede dejar al controlador y dedicarse a otro trabajo, y cuando ya termina el comando de ejecutarse el controlador provoca una interrupción que permita que el SO obtenga el control del CPU y verificar los resultados de la operación.

Controlador de E/S	Dirección de E/S	Vector de Interrupciones
Reloj	040 – 043	8
Teclado	060 – 063	9
Disco duro	320 – 32f	13
Impresora	378 – 37f	15
Disco flexible	3f0 – 3f7	14
Rs232 primario	3f8 – 3ff	12
Rs232 secundario	2f8 – 2ff	11

Figura 3. Vector de interrupciones para dispositivos de E/S.

Cada dispositivo de E/S conectado a una computadora necesita código específico que sirva para controlar ese dispositivo (ver Figura 3). Este código, llamado controlador de dispositivo (*device driver*), por lo general es escrito por el fabricante del dispositivo y se proporciona junto con el hardware. Puesto que cada SO necesita sus propios controladores, los fabricantes suelen proporcionar controladores para varios de los SO más utilizados [4].

Cada controlador de dispositivo normalmente maneja todo tipo de dispositivo o, cuando más, una clase de dispositivos muy relacionados entre sí. Para acceder al hardware del dispositivo, es decir, los registros de la controladora, por lo general es necesario que el controlador forme parte del *Kernel* del SO, al menos en las arquitecturas actuales. Los controladores de dispositivos comúnmente se ubican debajo del resto del SO.

Los SO por lo general clasifican los controladores en unas cuantas categorías, las más comunes son los dispositivos de bloqueos, como los discos, que contienen varios bloques de datos susceptibles de direccionarse en forma independiente, y los dispositivos de caracteres, como los teclados e impresoras, que generan o aceptan un flujo de caracteres.

Un controlador de dispositivos tiene varias funciones. Las más común es aceptar solicitudes abstractas de leer o escribir enviadas por el software independiente del dispositivo que está más arriba, y encargarse de que se realicen. Pero además debe asignar valores iniciales al dispositivo, en caso de ser necesario, también podría administrar su consumo de energía eléctrica y ausentar

sucesos en un registro. Muchos controladores de dispositivos tienen una estructura general similar. Lo primero que hace un controlador típico es verificar si los parámetros de entrada son válidos. Si no lo son, se devuelve un error. Si los parámetros son válidos, podría requerirse una traducción de términos abstractos concretos.

En el caso de un controlador de disco, esto podría implicar la conversión de un número de bloqueo lineal a los números de cabeza, pista, sector y cilindro apropiados para la geometría del disco [4]. Posteriormente el controlador podría verificar si el dispositivo se está usando. En tal caso, la solicitud se pondrá en cola para procesarse después. Si el dispositivo está inactivo, se examinará el estado del software para ver si ya puede atenderse la solicitud. Podría ser necesario encender el dispositivo y listo para trabajar, podrá iniciarse el control propiamente dicho. Controlar el dispositivo implica enviarle una serie de comandos. El controlador de dispositivos es el lugar donde se determina la sucesión de comandos, dependiendo de la tarea a realizar. Cuando dicho controlador sabe qué comandos va a emitir, comienza a escribirlos en los registros de dispositivo de la controladora. Después de escribir cada comando en la controladora, podría ser necesario verificar que esta lo haya aceptado y esté preparado para recibir el siguiente. Este proceso continúa hasta emitir todos los comandos.

Algunas controladoras pueden aceptar una lista enlazada de comandos y procesarlos por su cuenta sin ayuda del SO. Después de emitir los comandos, se tendrá una de dos situaciones. En muchos casos, el controlador de dispositivo debe esperar hasta que la controladora realice algunas tareas, así que se bloquea hasta que llegue la interrupción que lo desbloquea. En otros casos, la operación termina de inmediato, y el controlador de dispositivo no necesita bloquearse. Un ejemplo de la segunda situación es el desplazamiento vertical de la pantalla en modo de caracteres, lo cual solo requiere escribir unos cuantos bytes en los registros de la controladora.

No se requiere ningún movimiento mecánico, por lo que toda la operación puede terminarse en nanosegundos. En el primer caso, la interrupción despertará al controlador del dispositivo bloqueado. En el segundo, el controlador de dispositivo nunca se dormirá. En ambos casos, una vez que ha terminado la operación, el controlador de dispositivo debe comprobar si hubo errores. Si todo salió bien, dicho controlador podría tener datos que comunicar al software independiente del dispositivo. Por último, el controlador devuelve cierta información de estado a quien lo invocó, para informarle si todo salió bien o si hubo errores, y cuales fueron. Si hay más solicitudes en cola,

podrá seleccionarse una de ellas y atenderse. Si la cola está vacía, el controlador de dispositivo se bloquea en espera de la siguiente solicitud.

Este sencillo modelo no es más que una aproximación burda de la realidad. Muchos factores hacen que el código sea mucho más complicado. Por ejemplo, un dispositivo de E/S podría terminar mientras un controlador de dispositivo se está ejecutando, e interrumpirlo. La interrupción podría hacer que se ejecute un controlador de dispositivo. De hecho, podría hacer que se ejecute el controlador actual. Por ejemplo, mientras en el controlador de red está procesando un paquete que llega, podría llegar otro paquete. Por ello, los controladores deben ser reentrantes, lo que significa que cuando un controlador de dispositivo se está ejecutando, debe considerar la posibilidad de que se le invoque una segunda vez antes de que haya terminado la primera llamada.

En un sistema que permite la conexión en funcionamiento es posible añadir o quitar dispositivos mientras la computadora está encendida. En tal caso, cabe la posibilidad de que, mientras un controlador está ocupado leyendo de un dispositivo, el sistema podría informarle que el usuario ha quitado repentinamente ese dispositivo del sistema. No solo será necesario parar la transferencia de E/S actual sin dañar alguna estructura de datos del *Kernel*, sino que las solicitudes pendientes para el dispositivo deberán eliminarse del sistema en forma delicada, comunicando la mala noticia a quienes las hicieron. Además, agregar inesperadamente nuevos dispositivos podría obligar al *Kernel* a resignar recursos, quitándole unos al controlador y dándole otros a cambio [4].

Los controladores de dispositivos no están autorizados para emitir llamadas al sistema, pero a menudo necesitan interactuar con el resto del *Kernel*. Por lo regular se permiten llamadas a ciertos procedimientos del *Kernel*. Por ejemplo, casi siempre hay llamadas para asignar y liberar páginas físicas de la memoria que se usan como búferes. Se requieren otras llamadas para manejar la Unidad de Gestión de Memoria (*MMU*), temporizadores, la controladora de Acceso Directo a Memoria (*DMA*), la controladora de interrupciones, etc.

Estructura del sistema de E/S de la computadora

Los elementos principales que forman el sistema de E/S son los siguientes:

- Periféricos.
- Módulos de E/S
- Sistemas de interconexión externos.
- Mapa de memoria e instrucciones de E/S.

A continuación, se realiza una breve descripción de estos elementos y de cómo interactúan entre sí.

Periféricos

Son dispositivos que se conectan a la computadora a través de los módulos de E/S y que sirven para almacenar información o para llevar a cabo un tipo específico de comunicación con el exterior, sean las personas, máquinas o con otras computadoras [5].

La clasificación [5] más común es la siguiente:

- Para la interacción con humanos que son: entrada y salida.
- Para la interacción con otras computadoras o sistemas físicos (en los que las operaciones que se hacen son generalmente de E/S) son: el almacenamiento y comunicación.

En un periférico se puede diferenciar en dos partes: una mecánica y una electrónica. La parte mecánica hace funcionar los elementos principales que forman el periférico, tales como el motor para hacer girar un disco o mover el cabezal de una impresora, el botón de un ratón o el láser de un dispositivo óptico. La parte electrónica nos concede, por una parte, generar las señales eléctricas para permitir el funcionamiento de los elementos mecánicos, por otra parte, hacer la conversión de los datos provenientes de la computadora a señales eléctricas o viceversa.

La conexión física entre un periférico y la computadora se lleva a cabo por medio del denominado sistema de interconexión de E/S. Dicho sistema de interconexión de E/S nos permite gestionar las señales de control, de estado y de datos necesarias para llevar a cabo una transferencia de información [5].

La cantidad de información que puede enviar o recibir el periférico por unidad de tiempo la denominamos velocidad de transferencia y generalmente se expresa en bits o bytes por segundo.

La velocidad de transferencia puede ir de unos pocos bits por segundo a Gb por segundo, pero hemos de tener en cuenta que una computadora puede llegar a trabajar a velocidades bastante superiores y hay de garantizar que durante una transferencia no se pierdan datos.

Módulos de E/S

Un módulo de E/S es un controlador de uno o más periféricos que establece una interfaz entre el periférico y la computadora (procesador y memoria) para facilitar la comunicación entre uno y otro de manera que, una parte de los detalles técnicos del periférico queden ocultos al resto de la computadora.

Del módulo de E/S distinguimos tres partes fundamentales (ver Figura 4):

1. Una interfaz interna normalizada con el resto de la computadora mediante el bus de sistema que nos da acceso al banco de registros del módulo de E/S.
2. Una interfaz externa específica para el periférico que controla. Habitualmente la conexión con el periférico se realiza mediante un sistema de interconexión normalizada de E/S.
3. La lógica necesaria para gestionar el módulo de E/S. Esta es responsable del paso de información entre la interfaz interna y externa.

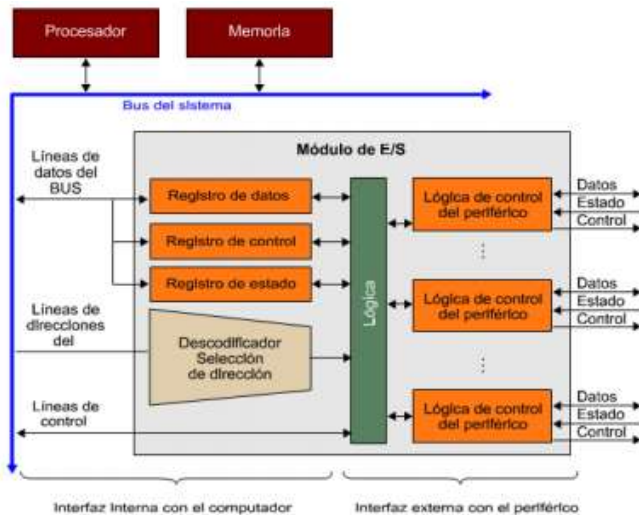


Figura 4. Esquema general de un módulo de E/S.

La forma en que se comunica el módulo de E/S y el periférico es específica para cada periférico. Por lo tanto, depende de las características del periférico que queremos controlar, pero también del sistema de interconexión utilizado para comunicarse. Esta conexión tiene habitualmente unas especificaciones normalizadas y adaptadas al tipo de transferencia que se debe realizar y se denomina sistema de interconexión de E/S. Esto hace que la interfaz externa tenga unas características propias que difícilmente se pueden generalizar [5]. Cuando un módulo de E/S gestiona el funcionamiento de más de un periférico, hay dos configuraciones básicas, la conexión punto a punto y la de multipunto, aunque las configuraciones que encontramos en máquinas reales son muy variadas. En la conexión punto a punto el módulo de E/S gestiona la comunicación con cada periférico individualmente; no es un bus de E/S, pero sí que tiene unas especificaciones normalizadas de la conexión, de manera parecida a las de un bus normalizado de E/S. En cambio, la conexión multipunto el módulo de E/S gestiona la comunicación con los periféricos mediante un bus normalizado de E/S y hay que añadir la lógica para acceder al bus [5].

Sistemas de interconexión externos

En una computadora se distinguen dos tipos básicos de sistemas de interconexión: los internos de la computadora, que nos permiten conectar el procesador, la memoria y el sistema de E/S y que denominamos bus del sistema, y los externos a la computadora, que nos permiten conectar el sistema de E/S con los diferentes periféricos y que se denominan sistemas de interconexión de E/S o buses de E/S.

Desde el punto de vista del sistema de E/S, el bus del sistema nos permite la comunicación entre los módulos de E/S y el resto de la computadora. Este bus tiene una estructura jerárquica formada por diferentes tipos de buses y estos se encargan de aislar los elementos más rápidos de los más lentos y, de esta manera, mejorar las prestaciones del sistema [6]. Los sistemas de interconexión de E/S o buses de E/S permiten la comunicación de los módulos de E/S con los periféricos o dispositivos con suficiente autonomía para gestionar una operación de E/S y los módulos de E/S. Las características de estos sistemas se adaptan al tipo de dispositivos que se ha de conectar.

Físicamente, un sistema de interconexión está formado por un conjunto de hilos conductores o líneas que interconectan diferentes dispositivos. Por estas líneas circulan señales eléctricas que los dispositivos conectados pueden interpretar como señales binarias. Hay tres tipos de señales básicas: señales de datos, de direcciones y de control [6]. Las siguientes son las características principales de los sistemas de interconexión externos:

- **Ancho de banda:** la cantidad máxima de información que podemos transmitir por unidad de tiempo. Se expresa en bits o bytes por segundo.
- **Serie/paralelo:** en una interconexión paralela hay varias líneas que conectan el módulo de E/S y el periférico y pueden transmitir varios bits simultáneamente mediante las líneas de datos. En una interconexión serie solo hay una línea para transmitir los datos y los bits se transmiten uno a uno. Anteriormente las interconexiones de tipo serie eran para dispositivos lentos y las de tipo paralelo, para dispositivos más rápidos, pero con las nuevas generaciones de sistemas de interconexión serie de alta velocidad las paralelas cada vez son menos utilizadas.
- **Punto a punto/multipunto:** una interconexión punto a punto tiene un enlace dedicado entre el módulo de E/S y el periférico. En una interconexión multipunto, que habitualmente se denomina bus de E/S y que dispone de un enlace compartido entre diferentes periféricos y el módulo de E/S, el hecho de tener múltiples

dispositivos conectados a un mismo conjunto de líneas hace necesario establecer mecanismos para controlar el acceso.

Otras características típicas de los buses de E/S son:

- *Modo de operación síncrono/asíncrono/semisíncrono*: si el control de los accesos al bus es controlado o no por un reloj.
- *Multiplexación de datos y direcciones*: si las líneas del bus están dedicadas a datos y direcciones o si se comparten las mismas líneas para datos y para direcciones.
- *Arbitraje centralizado y distribuido*: es centralizado cuando un único árbitro o controlador determina quién tiene que acceder al bus en cada momento y es distribuido cuando los dispositivos conectados al bus disponen de capacidad de controlar el acceso al bus.
- *Tipos de operaciones de lectura/escritura*: diferentes maneras de hacer las operaciones de lectura y escritura, como la transferencia de bloques o la combinación de operaciones de lectura y escritura.
- *Esquema de direccionamiento*: hay dos tipos básicos; el direccionamiento lógico, que es cuando el espacio de direccionamiento de memoria es común a todos los dispositivos y cada uno dispone de un rango de direcciones único y los dispositivos para descodificar la dirección para saber si esta dirección está dentro de su rango; el direccionamiento geográfico, que es cuando cada dispositivo tiene una dirección propia y se separa la identificación del módulo de la selección de la dirección dentro del módulo.

Estructura del software de E/S

Tiene una serie de capas (ver Figura 5) donde las capas inferiores son las que se encargan de esconder las peculiaridades del hardware de las capas superiores y en cambio las capas superiores tienen una interfaz que es agradable hacia el usuario, y es muy útil y sencilla para ser usada por los usuarios.

Tienen como objetivo escribir programas para usar archivos, pero con distintos dispositivos, teniendo en cuenta no modificar ningún programa para cada dispositivo y si se llega a presentar algún problema el SO es el encargado de resolverlo, y sus archivos y dispositivos obtienen una dirección por medio de las rutas que se le asigne con el nombre correspondiente.

Dentro del software se realiza el manejo de errores de E/S, donde es el encargado de manejar lo más cerca posible al error dentro del hardware, los encargados de resolver los problemas son los niveles inferiores y si estos no pueden

resolver, se encargan de avisarle a los informes superiores, y si produce una falla y se requiere hacer alguna recuperación, se lo puede hacer mediante un nivel inferior y de forma transparente.

El SO es el único encargado de realizar las operaciones que son controladas por los interruptores y hace que parezcan del tipo de bloques para el usuario.

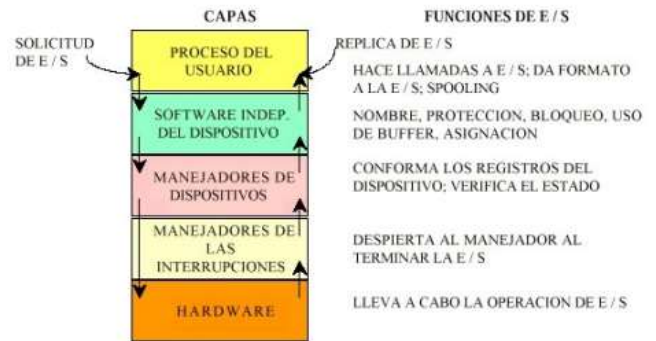


Figura 5. Capas del sistema de E/S y sus principales funciones.

Dentro de la estructura del software de E/S tenemos los manejadores de interrupciones y manejadores de dispositivos y software de E/S en el espacio del usuario.

Manejadores de interrupciones

Los procesos que efectúen un inicio de operación de E/S bloquean hasta que termine la E/S y a su vez ocurra una interrupción. El procedimiento de la interrupción es el encargado de realizar lo necesario para desbloquear los procesos que lo inicia.

Manejadores de dispositivos

Cada controlador posee uno o más registros de dispositivos, ya que tienen una funcionalidad por medios de comandos y así verifican su ejecución que sea muy útil. El manejador de dispositivos es el encargado de aceptar todas las solicitudes abstractas que hace que el software (sea) independiente del dispositivo y a su vez verifica las ejecuciones de las solicitudes.

Software de E/S en el espacio del usuario

Las bibliotecas estándar de E/S tienen varios procedimientos que están relacionados con la E/S y aquí todos se ejecutan como una parte de los programas del usuario. Para imprimir un archivo, un proceso es el que genera todo el archivo y después lo coloca dentro del directorio de spooling [6].

Métodos de E/S

- *Por programas*: este es un método el cual nos brinda una forma más sencilla de realizar operaciones de E/S. Este proceso ocurre cuando el procesador y el dispositivo coinciden en el

mismo tiempo definido por la computadora, después el procesador interactúa directamente con el dispositivo de E/S comprobando el estado o si tiene algún dato listo para así poder recibirlo [2].

- *Por interrupciones:* en este método el procesador espera un tiempo indefinido hasta que el módulo que es el encargado de mostrar la operación de E/S, esté preparado para ejecutar la operación.
- *El microprocesador:* este método se encarga de decidir el estado de los dispositivos indicando información del dispositivo como: datos listos para transmitir, dispositivo ocupado, dispositivo no conectado o error, etc.
- *E/S sincrónica:* retorna al programa usuario sin esperar que la operación de E/S finalice. Se necesita una llamada al sistema que le permita al usuario esperar por la finalización de E/S, si es requerido.
- *DMA:* este método tiene la habilidad de transferir directamente la información entre la memoria y los dispositivos de entrada o estos, a la memoria sin la necesidad de un canal que intercambie datos.
- *Buffer de E/S:* Estos son espacios de memoria principal que se mantienen solo para almacenar datos que envían de los dispositivos.

Conclusiones

Una de las características principales de un SO es el control de todos los dispositivos de E/S de la computadora, esto comprende desde la transferencia entre diversos niveles de memoria hasta la comunicación entre los periféricos, además de que la interfaz que presente sea sencilla y fácil de implementar. El SO tiene la obligación de controlar todos los dispositivos E/S para cumplir con objetivos fundamentales para su uso, tales como:

- a) Facilitar el manejo de los dispositivos periféricos.
- b) Optimizar la E/S del sistema.
- c) Proporcionar dispositivos virtuales que permitan conectar cualquier tipo de dispositivos físicos sin que sea necesario remodelar el sistema de E/S del SO.
- d) Permitir la conexión de dispositivos nuevos de E/S, solventando de forma automática su instalación usando mecanismos del tipo Plug Play.

Los dispositivos E/S abarcan un considerable rango de velocidades, por esta razón se le impone una presión considerable al software con el fin recibir un buen desempeño sobre muchas ordenes de magnitud. Ahora

bien, hay que considerar que cada controlador posee registros los cuales son utilizados para establecer la comunicación al CPU donde pueden ser parte del espacio normal de las direcciones de la memoria de E/S mapeada a la memoria. Por lo que cada dispositivo de E/S que se encuentre conectado a la computadora necesita un código específico, el mismo que servirá para controlarlo, a este código se denomina controlador de dispositivo, y generalmente es escrito por el fabricante y es proporcionado junto con el hardware. Ya que cada SO, sea cual sea necesita sus propios controladores.

Un controlador de dispositivos tiene a cargo varias funciones, la más evidente es aceptar las solicitudes abstractas de leer o escribir enviadas por el software independiente del dispositivo que está más arriba, y encargarse de que se satisfagan, administrar el consumo de energía, ausentar sucesos en un registro, asignar valores iniciales al dispositivo como tal. Para controlar un dispositivo es necesario enviarle una serie de comandos, el controlador de dispositivos es ese sitio donde se determina la sucesión de comandos, pero esto dependiendo de la tarea que se va a realizar.

Este sencillo modelo no es más que una aproximación burda de la realidad. Muchos factores hacen que el código sea mucho más complicado. Un ejemplo de esto, un dispositivo de E/S podría terminar mientras un controlador de dispositivo se está ejecutando, e interrumpirlo. La interrupción podría hacer que se ejecute un controlador de dispositivo. Incluso, podría hacer que se ejecute el controlador actual. Por ejemplo, mientras en el controlador de red está procesando un paquete que llegó, podría llegar otro paquete. Por ello, los controladores deben ser reentrantes, lo que significa que cuando un controlador de dispositivo se está ejecutando, debe considerar la posibilidad de que se le invoque una segunda vez antes de que haya terminado la primera llamada.

Referencias

- [1] A. S. Tanenbaum, Sistemas Operativos Modernos, Tercera ed., México: PEARSON EDUCACION,(2009).
- [2] T. A. S., Sistemas Operativos Modernos, 2da Edición ed., México: Pearson Education 2003, p. 825.
- [3] M. D. L. L. R. MARTINEZ, SISTEMAS OPERATIVOS, U.N.N.E ARGENTINA, 2015.
- [4] W. Stallings, Sistemas Operativos Aspectos internos y principios de Diseño, Pearson Educación, S.A, 2005.
- [5] Miquel Albert Orega, Gerard Enrique Manonellas, Sistema de entrada/salida, Pearson Educación, S.A, 2005.
- [6] WebMaster. (Exa, Editor, David, Productor), <http://exa.unne.edu.ar/informatica/SO/SO5.htm> PSES, Pearson Educación, 2018.