

Entrada/Salida (E/S) en Computadoras

1. Introducción

En la arquitectura de computadoras, la conexión entre el teclado/monitor (E/S), la CPU y la memoria es muy importante para el funcionamiento y uso de cualquier sistema. Esta interacción no solo depende de cómo pasa la información físicamente y lógicamente, sino también de cómo se organizan y hacen mejor los procesos para que la información llegue de la forma más eficiente posible. Un caso ilustrativo de la importancia de un planteamiento adecuado lo encontramos en Structured Input-Output Tools for Modal Analysis of a Transitional Channel Flow, en el que se utilizan herramientas estructuradas de E/S para el análisis de flujos transicionales complejos. Es en este trabajo en el que disponer de un sistema de E/S que opera de manera adecuada nos permite capturar y detectar de manera precisa los puntos existentes y las estructuras coherentes, sacando todo el partido de la capacidad de cálculo sin tirar recursos por errores [1].

Por otro lado, Advances in Microprocessor Cache Architectures Over the Last 25 Years traza el desarrollo de las arquitecturas de caché en microprocesadores que han cambiado las reglas de cómo los datos son transferidos de la CPU a la memoria principal y los periféricos. La inteligente introducción de niveles de caché (L1, L2, L3) y técnicas avanzadas de particionamiento han conseguido reducir este retardo por un lado y, por otro, mejorar el rendimiento de las aplicaciones, especialmente en escenarios en que es necesario aumentar la capacidad de almacenamiento y al mismo tiempo ofrecer un rápido acceso a un gran volumen de datos [2].

En su conjunto, ambos métodos muestran un mismo principio: el trabajo de los movimientos de E/S necesita soluciones de hardware buenas, bases de datos mejoradas y estructuras para guardar informaciones fuertes. Cuando se trata de imitar un hecho físico, hay que programarlo para usar E/S oculta, E/S que para, acceso recto a memoria (DMA) o algo parecido, luego de pensar cómo se moverá la información por el sistema de manera eficaz. Al juntar lo teórico, el diseño y la practica, nosotros podemos trabajar en sistemas que sean capaces de procesar los datos más rápido seguro y esperado. Es muy importante para crear tecnología nueva y resolver problemas más difíciles [1], [2].

2. Conceptos básicos de Entrada y Salida

En el ámbito de los computadores, la Entrada/Salida (E/S) es el conjunto de transacciones, formas y dispositivos que un sistema permite operar con los objetos de fuera, mezclando datos entre la CPU, la memoria y los dispositivos conectados. Este proceso es el fundamento para que los programas se puedan beneficiar de datos procedentes de sensores, soportes, redes u otros sistemas, para luego producir salidas a raíz del almacenamiento, la representación o la remisión de datos. La E/S es por ello el vínculo entre el funcionamiento interno de la máquina y las fuentes o destinos que ocupan información exterior [3].

En sistemas de alto rendimiento, como los que ADIOS2 gestiona, la entrada/salida es crucial por la enorme cantidad de datos de simulaciones científicas, análisis masivos y computación distribuida. Aquí, los conceptos básicos de E/S incluyen la comunicación con dispositivos, la optimización de velocidad, la baja latencia y la gestión de grandes flujos de información. En este entorno, el diseño de un sistema de E/S debe pensar en cómo reducir los cuellos de botella y usar al máximo el hardware disponible se pueden clasificar en varias categorías [1], [3]:

- **E/S programada**, donde la CPU controla directamente la transferencia de datos, ejecutando instrucciones específicas para comunicarse con el dispositivo.
- **E/S mediante interrupciones**, que permite a los dispositivos notificar a la CPU cuando están listos para enviar o recibir datos, liberando a la CPU de la espera activa.
- **Acceso Directo a Memoria (DMA)**, que habilita la transferencia de datos entre memoria y dispositivo sin intervención constante de la CPU, mejorando la eficiencia.

Con ello, la E/S deja de ser solo un canal de comunicación básico para convertirse en un componente estratégico que define el rendimiento global de un sistema [3].

3. Interacción entre CPU, memoria y dispositivos de E/S

Un ordenador debe permitir a los usuarios o a otros ordenadores encontrar una relación definida y determinada en el sentido establecido antes para poder llevar a cabo el tratamiento de la información que se quiere obtener. La forma en que interactúan el microprocesador, la memoria y la E/S es crucial para el adecuado tratamiento informático de la información. El microprocesador realiza las instrucciones y controla la información, pero la memoria sirve para almacenar temporalmente los datos y los programas necesarios para su funcionamiento. En cuanto a los periféricos, estos son los elementos que permiten que el sistema intercambie información con el exterior comunicándose obteniendo información (entrada) o enviándola (salida) [4].

En este proceso, la CPU asume la función de la dirección de todo, controlando con instrucciones precisas las peticiones de entrada y salida; es decir, controla cuándo y cómo se produce la transferencia de datos entre los dispositivos y la memoria. Para mejorar este proceso, se pueden utilizar técnicas como la E/S programada, las interrupciones y el acceso directo a la memoria (DMA), con el objetivo de reducir los tiempos de espera y aumentar el rendimiento del sistema; la intención es conseguir que los datos circulen sin atascos y que la lentitud de un dispositivo no acabe por deteriorar el rendimiento del sistema [4], [5].

4. Módulos de E/S

La tarea del controlador de E/S es controlar, dirigir y mantener todo el sistema de periféricos. Entre sus funciones se incluye traducir las señales energéticas o analógicas en información binaria y viceversa. También se encarga de hacer que el ritmo de los dispositivos periféricos, que suelen funcionar más despacio, sea compatible con el de la CPU, que es muy rápido. Esto lo consigue almacenando temporalmente los datos en una memoria llamada buffer cuando los dispositivos periféricos no están preparados para trabajar o bien leyendo los datos a un ritmo mayor cuando lo requiere el dispositivo periférico. El controlador de E/S también tiene que detectar errores y, si los hay, determinar con qué periférico se ha producido el error y qué tipo de error ha sido y, a continuación, informar a la CPU. Para ello, utiliza varios registros especiales que posee, y que la CPU puede examinar y programar, de forma similar a como se hacía con el conjunto de registros del procesador antes de que se iniciara la transferencia [6].

5. Métodos de Entrada/Salida

A la hora de diseñar ordenadores, las operaciones de E/S (entrada y salida) se pueden realizar por tres métodos: E/S programada (polling), E/S por interrupción, y la propia memoria (DMA). En la E/S programada o polling, la CPU controla el dispositivo directamente y espera activamente hasta que el dispositivo está preparado para enviar o recibir datos, consultando frecuentemente su registro de estado. Es un método fácil de implementar y sirve bien para dispositivos rápidos o

procesos extremadamente cortos, pero es un sistema ineficiente para dispositivos lentos, donde el procesador estará inactivo durante largos períodos en espera. Un caso típico será leer la tecla presionada de una manera que estará comprobando continuamente si hay datos disponibles en el buffer del teclado [5], [6].

Mediante el uso de entrada/salida a través de interrupciones, el CPU no se encuentra sondeando el estado del dispositivo, sino que sigue con otras tareas; cuando el periférico se encuentra listo, éste envía una señal de interrupción que provocará que el CPU detenga provisionalmente la actividad, atienda la solicitud y tras ello vuelva a lo que estaba haciendo. Este procedimiento mejora el funcionamiento de los recursos del procesador; además, le va muy bien a los dispositivos lentos o a aquellas tareas que se producen de forma ocasional, aunque necesita de un cuidado mayor y puede llegar a complicarse en caso de que haya en el sistema un gran número de interrupciones. Un caso habitual es el de una tarjeta de red, que interrumpe al procesador al haber recibido un paquete de datos [2], [6].

Finalmente, la transferencia directa de datos a memoria, o DMA (Direct Memory Access), permite que un controlador específico se encargue de la transferencia de datos entre memoria y dispositivo sin que la CPU tenga que estar siempre ocupada con ese tipo de tarea. El procesador sólo tiene que especificar la dirección de memoria, la dirección de dispositivo y la cantidad de datos que se tienen que transferir en la correcta configuración del controlador DMA y después el DMA se encarga, por sí mismo, de enviar o recibir los datos y efectuar el correspondiente aviso a la CPU. Se trata de un método muy eficiente para mover muchos datos con el hecho soportado de que la CPU queda liberada para ocuparse o realizar otras tareas, aunque requiere hardware adicional y genera competencia por el uso de la memoria. Un ejemplo de este uso sería la transferencia de un archivo desde disco a la RAM [6].

BIBLIOGRAFIA

- [1] “Structured Input-Output Tools for Modal Analysis of a Transitional Channel Flow,” Jan. 23, 2023. doi: 10.2514/6.2023-1805.vid.
- [2] R. Iyer *et al.*, “Advances in Microprocessor Cache Architectures Over the Last 25 Years,” *IEEE Micro*, vol. 41, no. 6, pp. 78–88, Nov. 2021, doi: 10.1109/MM.2021.3114903.
- [3] W. F. Godoy *et al.*, “ADIOS 2: The Adaptable Input Output System. A framework for high-performance data management,” *SoftwareX*, vol. 12, p. 100561, Jul. 2020, doi: 10.1016/j.softx.2020.100561.
- [4] L. Robaldo, C. Bartolini, M. Palmirani, A. Rossi, M. Martoni, and G. Lenzini, “Formalizing GDPR Provisions in Reified I/O Logic: The DAPRECO Knowledge Base,” *J Logic Lang Inf*, vol. 29, no. 4, pp. 401–449, Dec. 2020, doi: 10.1007/s10849-019-09309-z.
- [5] D. Comer, *Essentials of Computer Architecture*. Boca Raton: Chapman and Hall/CRC, 2024. doi: 10.1201/9781003410140.
- [6] D. P. López Carrillo, Á. M. Acuña Félix, R. F. Tipan Tisalema, G. I. Vanegas Zabala, and D. F. Yumisa León, *Arquitectura de computadoras*. Centro de Investigación y Desarrollo Ecuador, 2025. doi: 10.33996/cide.ecuador.AC2679376.