

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI-0118 Lenguaje Ensamblador

Tema: Drivers en Windows

Integrantes:

Madrigal Barrantes Braunny (C24436)

Picado Leiva Marcelo (C15950)

Rojas Carrillo Henoc (C26764)

Villafuerte Beita Kendall (C28387)

I Semestre

2024

Tabla de contenidos

Portada.....	p. 01
Tabla de contenidos.....	p. 02
Introducción.....	p. 03
Antecedentes.....	p. 03
Objetivos.....	p. 03
Alcances y limitaciones.....	p. 04
Marco teórico.....	p. 04
Metodología.....	p. 05
Tipo de investigación.....	p. 05
Enfoque de la investigación.....	p. 06
Técnicas de investigación	p. 06
Capítulo I: Concepto y función de los drivers en Windows.....	p. 06
Capítulo II: Tipos de drivers.....	p. 07
Análisis de resultados.....	p. 08
Conclusiones.....	p. 08
Bibliografía.....	p. 09

Introducción

Para Windows, es especialmente importante entender cómo se da la conexión entre los diversos sistemas de entrada y salida que solo pueden ser programados al entender los drivers como piezas transparentes (Oney, 2002). Y es que, los sistemas informáticos dependen en gran medida de la interacción entre el software y el hardware. Esta sinergia es posible gracias a los controladores o drivers, que actúan como intérpretes entre el sistema operativo y los componentes físicos de la computadora.

Antecedentes

La historia de los controladores inicia en 1980, con los primeros ordenadores que funcionaban con chips de procesador Intel los cuales tenían una capacidad de direccionamiento limitada y los controladores eran programas en lenguaje ensamblador que se instalaban a través de un archivo llamado CONFIG.SYS, escritos principalmente para sistemas de 16 bits.

A medida que avanzaba la tecnología de procesadores, surgieron los primeros controladores de Windows en la década de 1990, los cuales eran conocidos como MS-DOS, estaban escritos en ensamblador y eran usados para manejar hardware estándar como pantalla, teclado y ratón. Sin embargo, MS-DOS no era capaz de llevar muchas tareas a la vez, esto llevó a la necesidad de crear una máquina virtual, donde las aplicaciones compartieran el hardware físico.

Posteriormente, Windows 3.0 introdujo la idea de controladores de dispositivo virtuales para "virtualizar" hardware, permitiendo la aparición de máquinas virtuales equipadas con instancias independientes de muchos dispositivos de hardware. Con el lanzamiento de Windows NT a principios de la década de 1990, se introdujo una nueva tecnología de modo de núcleo para los controladores, independiente de las tecnologías anteriores. A partir de Windows 95, la función Plug and Play automatizó la identificación y configuración del hardware, reduciendo la necesidad de controladores en modo real. En la actualidad, el estándar Plug and Play y el bus PCI han simplificado enormemente la instalación y configuración de hardware en los ordenadores modernos.

Objetivos

Objetivo General:

-Analizar el papel de los drivers en el sistema operativo Windows y su impacto en el sistema.

Objetivos Específicos:

- Definir el concepto y describir las funciones de los drivers en Windows.
- Examinar los diferentes tipos de drivers disponibles en Windows.

Alcances y limitaciones

El alcance de esta investigación se limita al estudio de los drivers en el sistema operativo Windows. Se abordarán aspectos teóricos, metodológicos y prácticos, sin embargo, no se profundizará en el desarrollo de drivers específicos ni se hará un análisis exhaustivo de los manuales de usuario de Windows.

Marco Teórico

Sistema operativo Windows

Basándonos en la cita proporcionada del libro "Operating Systems: Internals and Design Principles" de Stallings (2005), podemos definir el concepto de sistema operativo Windows de la siguiente manera:

El sistema operativo Windows se caracteriza por funcionar bajo un modelo de arquitectura modular. En esta arquitectura, el sistema operativo está dividido en componentes, donde cada componente se encarga de una función específica. La modularidad del sistema permite que cada módulo pueda ser actualizado o removido de forma independiente. Esto implica que, al referirse al sistema operativo Windows en general, se está haciendo alusión a la totalidad de sus componentes y funcionalidades, los cuales están organizados de manera modular y pueden ser gestionados de forma independiente.

Esta definición resalta la estructura modular del sistema operativo Windows, donde la capacidad de actualizar o remover componentes individualmente contribuye a su flexibilidad y mantenimiento a lo largo del tiempo.

Windows funciona bajo un modelo de arquitectura modular. El sistema operativo está dividido en componentes y cada componente se encarga de una función específica. En principio cada módulo del sistema operativo puede ser actualizado o removido de forma independiente y es por ello que al hablar del sistema operativo se habla de él mismo en general (Stallings, 2005).

Este enfoque modular se refleja en la manera en que el sistema operativo maneja los controladores o "drivers".

Drivers

La conceptualización del término "Driver" se puede abordar desde su origen etimológico y su aplicación en el ámbito de la informática, según lo planteado por Benjamin Veschi en su artículo "Driver" publicado en diciembre de 2018.

El término "Driver", originario del inglés, se remonta al siglo XVIII y se ha incorporado al español como "controlador", entendido como una pieza de software esencial para la interacción entre un sistema informático y el hardware asociado. Veschi señala que, si bien en algunos contextos se emplea el término "controlador" o "manejador" en países de habla hispana, como una traducción directa, en lugares como España, México y Argentina es común el uso directo del término en inglés.

El concepto deriva del verbo inglés "drive", que significa manejar o dirigir, al que se le añade el sufijo "-er" para convertirlo en un sustantivo. Esta estructura lingüística refleja la función del software en la gestión y dirección de la interacción entre el sistema operativo, las aplicaciones y el hardware de una computadora. El propósito fundamental de un "Driver" es permitir que el sistema operativo y las aplicaciones emitan órdenes específicas al hardware, facilitando así su funcionamiento adecuado.

Los drivers son programas diseñados para permitir que el sistema operativo de una computadora se comunique directamente con el hardware. Por ejemplo, si un usuario desea imprimir un documento, los drivers de la impresora deben de interpretar los comandos del sistema operativo en un lenguaje que la impresora entienda.

Dichos drivers se comunican de los dispositivos de entrada y salida al buffer del kernel en el sistema operativo, dicha comunicación no está bien documentada para el sistema Windows y es por ello importante su estudio (Tsegaye & Foss, 2004).

Metodología

Tipo de Investigación

La investigación será de carácter descriptivo, utilizando enfoques cualitativos para examinar las características, tipos y el impacto de los drivers en el sistema operativo Windows.

Enfoque de la Investigación

Se utilizarán métodos cualitativos para entender las experiencias y opiniones de usuarios y profesionales de informática.

Técnicas de Investigación

Se revisarán estudios previos, documentaciones técnicas y otros recursos relevantes. Además, se examinarán manuales y guías de instalación de drivers.

Capítulo I: Concepto y función de los drivers en Windows

Cuando nos referimos a Drivers o Controladores, estamos hablando de un tipo de software en específico que cumple con una tarea fundamental en el correcto funcionamiento de una computadora. En Windows el concepto correcto para drivers es un componente de software que permite la comunicación entre un dispositivo y el sistema operativo. Dado que las computadoras están compuestas por múltiples componentes de hardware en coexistencia, la existencia de estos componentes de software es imprescindible, ya que su función es cumplir de intermediarios, traduciendo solicitudes del sistema operativo en comandos que el hardware pueda comprender y viceversa. Un ejemplo de esto son los drivers encargados de comunicar al sistema operativo con tarjetas de video, dispositivos de audio, unidades de almacenamiento, entre otros.

Los drivers cumplen un rol fundamental en el funcionamiento adecuado de una computadora, garantizando compatibilidad e intercomunicación entre hardware y software. Sin los drivers adecuados un sistema informático puede sufrir diferentes consecuencias como, incompatibilidad, inestabilidad, ralentización de procesos e incluso puede causar que los dispositivos dependientes de estos drivers no sean reconocidos.

Es importante recalcar que no todos los controladores se comunican directamente con un dispositivo, pues muchos de los controladores cumplen únicamente la función de intermediar entre un controlador y otro, lo cual se conoce como una pila de controladores, donde arriba de la pila se encuentra la aplicación o programa que hace una solicitud al sistema operativo Windows, que a su vez hace una solicitud a los controladores que se encuentran por debajo, estos controladores se van comunicando la solicitud entre sí hasta llegar al dispositivo. Entonces de habla de controladores de filtro, que llevan las solicitudes y verifican que estas no contengan

errores, y el controlador de función que es quien conecta directamente con el dispositivo, llevando solicitudes y recibiendo sus entradas.

Retomando el controlador de función, estos se conectan directamente al dispositivo si este está conectado directamente al bus de interconexión de componentes periféricos (conocidos como dispositivos PCI). Dicho controlador obtiene las direcciones asignadas al puerto y a los recursos de memoria del dispositivo, y este controlador se comunica con el dispositivo escribiendo en dichas direcciones.

En resumen, los controladores son quienes comunican los dispositivos con el sistema Windows, pero existen varios tipos de controladores y todos trabajan en conjunto transportando la información para que dicha comunicación se pueda dar de manera eficiente.

Capítulo II: Tipos de drivers

En general, en el sistema operativo Windows hay dos modos en que pueden operar los drivers. El primero es el modo de usuario y el segundo es el modo kernel.

En modo usuario, el código posible a ejecutar está restringido a únicamente operaciones seguras. El código no es capaz de tocar la memoria perteneciente a otras aplicaciones o interactuar con dispositivos de entrada o salida. Un concepto necesario para operar drivers en modo usuario es conocer como el dispositivo de hardware específico hace las llamadas al Kernel para pedir permisos, si es que son requeridos (Baker & Lozano, 2001).

Por otra parte, en modo kernel el código tiene disponible la totalidad de instrucciones del CPU. Dicha disponibilidad incluye manipulaciones de memoria potencialmente peligrosas, ya que si un dispositivo corre en modo kernel, en esencia lo que hace es tener acceso a instrucciones a nivel de sistema (Baker & Lozano, 2001).

Decidir en qué modo deben ser escritos los drivers es una tarea por sí misma. Para tomar dicha decisión ya existen plantillas preconcebidas dependiendo del dispositivo de hardware y por ende el tipo de driver que él mismo puede requerir.

Entre los tipos de drivers más comunes están los de impresión, multimedia y red. Los de impresión operan haciendo uso de dos conceptos; gráficos de impresión, ellos se encargan de la renderización para el dispositivo de hardware requerido y la interfaz de impresión, que es la que se encarga de la comunicación entre el periférico y el sistema operativo. Los drivers de multimedia utilizan un objeto filtro que se encarga de administrar eventos; los eventos son

producidos por el periférico y consumidos por el sistema operativo. Por último, los de red operan bajo un sistema de capas que el propio Windows administra, en general los drivers de red operan comunicando protocolos con puertos utilizando el sistema de capas de la versión de Windows pertinente (Baker & Lozano, 2001).

En general, a la hora de hablar de drivers se debe de hablar de las estructuras, que componen a cada controlador. Para cada controlador existen modelos documentados con especificaciones y regulaciones a cumplir, por ejemplo, especificaciones que dicen al kernel que rutas puede tomar. Es por ello que se recalca la importancia de conocer qué tipo de driver se quiere desarrollar, antes de empezar la implementación del mismo.

En relación a lo anterior, una forma de saber que tipo de driver se requiere, es conociendo qué especificaciones típicas el mismo puede requerir. Por ejemplo, la mayoría de drivers requieren que se le especifique qué caminos puede tomar para ejecutar código, que secciones de memoria tiene permitido leer o escribir, bajo qué condiciones se puede hacer lecturas de estados y en general las especificaciones se encargan de definir como el periférico y en qué estados de ejecución puede comunicarse con el sistema operativo, sin el conocimiento de dichos estándares errores pueden surgir y situaciones críticas adversas puede ser el detonante de una falla completa en el equipo (Matoušek, 2005).

Análisis de Resultados

El estudio muestra que los modos de los drivers, usuario y kernel, tienen impactos significativos en la seguridad y la estabilidad del sistema. El modo kernel facilita funciones complejas pero con riesgos mayores, mientras que el modo usuario asegura mayor estabilidad al limitar las operaciones.

Conclusiones

Los drivers son esenciales para asegurar la compatibilidad y estabilidad del sistema operativo Windows. Una elección adecuada y gestión efectiva de estos componentes son cruciales para evitar fallos y optimizar el rendimiento.

Los resultados destacan la importancia de un diseño informado de drivers que equilibre funcionalidad y seguridad. Por ello, los administradores y desarrolladores deben interesarse por el estudio de los drivers que manejan el sistema operativo en el cual trabajan.

Bibliografía

Baker, A., & Lozano, J. (2001). The Windows 2000 device driver book: a guide for programmers. Prentice Hall Professional.

Oney, W. (2002). *Programming the Microsoft Windows driver model*. Microsoft Press.

Tsegaye, M., & Foss, R. (2004). A comparison of the Linux and Windows device driver architectures. *ACM SIGOPS Operating Systems Review*, 38(2), 8-33

Stallings, W. (2005). The windows operating system. *Operating Systems: Internals and Design Principles*.

Matoušek, T. Model of the Windows Driver Environment (Doctoral dissertation, Master Thesis at Department of Software Engineering, Charles University in Prague, 2005, <http://nenya.ms.mff.cuni.cz/publications/Matousek-thesis.pdf>).

Microsoft. (2023). ¿Qué es un controlador? Microsoft Learn. <https://learn.microsoft.com/es-es/windows-hardware/drivers/gettingstarted/what-is-a-driver->

Veschi, B. (2018, diciembre). Driver [Artículo web]. Recuperado de <https://etimologia.com/driver/>