

MÓDULO 2 MF0964_3: DESARROLLO DE ELEMENTOS DE SOFTWARE PARA GESTIÓN DE SISTEMAS

UNIDAD 2 UF1287: DESARROLLO DE
COMPONENTES SOFTWARE PARA EL
MANEJO DE DISPOSITIVOS (DRIVERS)

2 Programación de controladores de
dispositivo

CORE
networks



2 Programación de controladores de dispositivo

2.1 Introducción

CORE
networks

Controladores de dispositivo

Un controlador de dispositivo (driver) es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz (posiblemente estandarizada) para utilizar el dispositivo.

Es una pieza esencial del software, y en particular, del núcleo de un sistema operativo, sin la cual el hardware sería inutilizable.



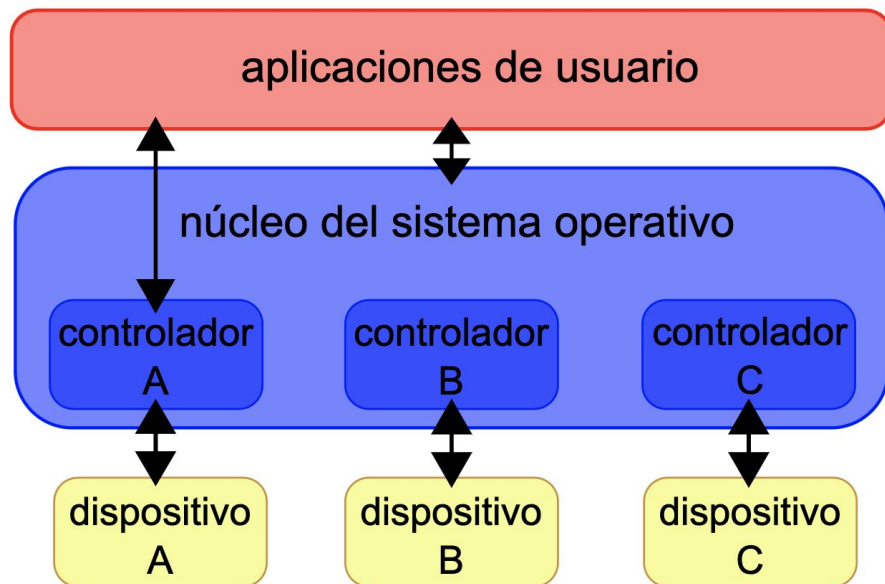
2 Programación de controladores de dispositivo

2.2 Funcionamiento general de un controlador de dispositivo

CORE
networks

Funcionamiento de los controladores de dispositivo

Los sistemas operativos están preparados para incorporar los controladores, de forma que puedan interactuar con el dispositivo y con el resto del sistema operativo.



Interfaces de los controladores de dispositivo

- Con el núcleo del sistema operativo, de manera que comunica las peticiones y permite el acceso a los servicios de este.
- Con el bus, para gestionar la comunicación con el dispositivo.
- Con el dispositivo, para ejecutar las operaciones solicitadas.

[Ver esquema página 60](#)



2 Programación de controladores de dispositivo

2.3 Principales tipos de controladores de dispositivo

CORE
networks

Tipos de controladores de dispositivo

- Caracter. La transferencia de información se realiza mediante una secuencia de caracteres con un orden específico.
 - Puertos.
 - Teclados.
 - Impresoras.
 - Tarjetas gráficas.
- Bloque. Los que son capaces de montar un sistema de fichero.
 - Discos duros.
 - Memorias USB.

Tipos de controladores de dispositivo(II)

- Paquetes. La transferencia de datos se hace en unidades de tamaño fijo (paquetes)
 - Tarjetas de red.
 - Tarjetas WiFi



2 Programación de controladores de dispositivo

2.4 Técnicas básicas de programación de controladores de dispositivos

CORE
networks

Estructuras básicas de datos

Los sistemas cuentan con dos tipos de elementos que interactúan sobre los dispositivos.

- Módulo de E/S.
- Controlador de dispositivo.

Ver esquema página 67 y 68.

Errores de dispositivos

- Errores de programación.
- Errores en los parámetros.
- Errores propios de las E/S.
- Errores sin clasificar.

Gestión de memoria de dispositivos

Para gestionar el paso de datos entre los dispositivos y la memoria se utilizan varias soluciones:

- Zonas de almacenamiento intermedio (caché).
- Buses de interconexión de mayor velocidad.
- Multiprocesadores.

Gestión de puertos de entrada y salida

Los dispositivos están conectados a través del bus de conexión y disponen de una serie de direcciones, llamados puertos de E/S, que sirven como enlace para que puedan comunicarse.

[Ver esquema página 97](#)

Uso de Acceso Directo a Memoria (DMA)

Las E/S programadas provocan que la CPU se dedique en exclusiva a la operación que esté realizando y, aunque se usen interrupciones para liberarla, estas interrupciones a su vez necesitan instrucciones que requieren CPU.

Para mejorar el rendimiento se utiliza el Uso de Acceso Directo a Memoria (DMA), un módulo que puede leer y escribir en la memoria principal los datos utilizados en los dispositivos de E/S.

[Ver esquema página 104](#)



2 Programación de controladores de dispositivo

2.5 Técnicas de depuración y prueba

CORE
networks

Depuración

La depuración de programas es el proceso de identificar y corregir errores de programación.

En inglés se conoce como debugging, porque se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación.



2 Programación de controladores de dispositivo

2.6 Compilación y carga de controladores de dispositivos

CORE
networks

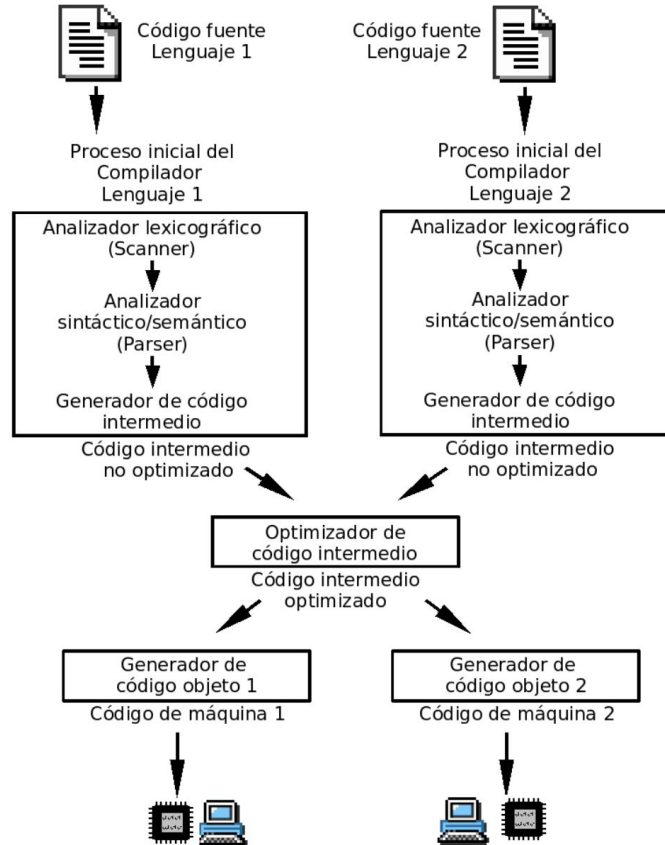
Compilación

En programación, un compilador es un tipo de traductor que transforma un programa entero de un lenguaje de programación (llamado código fuente) a otro.

Usualmente el lenguaje objetivo es código máquina, aunque también puede ser traducido a un código intermedio (bytecode) o a texto.

A diferencia de los intérpretes, los compiladores reúnen diversos elementos o fragmentos en una misma unidad (un programa ejecutable o una librería), que puede ser almacenada y reutilizada. Este proceso de traducción se conoce como compilación.

Compilación (II)





2 Programación de controladores de dispositivo

2.7 Distribución de controladores de dispositivo

CORE
networks

Distribución

Una distribución de software, es un conjunto de software específico (o una colección de múltiple software, incluso un sistema operativo), ya compilado y configurado para ser instalado y usado.

Por distribución también nos referimos a la forma en la que el end user conseguirá el software, por ejemplo descargándolo de internet, a través de virtualización, directamente en un intérprete como el navegador web, etc.



2 Programación de controladores de dispositivo

2.8 Particularidades en el desarrollo de dispositivos en sistemas operativos de uso común

CORE
networks

Dispositivos Plug & Play

Plug-and-Play o PnP (en español "enchufar, conectar y usar") es la tecnología o cualquier avance que permite a un dispositivo informático ser conectado a una computadora sin tener que configurar, mediante jumpers o software específico (no controladores) proporcionado por el fabricante, ni proporcionar parámetros a sus controladores.

Para que sea posible, el sistema operativo con el que funciona el ordenador debe tener soporte para dicho dispositivo.



2 Programación de controladores de dispositivo

2.9 Herramientas

CORE
networks

Entorno de desarrollo

Un entorno de desarrollo es el conjunto de elementos de software y los requerimientos de hardware necesarios de los mismos, para que un desarrollador o un equipo de desarrolladores pueda llevar a cabo una determinada fase del desarrollo de software.

Incluye elementos como:

- Entornos de ejecución e intérpretes.
- Librerías de desarrollo o código.
- Depuradores y compiladores.
- Editores de código o IDEs (Entornos de desarrollo integrados).
- Sistemas de control de versiones y despliegue.
- Herramientas de comunicación entre equipos.



2 Programación de controladores de dispositivo

2.10 Documentación de manejadores de dispositivo.

CORE
networks

Documentación

Todos los procesos del ciclo vital del software en el desarrollo de cualquier software se registran como documentación.

La documentación sirve como información escrita sobre definición de requerimientos, especificaciones generales del sistema, especificación de cada componente y los planes integrales de prueba y mantenimiento. Las herramientas de gestión de configuración también son de gran utilidad en la documentación del software.

Se puede distinguir entre:

- Documentación de software para desarrolladores.
- Documentación de software para usuarios finales.



2 Programación de controladores de dispositivo

2.11 Resumen

CORE
networks

Resumen

- Funcionalidades del núcleo del sistema operativo.
- Subsistemas del núcleo del sistema operativo.
- Aspectos de seguridad en sistemas operativos.