

UNIDAD 4. PROGRAMACIÓN DE DISPOSITIVOS

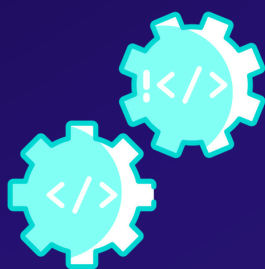
La programación de dispositivos controla el hardware usando lenguajes de bajo nivel, como ensamblador, para gestionar sus funciones específicas.

¿QUÉ ES LA PROGRAMACIÓN DE DISPOSITIVOS DESDE EL LENGUAJE ENSAMBLADOR?

La programación de dispositivos en ensamblador controla el hardware mediante instrucciones de bajo nivel y manipulación de registros y memoria.

ACCESO A REGISTROS

Uso de registros de control para manejar la comunicación con los dispositivos.



INSTRUCCIONES DE ENTRADA/SALIDA (I/O)

Cómo utilizar las instrucciones de ensamblador para leer o escribir datos en puertos I/O.

INTERRUPCIONES

Uso de interrupciones para gestionar eventos de hardware de manera eficiente.



ACCESO DIRECTO AL HARDWARE

Programación para interactuar directamente con el hardware sin intervención del sistema operativo.



EJEMPLOS DE DISPOSITIVOS (PUERTOS I/O, TECLADO, PANTALLA, TEMPORIZADORES, ETC.)

Los dispositivos son hardware controlado por ensamblador, como puertos I/O, teclados y pantallas.

PUERTOS I/O

Dispositivos que permiten la entrada y salida de datos entre el procesador y el mundo exterior.

TECLADO

Dispositivo de entrada utilizado para capturar caracteres y comandos.



PANTALLA

Dispositivo de salida que permite visualizar datos y texto generados por el sistema.



TEMPORIZADORES

Dispositivos que generan señales en intervalos específicos para controlar la sincronización del sistema.



CONCEPTOS DE DIRECCIONAMIENTO, INTERRUPCIONES Y COMUNICACIÓN HARDWARE-SOFTWARE

Estos conceptos son clave para entender la comunicación eficiente entre sistemas operativos y hardware.

DIRECCIONAMIENTO DE MEMORIA

Método para acceder a las ubicaciones de memoria específicas de los dispositivos a través de direcciones

INTERRUPCIONES

Mecanismo para gestionar eventos de hardware que requieren atención inmediata.



COMUNICACIÓN HARDWARE-SOFTWARE

Interacción entre el software (programas) y el hardware (dispositivos) para ejecutar tareas.



CONTROL DE FLUJO

Mecanismos para controlar la secuencia de ejecución, como los temporizadores e interrupciones.



EL BUFFER DE VIDEO EN MODO TEXTO

El buffer de video en modo texto almacena caracteres y atributos en memoria, permitiendo mostrar texto en la pantalla en lugar de imágenes complejas.

UBICACIÓN DEL BUFFER DE VIDEO EN MEMORIA (SEGMENTO 0XB800)

El buffer de video está ubicado en una dirección específica de la memoria para acceder directamente a los datos que se visualizan en la pantalla.

DIRECCIÓN DE MEMORIA

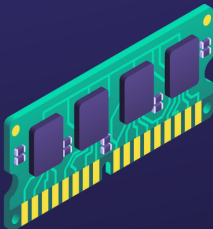
El buffer de video se encuentra en el segmento de memoria 0xB800 en arquitecturas como la x86.

ACCESO DIRECTO

Permite a los programas acceder directamente a la memoria para manipular el texto que se muestra en pantalla.

SEGMENTACIÓN DE MEMORIA

El segmento 0xB800 está dedicado exclusivamente a la visualización en modo texto.



INTERACCIÓN CON LA PANTALLA

Los datos en el buffer de video se traducen para ser mostrados en la pantalla.



FORMATO DEL CARÁCTER Y ATRIBUTO DE COLOR

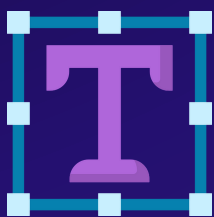
Cada carácter en el buffer de video se almacena con un código ASCII y un atributo de color para la visualización.

1234 CÓDIGO ASCII

Representación numérica del carácter que se mostrará en la pantalla.

TAMAÑO DE CARÁCTER

El formato del carácter generalmente sigue una estructura de 8 bits, determinando su forma.



PALETA ATRIBUTO DE COLOR

Un byte que define el color del texto y del fondo en pantalla.

COMBINACIÓN DE CARACTERES

El carácter y el atributo de color trabajan juntos para mostrar texto con el formato adecuado.

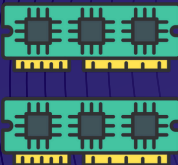
CÓMO MANIPULAR DIRECTAMENTE EL CONTENIDO DEL BUFFER DESDE ENSAMBLADOR

En ensamblador, se puede modificar directamente el buffer para mostrar texto de manera dinámica.



INSTRUCCIONES DE MEMORIA

Utilización de instrucciones de ensamblador para acceder y modificar el buffer de video.



CONTROL DE PANTALLA

Cambios inmediatos en la pantalla al escribir en el buffer de video.



MODIFICACIÓN DE CARACTERES

Escribir en las ubicaciones específicas de memoria para alterar el texto que se muestra.



ATRIBUTOS DE CARÁCTER

Controlar el color y el formato del texto mediante el ajuste de los atributos.



Elaborado por:

Dulce Ivon Colin Hernández, Juan Garduño Melquíades, Eduardo Pascual Vilchis, Jovani Emanuel Sánchez de Jesús.

Grupo: IC-0603

Asignatura: Lenguajes de Interfaz

EJEMPLOS DE RUTINAS SIMPLES PARA MOSTRAR TEXTO DIRECTAMENTE

Mostrar texto en pantalla utilizando rutinas simples en lenguaje ensamblador.

💡 RUTINA DE IMPRESIÓN

Una rutina básica para mostrar caracteres en la pantalla, usando el buffer de video.



🖥️ SALIDA DE TEXTO EN PANTALLA

Programación para que el texto se visualice correctamente en la pantalla desde el buffer.

📍 MANIPULACIÓN DE PUNTEROS

Cambio de la ubicación del puntero para escribir texto en diferentes posiciones.



✍️ VISUALIZACIÓN DE TEXTO ESTÁTICO

Rutinas que muestran líneas de texto predefinidas en la pantalla.

📀 ACCESO A DISCOS EN LENGUAJE ENSAMBLADOR

El acceso a discos en ensamblador interactúa directamente con los discos usando interrupciones para leer y escribir datos en sectores específicos.

ACCESO A DISCOS MEDIANTE INTERRUPCIONES

Se utiliza el mecanismo de interrupciones para interactuar con discos y realizar operaciones de lectura/escritura.

🚀 INTERRUPCIONES DE DISCO

Las interrupciones permiten realizar operaciones de I/O sin que el procesador quede ocupado continuamente.

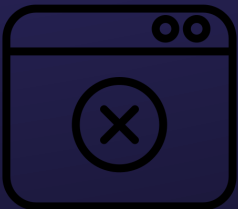


🔄 COMANDO DE LECTURA/ESCRITURA

Uso de interrupciones para leer o escribir datos en sectores del disco.

🔧 CONTROL DE DISPOSITIVOS DE ALMACENAMIENTO

Comandos específicos para controlar dispositivos de almacenamiento mediante interrupciones.



📶 ACCESO DIRECTO A SECTORES

Acceso directo a las ubicaciones físicas del disco sin intervención de un sistema operativo.

LECTURA Y ESCRITURA EN SECTORES

La manipulación de datos se realiza a nivel de sectores del disco, los cuales son unidades de almacenamiento básica.

SECTORES DE DISCO

Los discos están organizados en sectores, y cada sector tiene una capacidad definida de almacenamiento.

ACCESO DIRECTO A DATOS

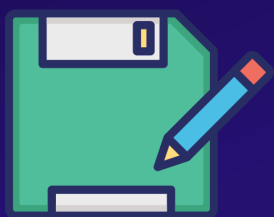
Acceder directamente a los sectores para mejorar el rendimiento de las operaciones.

OPERACIONES DE LECTURA/ESCRITURA

Programar operaciones para leer y escribir datos directamente en estos sectores.

CONTROL DE E/S

Gestionar las operaciones de lectura/escritura para asegurar la correcta manipulación de los datos.



ESTRUCTURA BÁSICA DE LOS SECTORES Y EL CONCEPTO DE CILINDRO, CABEZA, SECTOR

Los discos duros están organizados de manera estructurada para facilitar la localización y el acceso a datos.

CILINDRO, CABEZA, SECTOR

La geometría del disco se define por estos tres parámetros que indican la ubicación física de los datos.

ACCESO A SECTORES

Los datos se localizan en función de estos tres componentes dentro del disco.



SECTORIZACIÓN DEL DISCO

El disco está dividido en sectores que contienen datos organizados de manera eficiente.

GESTIÓN DE DATOS

Utilizar la geometría del disco para gestionar cómo los datos se almacenan y se accede a ellos.



Elaborado por:

Dulce Ivon Colin Hernández, Juan Garduño Melquíades, Eduardo Pascual Vilchis, Jovani Emanuel Sánchez de Jesús.

Grupo: IC-0603

Asignatura: Lenguajes de Interfaz

SEGURIDAD Y LIMITACIONES DEL ACCESO DIRECTO A DISCOS

El acceso directo a discos permite interactuar con los datos de manera eficiente, pero presenta riesgos de seguridad.

⚠️ RIESGOS DE ACCESO DIRECTO

El acceso directo sin protección puede llevar a la pérdida de datos o corrupción.

🚧 LIMITACIONES DEL ACCESO DIRECTO

Algunas restricciones pueden ser impuestas por el sistema operativo para evitar daños.

🔑 SEGURIDAD DEL ACCESO

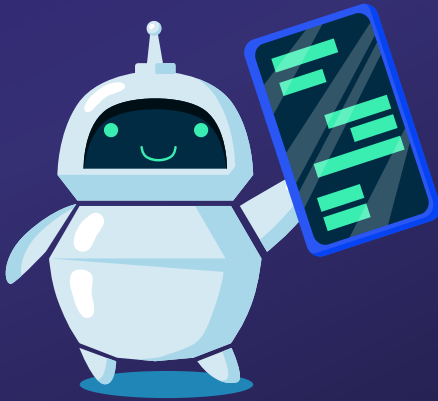
Se deben implementar mecanismos de control para prevenir el acceso no autorizado.

🛡️ PROTECCIÓN DE DATOS

Asegurar que los datos sean protegidos durante las operaciones de acceso directo a discos.



REFERENCES IEEE



[1] A. Silberschatz, P. B. Galvin, y G. Gagne, "Conceptos de sistemas operativos," 9ª ed., John Wiley & Sons, 2012.

[2] M. J. D. Hill, "Lenguaje ensamblador para procesadores x86," 7ª ed., Pearson, 2015.

[3] S. L. Sill, "Comprensión de la interfaz del sistema con ensamblador," en Actas de la Conferencia Internacional sobre Ciencias de la Computación e Ingeniería, 2017, pp. 100-107.

[4] R. J. Radovan, "Introducción a los sistemas de computación: Arquitectura y programación," McGraw-Hill Education, 2016.

[5] ISO/IEC 27001, "Sistemas de gestión de la seguridad de la información — Requisitos," Organización Internacional de Normalización, 2013. Disponible en: <https://www.iso.org/isoiec-27001-information-security.html>. [Accedido: 4-Jun-2025].