

Arquitectura básica de un sistema computador

Departamento de Ingeniería de Computadores

2025-2026

- 1 Bloques funcionales básicos de un computador de propósito general
 - Jerarquía de niveles estructurales
 - Arquitectura básica de un computador

- 2 Conceptos básicos del Sistema Operativo

- 3 Métricas de rendimiento
 - Definición de métricas de rendimiento
 - Ley de Amdahl

Contenido

- 1 Bloques funcionales básicos de un computador de propósito general
 - Jerarquía de niveles estructurales
 - Arquitectura básica de un computador
- 2 Conceptos básicos del Sistema Operativo
- 3 Métricas de rendimiento

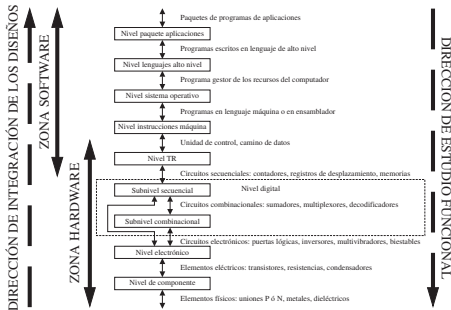
Jerarquía de niveles estructurales

Computador: Dispositivo electrónico formado por componentes y subsistemas digitales que permite el procesamiento de datos

Jerarquía de niveles estructurales

Computador: Dispositivo electrónico formado por componentes y subsistemas digitales que permite el procesamiento de datos

Jerarquía de niveles estructurales



- Nivel digital: Se corresponde con la máquina física (hardware)
- Nivel instrucciones máquina: El lenguaje ensamblador es la notación simbólica del lenguaje máquina
- Nivel sistema operativo: Capa software que rodea al hardware para facilitar su utilización
- Nivel lenguajes de alto nivel: Los programas escritos en un lenguaje de alto nivel necesitan un compilador para traducirlos a lenguaje máquina

Jerarquía de niveles estructurales

Lenguaje de alto nivel

Programa en alto nivel - C

```
B[0] = c[0] + c[1];
```



Compilador

Ensamblador - MIPS

```
lw $t1, 0($a0)
lw $t2, 4($a0)
add $t3, $t1, $t2
sw $t3, 0($a1)
```



Ensamblador

Lenguaje máquina -
MIPS

```
1000 1100 1000 1001 0000 0000 0000 0000
1000 1100 1000 1010 0000 0000 0000 0100
0000 0001 0010 1010 0101 1000 0010 0000
1010 1100 1010 1011 0000 0000 0000 0000
```

Arquitectura Von Neumann

Computador de programa almacenado (*stored-program computer*)

Almacena las instrucciones en memoria (el tratamiento de los programas y los datos en memoria es intercambiable)

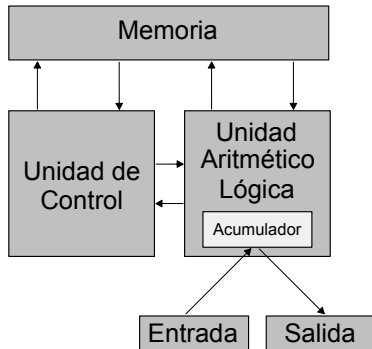


Figura: Diseño básico de un computador Von Neumann.

[fuente https://commons.wikimedia.org/wiki/File:Von_Neumann_architecture.svg GNU Free Docu. License]

Estructura básica de un computador

3 unidades funcionales básicas

CPU: Unidad de Control + ALU

Memoria

I/O: Subsistema de E/S

Estructura básica de un computador

3 unidades funcionales básicas

CPU: Unidad de Control + ALU

- Uno o múltiples microprocesadores
- Ejecuta instrucciones (Unidad de control) y lleva a cabo operaciones aritmético-lógicas (ALU)

Memoria

I/O: Subsistema de E/S

Estructura básica de un computador

3 unidades funcionales básicas

CPU: Unidad de Control + ALU

Memoria

- Distintos tipos y niveles de memoria en un computador.
Básicamente:
 - ▶ Memoria principal (RAM)
 - ▶ Memoria(s) caché(s)

I/O: Subsistema de E/S

Estructura básica de un computador

3 unidades funcionales básicas

CPU: Unidad de Control + ALU

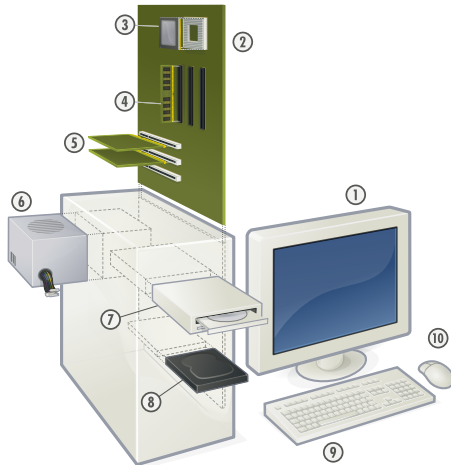
Memoria

I/O: Subsistema de E/S

- Interconexión de componentes internos y externos del computador
- Dispositivos externos de entrada y de salida

Un PC típico

Algunos componentes de un PC de sobremesa tradicional



- ① Monitor o Pantalla
- ② Placa base (*Motherboard* o *Mainboard*)
- ③ CPU
- ④ RAM
- ⑤ Tarjetas de expansión
- ⑥ Fuente de alimentación
- ⑦ Disp. de almacenamiento óptico (DVD, CD)
- ⑧ Disco duro
- ⑨ Teclado
- ⑩ Ratón

Figura: Imagen descriptiva de las partes de un PC

[fuente https://commons.wikimedia.org/wiki/File:Personal_computer,_exploded_5.svg GNU Free Doc. License]

Contenido

- 1 Bloques funcionales básicos de un computador de propósito general
- 2 Conceptos básicos del Sistema Operativo
- 3 Métricas de rendimiento

Introducción

Necesidad de un sistema operativo

- El ordenador sin sistema operativo solo podría ejecutar código máquina.
- Queremos algo más:
 - ▶ Ejecutar procesos simultáneamente, compartiendo recursos.
 - ▶ Olvidarnos del detalle de cada componenten hardware (ratón, tarjeta gráfica,...).
 - ▶ Tener un entorno de trabajo para lanzar programas, manejar ficheros, ...
 - ▶ Escribir programas en lenguajes de alto nivel.
 - ▶ Tener múltiples usuarios utilizando el sistema, incluso de forma concurrente.
 - ▶ Etc.

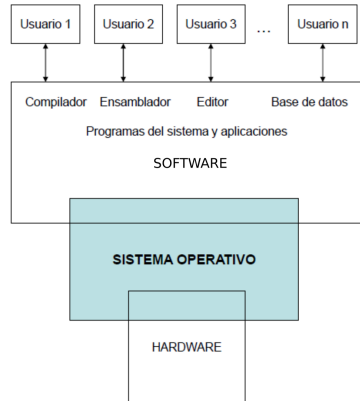
Introducción

Definición

Sistema operativo

- Es el software de base que gestiona los recursos del computador (CPU, memoria, almacenamiento, etc.) y le proporciona al usuario un entorno de trabajo:
 - ▶ Servicios para gestión de procesos, de E/S, de gestión de almacenamiento, protección y seguridad, ...
 - ▶ Asigna recursos y los recupera cuando dejan de ser utilizados.
- Actúa como interfaz entre el hardware y el usuario y las aplicaciones que ejecuta.

Introducción



Sistema Operativo

Ejemplos actuales

- **Familia Unix:** partieron de una raíz común y siguieron caminos separados.
 - ▶ Linux (Debian, Ubuntu, Suse, Red Hat, ...).
 - ▶ Solaris/SunOs.
 - ▶ FreeBSD, ...
- **Windows (Microsoft):**
 - ▶ DOS + Windows 3.1x.
 - ▶ Windows 95, 98, Me, 2000, Xp, Vista, 7, ..., 10.
 - ▶ NT, 2000server, 2003server, 2008, ...
- **Mac OS, iOS (Apple Inc.).** Están basados en Unix.
- **Android (Google).** Basado en kernel de Linux.

Contenido

- 1 Bloques funcionales básicos de un computador de propósito general
- 2 Conceptos básicos del Sistema Operativo
- 3 Métricas de rendimiento
 - Definición de métricas de rendimiento
 - Ley de Amdahl

Definición de métricas de rendimiento

- *Tiempo de respuesta* (latencia o tiempo de ejecución). Tiempo entre el inicio y el final de una tarea. Para maximizar el rendimiento se minimiza el tiempo de ejecución de una tarea:

$$R_x = \frac{1}{\text{tiempo de ejecucion}_x}$$

- *Productividad*. Cantidad de trabajo hecho en un cierto tiempo
 - ▶ **Teléfonos inteligentes** (*smartphones*): tamaño, latencia, vida de la batería, coste, ...
 - ▶ **Servidores**: productividad, latencia, consumo energético, calidad del servicio, fiabilidad, ...
 - ▶ **Portátiles**: tamaño, latencia, vida de la batería, coste
 - ▶ **PCs**: latencia, coste, ...

Diferente métrica implica diferente diseño

Definición de métricas de rendimiento

Tiempo de respuesta. Incluye los accesos a disco, accesos a memoria, actividades de E/S, y la carga adicional del S.O.

Tiempo de ejecución de la CPU. Tiempo que la CPU dedica a ejecutar una tarea concreta

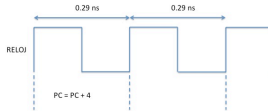
$$T_{cpu} = T_{user} + T_{s.o.}$$

Ejemplo

```
>time ./programa datos
real 0m11.588s -> tiempo total
user 0m3.735s -> tiempo de CPU del usuario
sys 0m3.286s -> tiempo de CPU del S.O.
```

Definición de métricas de rendimiento

Ciclo de reloj. Los procesadores tienen un reloj que funciona a una frecuencia concreta y determina el momento en que tienen lugar los eventos. Ejemplo: **Intel[®] Core[™] i7-990X processor Extreme Edition**: 3.46 GHz core speed



Tiempo de ejecución de CPU.

$$\text{Tiempo de CPU} = \text{ciclos de CPU} \times \text{tiempo de ciclo} = \frac{\text{ciclos de CPU}}{\text{frecuencia de reloj}}$$

El número de ciclos de reloj requerido por un programa es:

$$\text{ciclos de CPU} = \text{instrucciones} \times \text{media de ciclos por instrucción}$$

Definición de métricas de rendimiento

Ciclo de reloj por instrucción (CPI) Es el número medio de ciclos de reloj que una instrucción necesita para ejecutarse. Instrucciones diferentes pueden necesitar diferente cantidad de ciclos de reloj. El CPI proporciona una media:

$$CPI = \frac{\sum_{i=1}^m CPI_i NI_i}{N}$$

donde NI_i es el número total de instrucciones tipo i , CPI_i es el número de ciclos de reloj para esa clase de instrucciones, N es el número total de instrucciones, y m es el número total de tipos de instrucciones existentes.

Definición de métricas de rendimiento

Tiempo de CPU

$$T_{cpu} = N \times CPI \times T_{ciclo} = \frac{N \times CPI}{f_{CPU}}$$

donde T_{ciclo} es el tiempo de ciclo de reloj y f_{CPU} la frecuencia de reloj.

Esta fórmula se puede utilizar para comparar dos realizaciones diferentes o para evaluar un diseño alternativo si se conoce el impacto de los tres parámetros

Definición de métricas de rendimiento

GFLOPS: miles de millones de operaciones en punto flotante por segundo

$$GFLOPS = \frac{FLOPS}{T_{cpu} \times 10^9}$$

TFLOPS:

$$TFLOPS = \frac{FLOPS}{T_{cpu} \times 10^{12}}$$

El problema es su dependencia del programa

Comparación de rendimiento

Comparación de una máquina X con una máquina Y

$$R_x > R_y$$

$$\frac{1}{T_{exec_x}} > \frac{1}{T_{exec_y}}$$

$$T_{exec_x} < T_{exec_y}$$

Relación cuantitativa entre dos máquinas diferentes

$$n = \frac{R_x}{R_y}$$

la máquina X es n veces más rápida que la máquina Y

Aceleración

Aceleración (*Speedup*) Medida del tiempo logrado tras incorporar una mejora al sistema

$$A = \frac{R_{despues}}{R_{antes}} = \frac{T_{antes}}{T_{despues}}$$

Es fundamental acotar la aceleración potencial a obtener con una posible mejora (que puede ser muy costosa)

Ley de Amdahl

Ley de Amdahl. La mejora obtenida en el rendimiento al utilizar una parte optimizada está limitada por la fracción de tiempo que se puede utilizar esa parte

$$T_{despues} = \frac{T_{afectado}}{A_m} + T_{no_afectado}$$

$$A = \frac{T_{antes}}{T_{despues}} = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$

donde F_m es la fracción de tiempo afectada por la mejora y A_m es el factor de mejora