

Tema 1. Introducción

Arquitectura de Computadores

Área de Arquitectura y Tecnología de Computadores
Departamento de Informática
Universidad de Oviedo

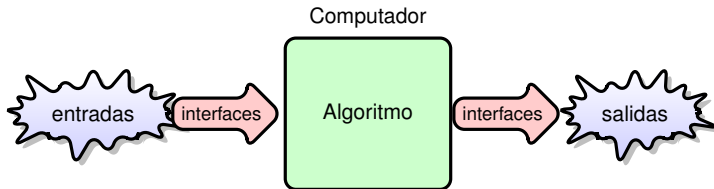
Curso 2020–2021

Computador

Definición

Máquina capaz de ejecutar algoritmos

- recibe entradas
- programa procesa la información
- genera salidas



Ejemplos

- ordenadores personales
- portátiles, tabletas
- teléfonos móviles

Diseño del computador

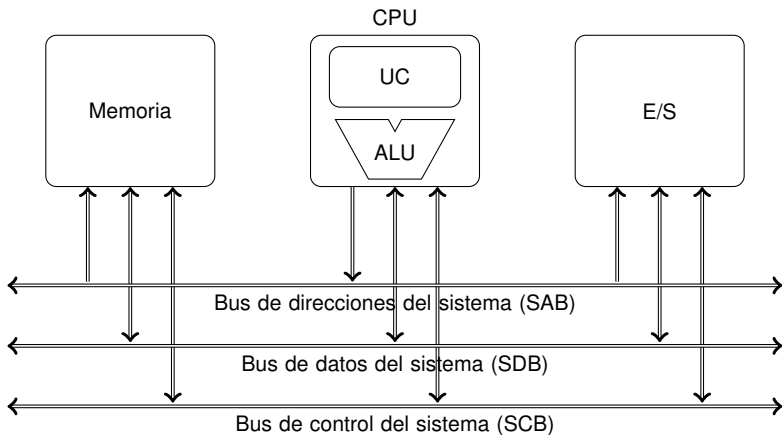
Necesidades

- almacenar (y eliminar) el programa a ejecutar
- almacenar los datos
- lectura y ejecución del programa
- mecanismo de entrada/salida

Soluciones: varios diseños

- memoria
- procesador (CPU)
- periféricos

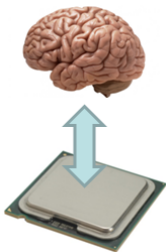
Arquitectura Von Neumann



Distancia conceptual

Problema

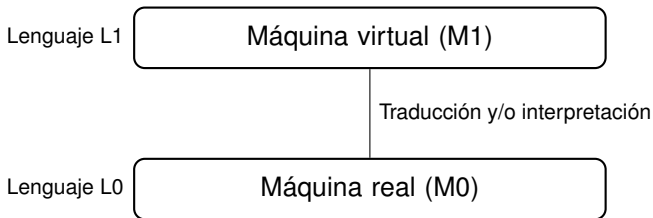
- interés del programador \Rightarrow lenguaje natural
- restricciones de la máquina \Rightarrow código máquina simple



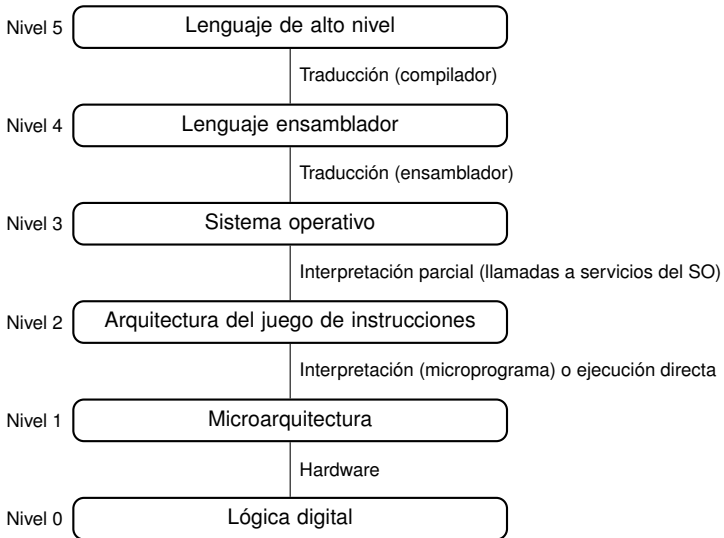
Máquina virtual

Solución

- definir nuevo lenguaje (L1)
- más cercano al lenguaje natural
- a partir del lenguaje máquina (L0)



Máquina multi-nivel



Principios básicos de la AC

- Diseño para el cambio rápido
- Uso de abstracciones
- Optimizar el caso más común
- Mejora del rendimiento mediante paralelismo
- Mejora del rendimiento mediante segmentación (*Pipeline*)
- Mejora del rendimiento por predicción
- Jerarquía de memoria
- Fiabilidad a través de redundancia

Arquitectura del juego de instrucciones

Visible al usuario programador

- tipos de datos
- registros
- juego de instrucciones
- formato de las direcciones

RISC (*Reduced Instruction Set Computer*)

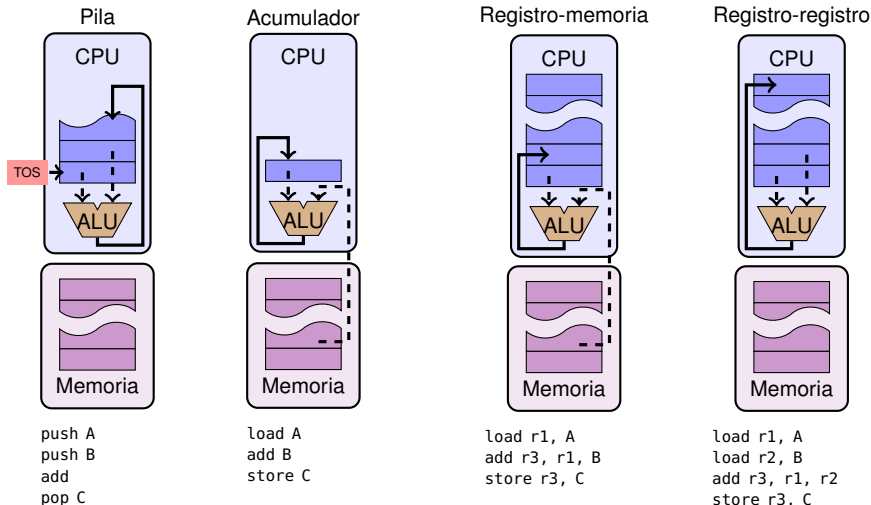
- instrucciones muy simples \Rightarrow rápida ejecución
- Ej.: PowerPC, IA-64, SPARC, MIPS, ARM

CISC (*Complex Instruction Set Computer*)

- instrucciones complejas \Rightarrow operaciones de alto nivel
- Ej.: x86, Motorola 68000

Arquitectura del juego de instrucciones

Tipos de ISA: modelos de máquina ($C = A + B$)



Microarquitectura

Implementación de una ISA

- unidades funcionales
- caché (tipo y niveles)
- ancho de buses

Organización interna de la CPU

- segmentadas
- superescalares
- multinúcleo
- VLIW (*Very Long Instruction Word*)
- vectoriales

Ejemplos de implementaciones de x86

- Intel Coffee Lake (Core i5-8400), AMD Raven Ridge (Ryzen 3 2200G)

Concepto de rendimiento

Quieres comprar un computador

¿Cuál de estos dos equipos comprar?

- **Apple iPad 9,7**
 - Apple A10 Quad-core (2.34 GHz) 64 bits
 - 2 GiB LPDDR4 RAM
 - 64 GiB almacenamiento NAND
 - Pantalla 2048×1536 IPS
 - Precio: 330€
- **Samsung Galaxy Tab S4**
 - Octa-core (4x2.35 GHz Kryo & 4x1.9 GHz Kryo)
 - 4 GiB RAM
 - 64 GiB NAND
 - Pantalla 2560×1600 AMOLED
 - Precio: 649€

Concepto de rendimiento

Es necesario comparar computadores

Pueden utilizarse muchos criterios para comparar

- Precio
- Servicio técnico
- Reputación del fabricante
- Garantía
- Peso
- Aspecto *cool*
- Rendimiento

Muchas veces el rendimiento está relacionado con el coste

- \uparrow coste \Rightarrow \uparrow rendimiento
- No siempre se cumple esta relación

Concepto de rendimiento

¿Cómo se mide el rendimiento?

Ejemplo



Modelo	Plazas	Velocidad	Consumo	Coste (euros)
Ferrari F430	2	320 km/h	18.3 l/100 km	186 000
Audi R8	2	300 km/h	14.2 l/100 km	130 000
Renault Scenic	5	170 km/h	4.5 l/100 km	17 000
Citroën C4	7	180 km/h	10.7 l/100 km	19 000

- 1 ¿Cuál es el coche más rápido?
- 2 ¿Cuál es el que menos consume?
- 3 ¿Cuál es el que presenta el menor coste por pasajero y km?

Concepto de rendimiento

¿Y cómo se mide el rendimiento de un computador?

- El rendimiento de un computador es un concepto muy amplio
 - Tiempo en ejecutar una tarea
 - Número de tareas por unidad de tiempo
 - Consumo eléctrico
 - etc.

Necesario definir métricas de rendimiento

Métrica

Unidad de medida para cuantificar una característica particular de un sistema

- Una métrica de rendimiento cuantifica el rendimiento de un computador
- Se necesita un número para poder comparar

Métricas de rendimiento

Tiempo de respuesta (*response time*)

Tiempo invertido en completar una **tarea**

- Tiempo entre inicio y fin de un programa, respuesta a una consulta en una base de datos, etc.
- También conocido como *elapsed time* (tiempo transcurrido), o latencia (para memoria)
- Símil con la velocidad: \uparrow velocidad $\Rightarrow \downarrow$ tiempo de respuesta

Productividad (*throughput*)

Número de tareas completadas por unidad de tiempo

- Número de instrucciones ejecutadas por segundo, número de consultas servidas por una base de datos en un segundo, etc.
- En sistemas de memoria se mide como ancho de banda
- Medible de distintas formas

$$\text{Rendimiento} = \text{Productividad} = \frac{\text{Tareas completadas}}{\text{Tiempo referencia}}$$

Métricas de rendimiento

Tarea

Debe estar claramente definida

- Ejecutar un programa
- Ejecutar una instrucción
- Descargar una página web
- Una consulta a un base de datos

Métricas de rendimiento

Ejemplo

Tenemos un computador A que es capaz de completar 30 tareas en un segundo, mientras que otro computador B es capaz de completar 20 tareas en el mismo tiempo. ¿Cuál es la diferencia de rendimiento entre ambos computadores? Esto es, ¿cuánto más rápido es el computador A que el computador B?

$$\frac{\text{Rendimiento}_A}{\text{Rendimiento}_B} = \frac{\text{Productividad}_A}{\text{Productividad}_B} = \frac{30}{20} = 1.5$$

Computador A tiene una aceleración (*speedup*) de 1.5 sobre el computador B

Métricas de rendimiento

Ejemplo

Relación entre productividad (ρ) y tiempo de respuesta:

$$\text{T. respuesta}_A = \frac{1}{\rho_A} = \frac{1}{30} = 0.0\hat{3} \text{ s}$$

$$\text{T. respuesta}_B = \frac{1}{\rho_B} = \frac{1}{20} = 0.05 \text{ s}$$

Esta relación no siempre se cumple

- Solo cuando el computador ejecuta una única tarea de cada vez

Métricas de rendimiento

Tiempo de respuesta y productividad no son constantes

- Varían de unas mediciones a otras
- Deben aproximarse por variables aleatorias
- Caracterizadas por un valor medio y una desviación típica
- Son necesarias varias mediciones para obtener resultados **significativos**

Ejemplo

Se mide 10 veces (n) el tiempo de respuesta en la ejecución de un programa:

Medida	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
T. resp (s)	3.2	2.9	3.1	3.0	2.8	3.1	3.2	3.0	3.3	2.7

¿Cuál es el tiempo de respuesta del computador para ese programa?

Métricas de rendimiento

Ejemplo

- 1 Calcular el tiempo de respuesta medio (\bar{x})

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{30.3}{10} = 3.03 \text{ s}$$

- 2 Calcular la desviación típica (s)

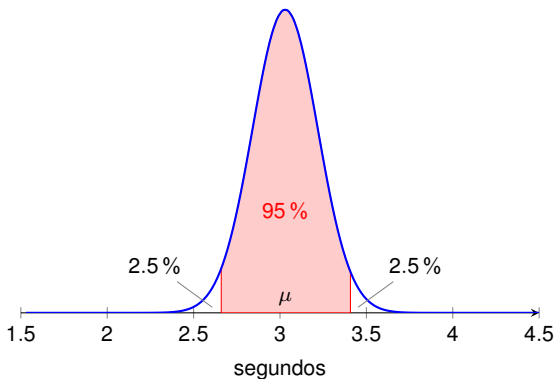
$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} = 0.19 \text{ s}$$

- 3 Se supone que la variable aleatoria sigue una distribución normal, luego usamos la media y desviación típica muestrales (que acabamos de calcular) como estimadores de la media (μ) y desviación típica (σ) poblacionales

Métricas de rendimiento

Intervalo de confianza

- Tiempo de respuesta entre $[\mu - 2\sigma, \mu + 2\sigma]$ con 95 % de probabilidad $\Rightarrow [2.62, 3.38]$



Mejoras de rendimiento

Se pueden plantear varios tipos de mejora

- Mejorar el tiempo de respuesta y la productividad
- Mejorar solo la productividad

Ejemplo: programas monohilo

Duplicar frecuencia de CPU

- Mejora tiempo de respuesta y productividad

Añadir una segunda CPU

- Dobra la productividad pero no se modifica el tiempo de respuesta

Mejoras de rendimiento

Técnicas para mejorar el rendimiento

- ① Reducir el tiempo de respuesta
 - Mejoras organizativas de los elementos funcionales
 - Mejoras tecnológicas
 - Se consigue al mismo tiempo incrementar la productividad
- ② Incrementar la productividad (más tareas en paralelo)
 - Aumentar el paralelismo
 - Normalmente no reduce el tiempo de respuesta. A veces incluso lo incrementa

Ley de Amdahl

- Computador formado por diferentes elementos
 - CPU, sistema de memoria, sistema de interconexión y periféricos
- Mejoras en el computador \Rightarrow mejorar alguno de sus componentes
- Una mejora de rendimiento en un componente en un factor p no supone una mejora en el rendimiento del computador por el mismo factor

Enunciado

La aceleración o ganancia obtenida (A) en el rendimiento de un sistema debido a la mejora de uno de sus componentes está limitada por la fracción de tiempo que se utiliza dicho componente

Ley de Amdahl

Tiempo de respuesta del computador se calcula sumando

- Fracción correspondiente a los componentes no mejorados
- Fracción correspondiente al componente mejorado

$$T. \text{ respuesta}_{\text{mejorado}} = T. \text{ respuesta}_{\text{original}} \times \left((1 - \text{Fracción}_{\text{mejorada}}) + \frac{\text{Fracción}_{\text{mejorada}}}{\text{Acelerac.}_{\text{mejorada}}} \right)$$

La fórmula de la ley de Amdahl se deduce

$$A = \frac{T. \text{ respuesta}_{\text{original}}}{T. \text{ respuesta}_{\text{mejorado}}} = \frac{1}{(1 - \text{Fracción}_{\text{mejorada}}) + \frac{\text{Fracción}_{\text{mejorada}}}{\text{Acelerac.}_{\text{mejorada}}}}$$

Ley de Amdahl

Ejemplo

Un computador tarda en ejecutar un programa 100 segundos: 20 debidos a la ejecución por parte de la CPU, 35 a los accesos a memoria, 40 a operaciones de entrada salida sobre el disco duro, y el resto a la espera por los diferentes mecanismos de interconexión dentro del computador

- ① ¿Cuál sería la aceleración en la ejecución del programa si se sustituye la CPU por otra el triple de rápida?

$$\text{Fracción}_{\text{CPU}} = \frac{20}{100} = 0.2 \quad A = \frac{1}{(1 - 0.2) + \frac{0.2}{3}} \approx 1.15$$

- ② ¿Y si se sustituye el disco duro por uno el doble de rápido?

$$\text{Fracción}_{\text{disco}} = \frac{40}{100} = 0.4 \quad A = \frac{1}{(1 - 0.4) + \frac{0.4}{2}} = 1.25$$

- ③ ¿Cuáles serían los nuevos tiempos de ejecución?

$$\text{T. respuesta}_{\text{mejorado}} = \frac{\text{T. respuesta}_{\text{original}}}{A}$$

$$\text{T. respuesta}_{\text{CPU mejorada}} = \frac{100}{1.15} = 86.96 \text{ s}$$

$$\text{T. respuesta}_{\text{Disco mejorado}} = \frac{100}{1.25} = 80 \text{ s}$$

Rendimiento CPU

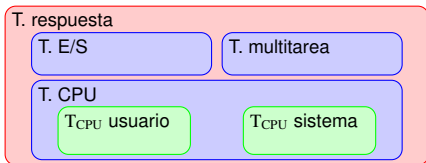
¿Cómo medir el rendimiento real de una CPU?

Se pueden utilizar las métricas vistas: tiempo de respuesta y productividad

- ✓ Fácil comprensión, intuitivas
- ✗ Comparación de CPU distintas complicada

Tiempo de respuesta

- Depende de muchos factores
- Muy variable en SS.OO. multitarea



¿Sirven los MIPS (Millones de Instrucciones Por Seg.)?

No en el caso de comparar CPU con juegos de instrucciones distintos

Necesarias métricas extrapolables

Tiempo de CPU

Tiempo de respuesta

- Tiempo que transcurre desde que comienza hasta que termina una tarea
- Incluye
 - Tiempo de CPU
 - Tiempo de entrada/salida
 - Ejecución de otras tareas
 - etc.

Tiempo de CPU

- ✓ Tiempo de usuario: Es el tiempo de CPU dedicado a la ejecución del código del programa
- Tiempo de sistema: Es el tiempo de CPU invertido en la ejecución del sistema operativo durante la ejecución de todo el programa

Tiempo de CPU

En los sistemas operativos UNIX existe un comando que nos muestra el tiempo de respuesta y el tiempo de CPU de un programa

```
$> time ./programa
```

```
real  0m2.822s
```

```
user  0m2.760s
```

```
sys   0m0.008s
```

real Tiempo transcurrido total \Rightarrow tiempo de respuesta

user Tiempo de CPU de usuario

sys Tiempo de CPU de sistema operativo

- Tiempo de CPU total:

$$2.76 + 0.008 = 2.768 \Rightarrow \frac{2.768}{2.822} \times 100 = 98.09\%$$

- Tarea intensiva en CPU

Tiempo de CPU

El tiempo de CPU de un programa (monohilo) se calcula a partir de la frecuencia de reloj

$$T_{\text{CPU}} = \frac{\text{Ciclos de CPU del programa}}{f} = \text{Ciclos de CPU del programa} \times T$$

CPI (Ciclos Por Instrucción)

$$\text{CPI} = \frac{\text{Ciclos de CPU del programa}}{\text{nº instrucciones del programa}}$$

Sustituyendo...

$$T_{\text{CPU}} = \frac{\text{Instrucciones del programa} \times \text{CPI}}{f} = \text{Inst. del programa} \times \text{CPI} \times T$$

Tiempo de CPU: *Iron Law*

$$T_{\text{CPU}} = \text{Instrucciones del programa} \times \text{CPI} \times T$$

Tiempo de CPU depende de:

- 1 El número de instrucciones por programa
 - Depende del nivel de abstracción del lenguaje de programación
 - Tecnología de compiladores
 - Optimización del código
- 2 El período de reloj
 - Indica lo rápido que trabaja la CPU
 - Depende de la tecnología de fabricación (7 nm, 5 nm)
 - Habitualmente, para subir la frecuencia es necesario mejorar tecnología de fabricación
 - ✗ Disipación de calor (y consumo)
 - También se puede conseguir con mejoras organizativas (*pipeline*)
- 3 Los ciclos por instrucción (CPI)...

Ciclos por Instrucción

El valor del CPI depende de la complejidad de las instrucciones

- CISC: complejidad de las instrucciones elevada
 - ✓ Programas compactos con número de instrucciones bajo
 - ✗ El CPI aumenta
- RISC: complejidad de las instrucciones baja
 - ✓ Se consigue un CPI bajo
 - ✗ Se necesitan más instrucciones en el programa

Tendencia actual

Seguir el principio enunciado por la ley de Amdahl al diseñar los juegos de instrucciones

- Instrucciones más utilizadas \Rightarrow CPI bajo
- Las menos utilizadas realizan operaciones complejas. Aunque tengan un CPI alto, su peso en el rendimiento final será menor
- CPU modernas siguen modelo híbrido RISC/CISC

Benchmarks

Rendimiento del computador no puede calcularse, debe medirse

- Resultado de las mediciones varía de una a otra
- Depende de las condiciones a que esté sometido el computador

Benchmark

Programa diseñado para evaluar el rendimiento del computador. También puede estar orientado a evaluar un único componente (CPU, memoria, disco duro, interfaz gráfica, etc.)

- Somete la parte a evaluar a una **carga de trabajo**
- Resultados varían de uno a otro benchmark

Carga de trabajo

Debe ser representativa del trabajo habitual del computador

- Equipo de oficina \Rightarrow tareas de ofimática
- Estación de trabajo \Rightarrow compilación, edición de vídeo, etc.
- Rendimiento medido varía en función de la carga

Benchmarks

Tipos de carga

- Carga real: se utilizan los programas reales que se utilizan en el trabajo habitual del computador
 - ✓ Rendimiento evaluado bajo condiciones reales de utilización
 - ✗ Son difíciles de reproducir
 - ✗ Gran variabilidad \Rightarrow elevado número de repeticiones para obtener resultados estadísticamente significativos
- Carga sintética: se utilizan pequeños programas que intentan reproducir las operaciones de una carga real
 - ✓ Condiciones fácilmente reproducibles
 - ✗ No son programas reales
 - ✗ A veces baja correlación con el rendimiento real observado

Benchmarks

Media aritmética vs. Media geométrica

	Productividad C1	Productividad C2
<i>Benchmark B1</i>	5	10
<i>Benchmark B2</i>	5	2

	$A_{C1/C2}$	$A_{C2/C1}$
<i>Benchmark B1</i>	0.5	2
<i>Benchmark B2</i>	2.5	0.4
Media aritmética B1 y B2	1.5	1.2
Media geométrica B1 y B2	1.12	0.89