

Introducción a los Sistemas Operativos y su relación con la Arquitectura de Computadores

Departamento de Automática



Universidad de Alcalá



/gso>

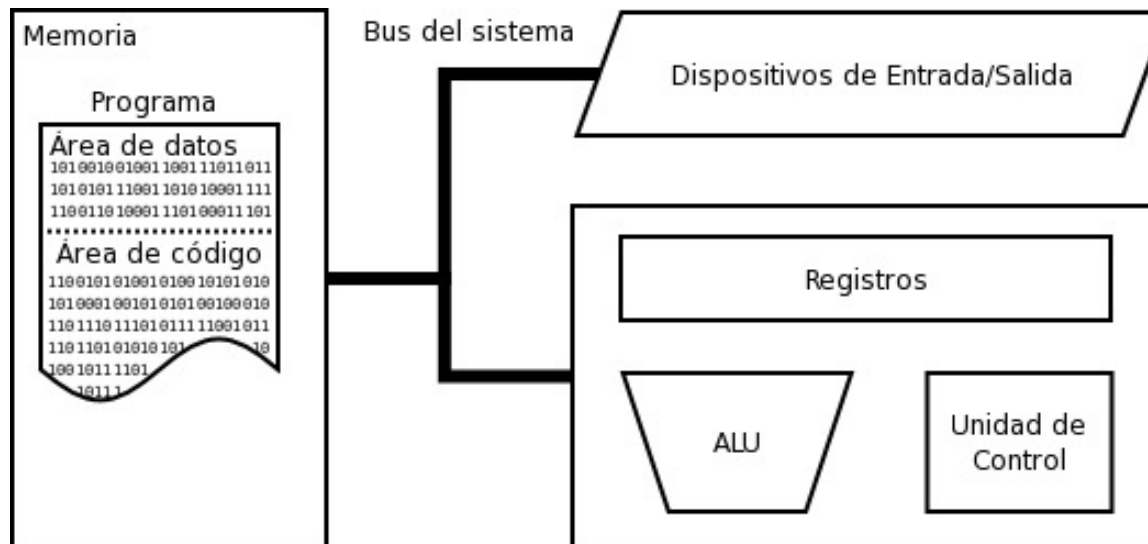
Índice

- Computador
- Sistema Operativo
- Arquitectura – Sistema operativo – Compiladores
- Arquitectura de von Neumann
- Modelo de programación de una CPU
- Arquitecturas actuales
- Evolución de los SSOO
 - Modelo máquina desnuda
 - Monitor simple residente
 - Sistemas de procesamiento por lotes (*batch*)
 - Multiprogramación
 - Sistemas de tiempo compartido
 - Sistemas de tiempo real
 - Sistemas distribuidos

Computador

- Máquina programable para el tratamiento de la información
- Arquitectura de von Neumann
 - 1949 - EDVAC - De propósito general con programa almacenado

↳ el primer computador programable fue EDVAC con memoria



✓ <https://www.youtube.com/watch?v=lsAeTXNQyIA>

Sistema Operativo

- Cada elemento *hardware* es diferente y su uso es complicado
 - ¿Tengo que aprender a programar cada uno de los dispositivos?
 - ¿Y si el fabricante cambia el modelo?
- Multiprogramación
Varios usuarios/programas usan el mismo *hardware*
 - Recursos limitados, ¿quién los repartirá?
 - ¿Y si dos o más participantes quieren el mismo recurso?
 - ¿Cómo se asegura la privacidad?
 - ¿Los demás pueden ver y alterar mis datos?

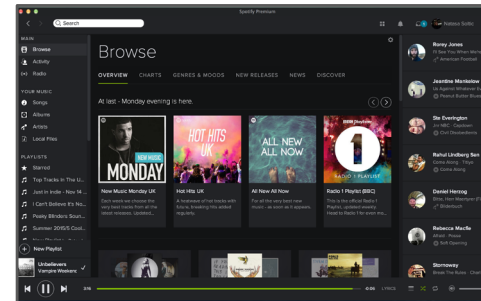
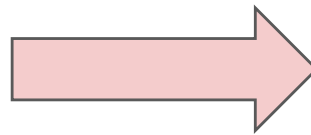
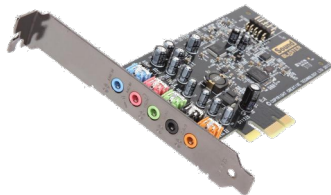
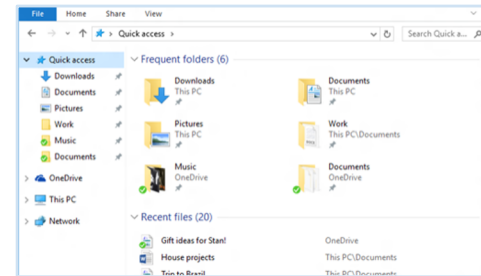
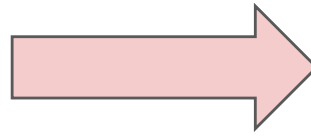
Sistema Operativo

```
mars@marsmain ~ $ pwd
/home/mars
mars@marsmain ~ $ cd /usr/portage/app-shells/bash
mars@marsmain /usr/portage/app-shells/bash $ ls -al
total 130
drwxr-xr-x  3 portage portage 1024 Jul 25 10:06 .
drwxr-xr-x 33 portage portage 1024 Aug  7 22:39 ..
-rw-r--r--  1 root  root    35808 Jul 25 10:06 ChangeLog
-rw-r--r--  1 root  root    27002 Jul 25 10:06 Manifest
-rw-r--r--  1 portage portage  4645 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r--  1 portage portage  5977 Mar 23 21:37 bash-3.2_p39.ebuild
-rw-r--r--  1 portage portage  6151 Apr  5 14:37 bash-3.2_p48-r1.ebuild
-rw-r--r--  1 portage portage  5988 Mar 23 21:37 bash-3.2_p48.ebuild
-rw-r--r--  1 portage portage  5643 Apr  5 14:37 bash-4.0_p10-r1.ebuild
-rw-r--r--  1 portage portage  6230 Apr  5 14:37 bash-4.0_p10.ebuild
-rw-r--r--  1 portage portage  5648 Apr 14 05:52 bash-4.0_p17-r1.ebuild
-rw-r--r--  1 portage portage  5532 Apr  8 10:21 bash-4.0_p17.ebuild
-rw-r--r--  1 portage portage  5660 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r--  1 root  root    5660 Jul 25 09:43 bash-4.0_p28.ebuild
drwxr-xr-x  2 portage portage  2048 May 30 03:35 files
-rw-r--r--  1 portage portage   468 Feb  9 04:35 metadata.xml
mars@marsmain /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
<use>
  <flag name='bashlogger'>Log ALL commands typed into bash; should ONLY be
  used in restricted environments such as honeypots</flag>
  <flag name='net'>Enable /dev/tcp/host/port redirection</flag>
  <flag name='plugins'>Add support for loading builtins at runtime via
  'enable'</flag>
</use>
</pkgmetadata>
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
```



Computador
Sistema Operativo
Arquitectura – Sistema operativo – Compiladores
Arquitectura de von Neumann
Modelo de programación de una CPU
Arquitecturas actuales
Evolución de los SSOO

Sistema Operativo

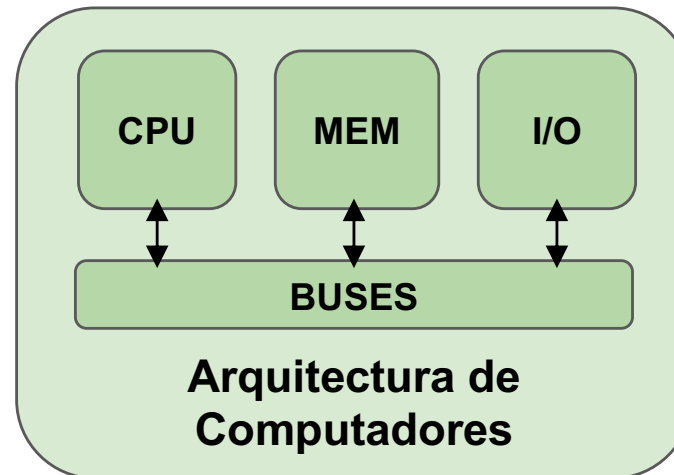


Sistema Operativo

Definición:

- S. Sánchez: conjunto de programas que, por medio de abstracciones, ponen el *hardware* del ordenador, de modo seguro, a disposición del usuario
- H. Deitel: programa que actúa como interfaz entre el usuario de un ordenador y el *hardware* del mismo, ofreciendo el entorno necesario para que el usuario pueda ejecutar programas
- H. Katzan: conjunto de programas y datos que ayudan a crear otros programas y a controlar su ejecución
- S. Madnik y J. Donovan: conjunto de programas que gestionan los recursos del sistema, optimizan su uso y resuelven conflictos

Arquitectura de computadores – Sistema operativo - Compiladores



El so se relaciona con el compilador.

Compiladores

CLang/LLVM

GCC

Visual Studio

Sistemas Operativos

macOS

Windows

Arquitectura de computadores – Sistema operativo – Compiladores

- **Lenguajes de alto nivel**
 - Python, C, Java
- **Lenguaje de bajo nivel o lenguaje máquina**
- **Sistemas de numeración posicionales**
 - Binario, base 2 $0010\ 0111_{(2)}\ 39_{(10)}$
 - bit, byte, **palabra** (relacionado con la arquitectura de un computador)
 - Octal, base 8 $047_{(8)}\ 39_{(10)}$
 - Hexadecimal, base 16 $0x27\ 39_{(10)}$
- **Representación:**
 - Big-endian (Motorola 6800): $0x56\ AB \rightarrow 0x56AB$
 - Little-endian (Intel x86): $0x56AB \rightarrow 0xAB56$

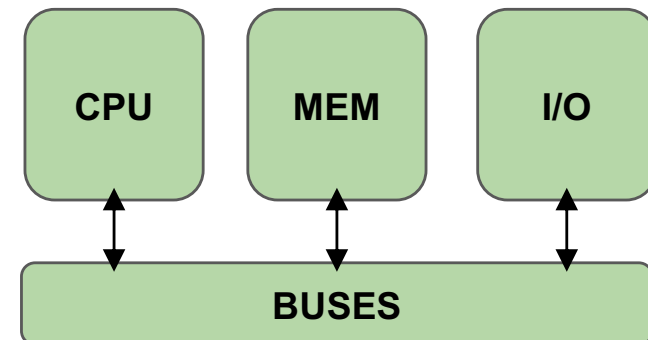
Arquitectura de von Neumann

Arquitectura

Se refiere a los atributos de un sistema que son visibles a un programador o, dicho de otra manera, aquellos que tienen un impacto directo en la ejecución de un programa

Elementos fundamentales

- CPU
- Memoria principal
- Entrada y salida
- Buses



Arquitectura de von Neumann

Unidad Central de Procesamiento (CPU)

- CPU = procesador = microprocesador (en un chip)
Circuito secuencial que interpreta y las ejecuta instrucciones de los programas. Consta de:
 - **ALU (Unidad Aritmético-Lógica):** realiza funciones aritméticas (suma, resta, etc.) y lógicas a nivel de bit (AND, OR, etc.) *la que se encarga de procesar y ejecutar las instrucciones*
 - **Unidad de Control (UC):** interpreta las instrucciones en lenguaje máquina y genera las señales de control para la ejecución de cada instrucción
 - **Registros**

Arquitectura de von Neumann

Memoria principal

- Almacena los procesos e información de un computador
- Cada dato ocupa una posición de memoria
- La capacidad y el tamaño del dato almacenado dependen del tipo de diseño

Arquitectura de von Neumann

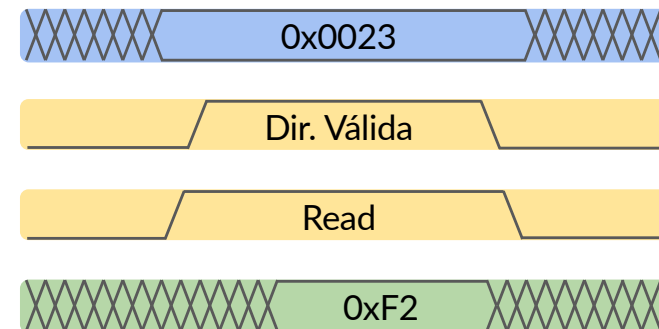
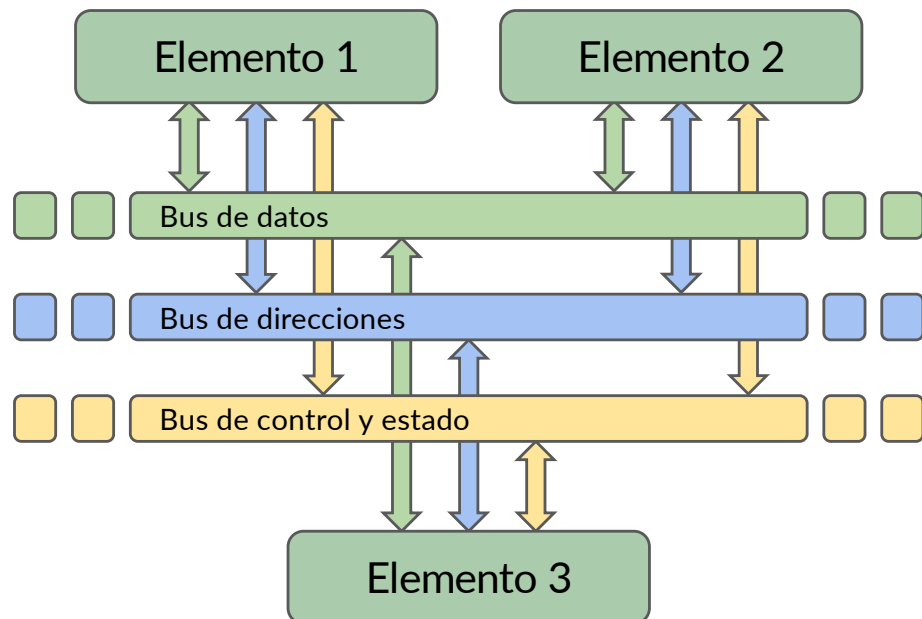
Entrada/salida

- Permite la comunicación con los dispositivos periféricos
- Carga en memoria la información (instrucciones y datos) que se procesan
- La naturaleza de cada dispositivo es totalmente diferente
- Todos deben comunicar con la CPU (buses)
- Es necesario establecer una arquitectura común: parte mecánica y controlador

Arquitectura de von Neumann

Buses

- Conectan entre sí los distintos componentes del sistema
- Son elementos muy complejos



Modelo de programación de una CPU

Define todas las funciones que puede realizar el procesador (CPU), determinado por el **juego de instrucciones**

- Decisiones de diseño
 - RISC (*Reduced-Instruction-Set Computing*): arquitectura en la que la mayoría de las instrucciones son relativamente simples. CPU de diseño más sencillo
Ej.: SPARC de Sun Microsystem's y el Alpha de Digital.
 - CISC (*Complex-Instruction-Set Computing*): arquitectura con una gran cantidad de instrucciones que procesando código complejo
Ej.: Intel

Modelo de programación de una CPU

- El juego de instrucciones define qué operaciones puede realizar la CPU
 - ¿Puede la CPU multiplicar matrices de números?
- Instrucción máquina
 - Secuencia de bits que representa una operación a realizar y dónde se encuentran los operandos
 - Código de operación, operando origen, operando destino
 - Ejemplo
 - Lenguaje ensamblador (mnemónicos): AND C, D
 - Lenguaje máquina: 0x58 02 03
 - Puede necesitar un número variable de operandos

Modelo de programación de una CPU

Juego de instrucciones

Aritméticas

ADD

sumar

SUB

restar

MUL

multiplicar

Lógicas bit-a-bit

AND

Y-lógica

OR

O-lógica

NOT

negación

Transferencia de datos

MOV

mover

PUSH

apilar

POP

extraer

Control/Salto

CALL

llamar a rutina

RET

retorno de rutina

JMP

salto incondicional

JNE

salto si no son iguales

Modelo de programación de una CPU

Almacenamiento

Dispositivos de almacenamiento.

- Registros de propósito general
- Registros de propósito específico
 - Puntero de pila - SP
 - Contador de programa - IP
- Registro de estado
 - Palabra de estado - PSW (condiciones)
Acarreo, desbordamiento, modo supervisor, etc.
- Otros registros: registros de coma flotante, registros transparentes, registros de datos de memoria, etc.
- Memoria
- (Otras memorias en un computador)

Modelo de programación de una CPU

Modos de direccionamiento

- Inmediato –operando en la propia instrucción

MOV B, 09

Cód.Op | Registro | Dato

Resultado: Registro B \leftarrow 0x09

- Directo o absoluto – operando en una dirección de memoria

MOV B, [09]

Cód.Op | Registro | Dir.mem.

Resultado: Registro B \leftarrow Contenido dir. mem (09)

- De registro –operando en un registro

MOV B, A

Cód.Op | Registro | Registro

Resultado: Registro B \leftarrow Registro A

Modelo de programación de una CPU

Modos de direccionamiento

- Indirecto mediante registro - operando en la dir. de memoria indicada por el registro

MOV B, [A]

Cód.Op | Registro | Dir.registro

Resultado: Registro B \leftarrow Contenido de dir.mem. indicada por A

- Relativo a un registro base

MOV B, [A+3]

Cód.Op | Registro | Dir.registro+offset

Resultado: Registro B \leftarrow Contenido de dir.mem. indicada por (A+offset)

- Otros: implícito, indirecto, absoluto, de pila, etc.

Modelo de programación de una CPU

Ejemplo:

- Arquitectura de 16 bits
- Memoria principal
 - 0x0000 – Programa
 - 0x0100 – Primeras potencias de 8
- Codificación binaria:
 - Registros generales
 A: 0x00, B: 0x01, C: 0x02
 - Códigos de operación:
 0x04 - MOV [Dirección registro], Registro
 0x06 - MOV Registro, Dato inmediato
 0x11 - ADD Registro, Registro

CPU Registers							
A	0000	B	0000	C	0003	D	02E0
IP	003A	SSP	0200	SR	s - - - - - M C Z F H		
		USP	0000		1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		

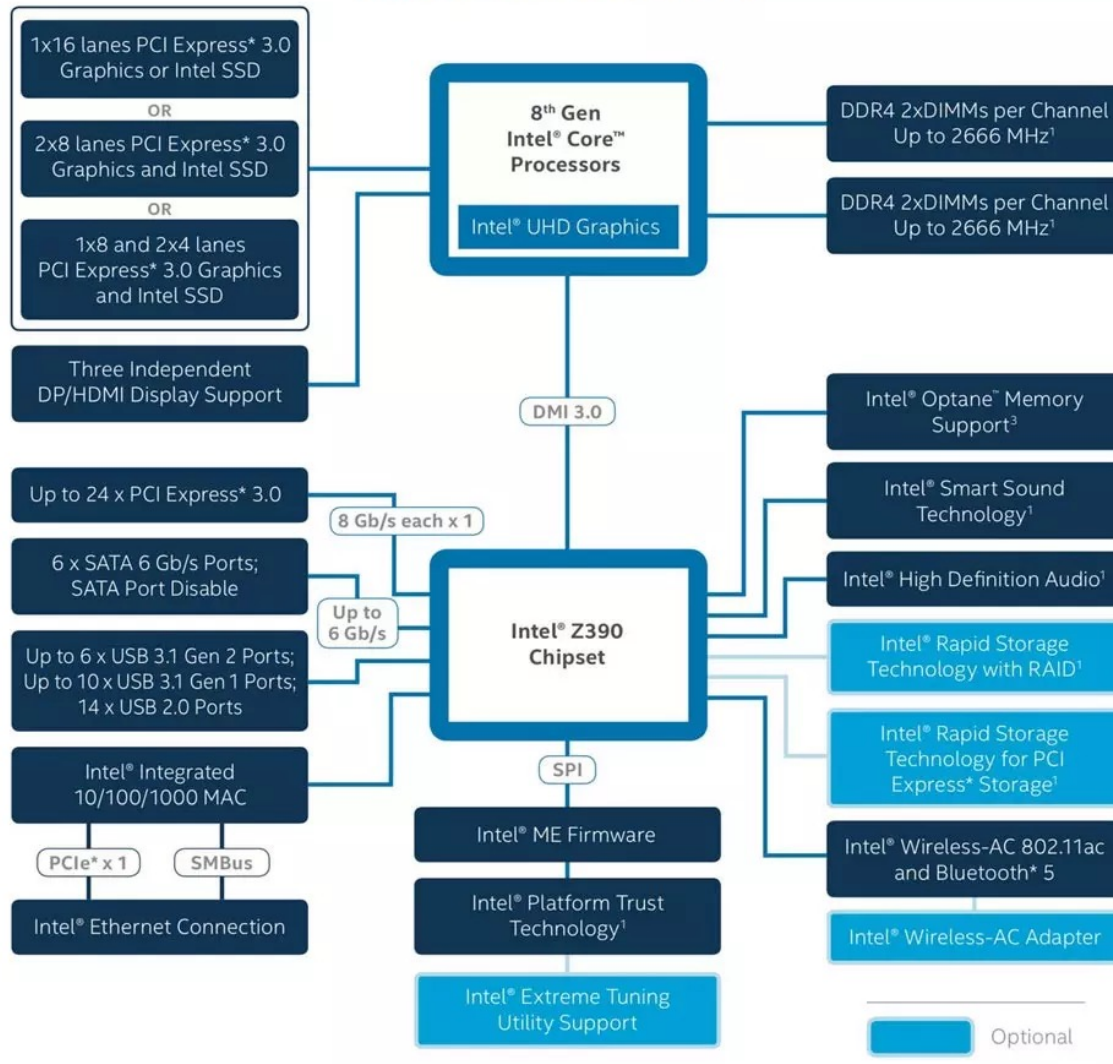
Memory																Split
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	2E	00	10	48	65	6C	6C	6F	20	57	6F	72	6C	64	21	00
001	06	04	02	00	06	02	00	03	06	03	02	E0	08	01	00	00
002	01	08	01	02	00	08	08	01	04	00	40	08	01	06	02	00
003	08	01	08	10	00	08	01	0A	80	00	06	00	00	03	06	01
004	00	04	06	02	01	50	11	00	01	04	06	02	00	00	00	00
005	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
010	00	01	00	08	00	40	02	00	10	00	80	00	00	00	00	00
011	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
012	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
013	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
014	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
015	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Modelo de programación de una CPU

- La arquitectura es (varias respuestas):
 - Big-endian
 - Little-endian
 - La palabra es de 16 bits
 - La palabra es de 8 bits
- Codifique las instrucciones y muestre el resultado en la arquitectura dada:
 - MOV A, 0x03
 - MOV B, 0x04
 - MOV C, 0x0150
 - ADD A, B
 - MOV [C+6], A

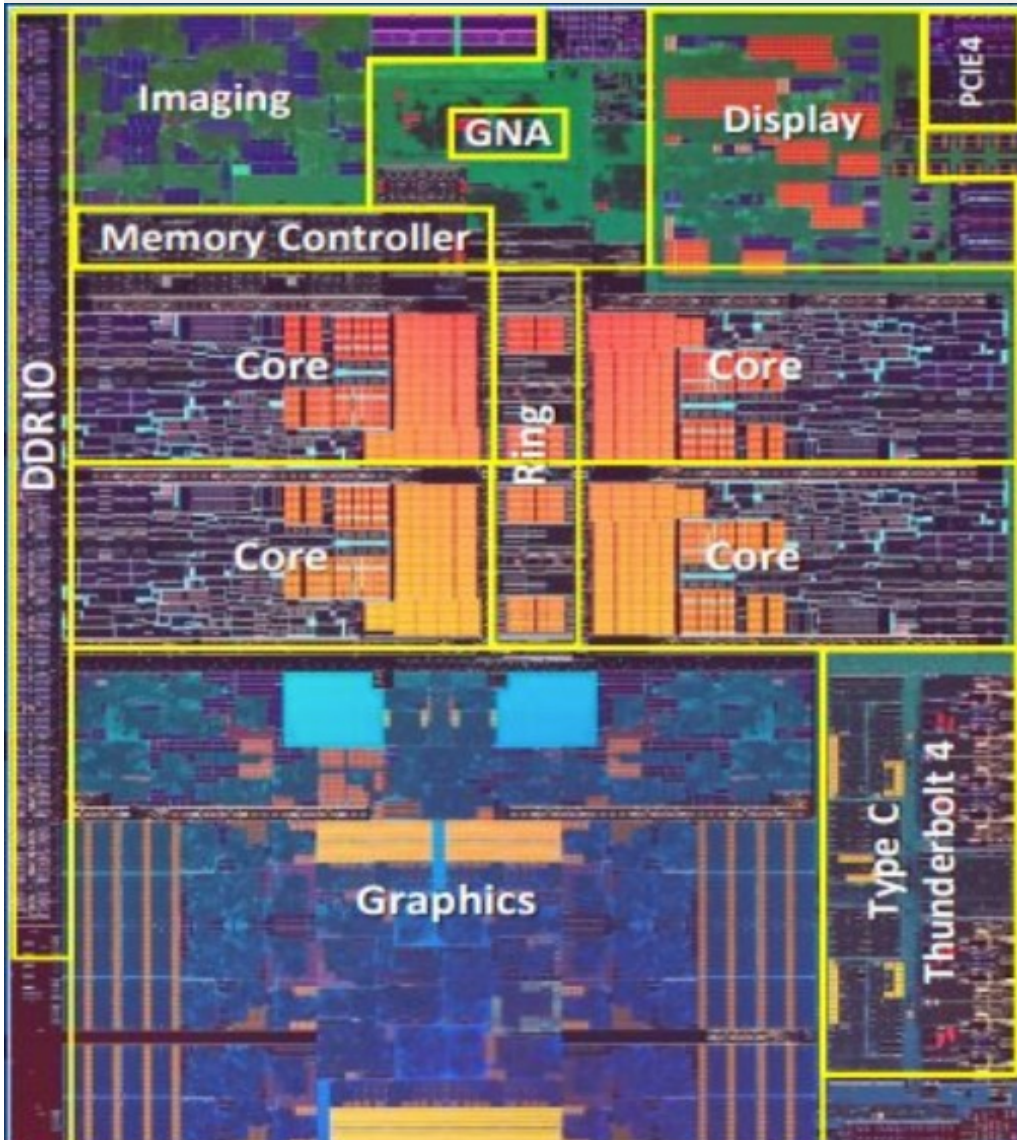
Arquitecturas actuales

INTEL® Z390 CHIPSET BLOCK DIAGRAM



<https://hardzone.es/2018/12/02/integrado-north-bridge-procesador>

Arquitecturas actuales



Intel 11^a Gen Tiger Lake

<https://www.xataka.com/componentes/intel-core-11a-generacion-caracteristicas-precio-ficha-tecnica>

¿Qué es el die del procesador?

<https://www.profesionalreview.com/2020/12/13/die-cpu/>

- <https://www.youtube.com/watch?v=fPQCfy6FNE8>

Modelo de máquina desnuda

Objetivo

Ejecutar programas almacenados en memoria

- Se programaba directamente sobre el *hardware*
- No existía nada similar al sistema operativo
- Los usuarios introducían código máquina mediante interruptores
- Problemas:
 - El usuario debía conocer todo el *hardware*
 - Cambios/fallos mínimos en el *hardware*

Monitor simple residente

Objetivo

Reutilizar código

- El código común se agrupa dentro de un monitor simple residente
 - El monitor es el primer germen de sistema operativo
 - Agrupa fundamentalmente rutinas de entrada/salida
 - Un monitor es, esencialmente, un manejador de dispositivo o *driver*
- El programador no tiene que programar directamente la E/S
- Se puede, por ejemplo, cambiar el lector de tarjetas sin cambiar los programas, basta con cambiar el monitor
- Problema: se pierde mucho tiempo entre la finalización de un proceso y el lanzamiento del siguiente

Sistemas de procesamiento por lotes (sistemas *batch*)

Objetivo

Reducir los tiempos de espera de E/S en la carga de programas

- Para la entrada/salida se utilizan ordenadores de bajo coste dedicados
- La ejecución de los programas se realiza por medio de ordenador de altas prestaciones y alto coste

Multiprogramación

Objetivo y requerimientos *hardware*

Objetivo

Solapar operaciones de E/S con la utilización de la CPU en la misma máquina

- Supone un salto de gran importancia y dificultad: tener más de un programa cargado en memoria a la vez
- Mientras se realiza una operación de E/S el microprocesador ejecuta otro programa de los que están en memoria
- Surgen nuevas necesidades *hardware*:
 - Interrupciones
 - Acceso directo a memoria (DMA)

Sistemas de tiempo compartido

Objetivo

Garantizar a los procesos el acceso equitativo a la CPU

- Protección contra monopolización de la CPU: interrupciones
- El tiempo de ejecución asignado a cada programa se divide en *quantum*
- Si un programa en ejecución agota su quantum, la CPU pasa a ejecutar otro programa (*round-robin*)
- Los SSOO de tiempo compartido son los habituales actualmente en la informática de consumo

Sistemas de tiempo real

Objetivo

Garantizar que los programas se ejecutan dentro de un plazo de tiempo acotado

- Para satisfacer restricciones de tiempo real no basta con tener un *hardware* más potente
- Tiempo real no es sinónimo de rápido
- Suelen encontrarse en sistemas empotrados




Sistemas distribuidos

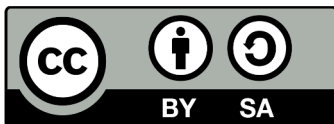
Objetivo

Ejecutar un sistema en distintos nodos interconectados por una red

- Características:
- Transparencia, rendimiento, capacidad de crecimiento, carácter abierto, fiabilidad
- Ejemplos:
 - Arquitectura cliente servidor
 - Cloud Computing – hw y sw ofrecido como servicio IaaS, PaaS, SaaS

Referencias bibliográficas

-  Sebastián Sánchez. *Sistemas Operativos*. Segunda edición. Universidad de Alcalá - Servicio de Publicaciones, 2005
-  A. S. Tanenbaum. *Sistemas Operativos Modernos*. Tercera edición. Prentice Hall, 2009
-  William Stallings. *Computer organization and architecture*. Décima edición. Pearson, 2015



© Pablo Parra, Óscar García, Elena Campo.
Departamento de Automática. Universidad de Alcalá.

Este documento se ha publicado con la licencia Creative Commons Attribution ShareAlike 4.0 (international):
<https://creativecommons.org/licenses/by-sa/4.0/>