

[simplyjaro.com](http://simplyjaro.com)

# FTEL

## Apuntes de clase

Apuntes y exámenes ETSIT UPM



Si alguna vez estos  
apuntes te sirvieron  
de ayuda, piensa que  
tus apuntes pueden  
ayudar a muchas  
otras personas.

Comparte tus apuntes  
en [simplyjarod.com](https://simplyjarod.com)

UCP = CPU

## ESTRUCTURA DE UN ORDENADOR:

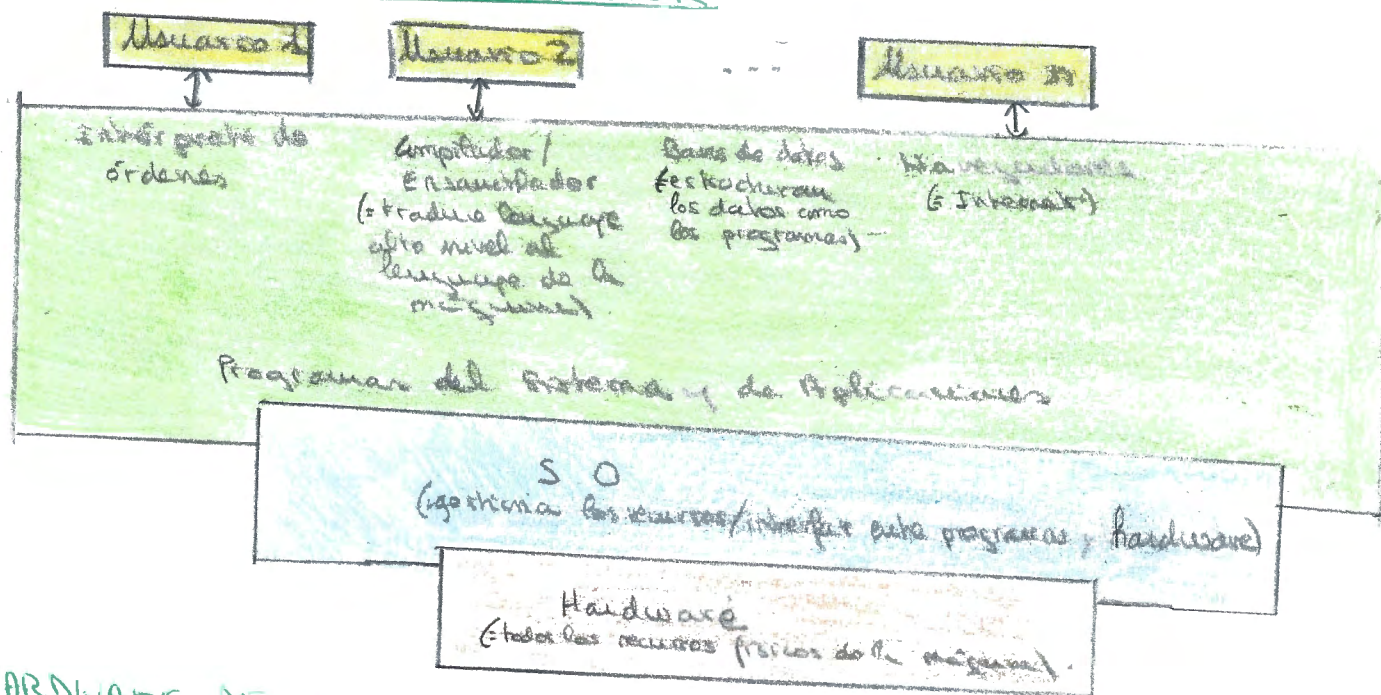
Hardware: CPU, Memoria, dispositivos E/S (todo lo q. se conecta al ordenador)

Programa de aplicación: Software → resolver pb o ofrecer servicios.

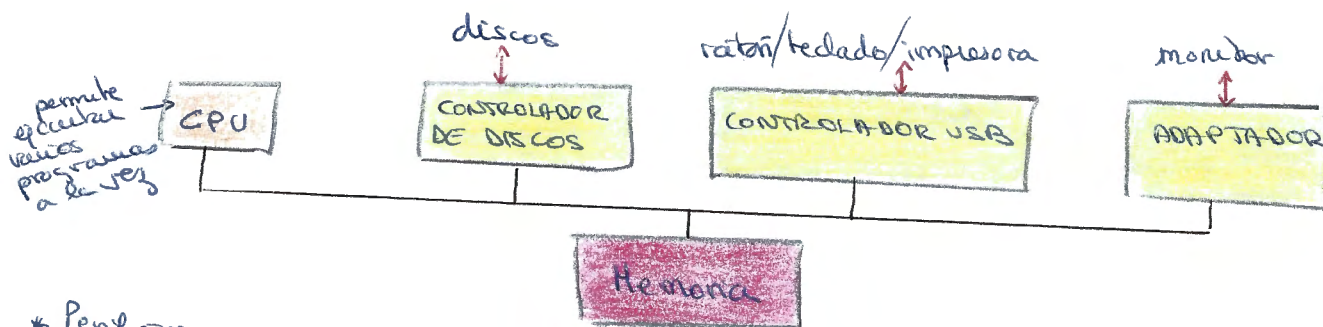
Sistema operativo (S.O.): Software → controla y coordina el uso del sistema repartiendo entre los programas de aplicac°.

Usuarios: personas, máquinas, otros ordenadores.

## COMPONENTES DE UN ORDENADOR



## HARDWARE DE LOS ORDENADORES



- \* Periféricos con características muy diferentes (velocidad, forma de funcionar, E/S, S/E...)
- se utilizan controladores específicos a cada periférico.
- \* Comunicac° entre periféricos a través de buses.
- \* Para ejecutar un periférico hay que almacenarlo en la memoria y este necesita electricidad → memoria elemento crítico

# Programas de sistema

= proporcionar un entorno adecuado para desarrollar y ejecutar programas.

- Gestión de ficheros y directorios
- Info. de Estado
- Modificac<sup>o</sup> de ficheros
- Procesadores de lenguajes
- Ejecución de programas
- Comunicaciones

Programas de aplicación (= navegadores web, hojas de cálculo...).

## FUNCIONAMIENTO DE 1 ORDENADOR: ARRANQUE

- 1) Un programa inicial ("bootstrap loader") se carga cuando se enciende un ordenador.
  - 2) Se suele almacenar en ROM ("firmware") → xq la memoria sólo funciona con electricidad  
→ usamos el ROM donde la info. no se borra.
  - 3) Inicia adecuadamente todos los elementos del sistema.
  - 4) carga el núcleo del S.O. en la memoria, éste comienza su ejec<sup>o</sup> y espera alguna interrupción:
    - Hardware (= lo producen los sistemas de hardware)
    - Software (= lo mandan los programas).
- \* Un programa no puede decir a la pantalla x ej. q. muestre algo, eso, lo hace el S.O. → los programas solicitan al S.O. de hacerlo.

## ALMACENAMIENTO

**Registros:** lo + cercanos a la CPU, almacena muy poca info. pero son muy rápidos  
(x ej. 1 instrucción: suma).

**Memoria principal (MP):** RAM (= memoria de acceso aleatorio) →

- Conjunto de palabras q. se identifican x una direcc. única.
- Acceso aleatorio o directo: tps de acceso es = si voy a direcc. 1 q. 20.
- Volátil: sin electricidad se pierde la info.
- Los programas deben estar en MP para poder ejecutarse.
- CPU accede directamente (CPU no lee en discos sino en la MP).
- Demasiado peque. para contener todos los programas → solo se encuentran en MP los programas q. se están ejecutando.

**Almacenamiento secundario:** Lenta pero mucha capacidad.

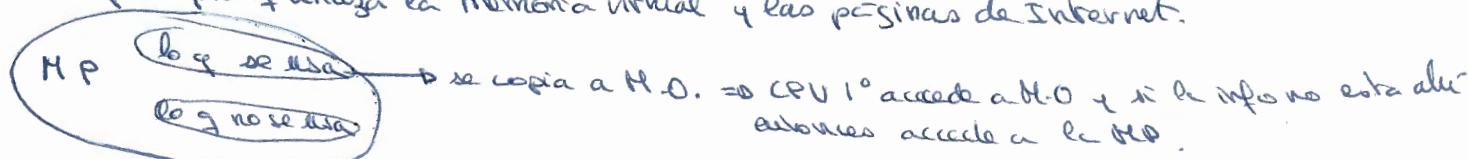
- Gran capacidad de almacenamiento no volátil
- Discos magnéticos (donde se puede grabar info.)
- La mayoría de los programas se almacenan en disco hasta q. se cargan en MP.

\* 1 Byte = 8 bits = 1 palabra.

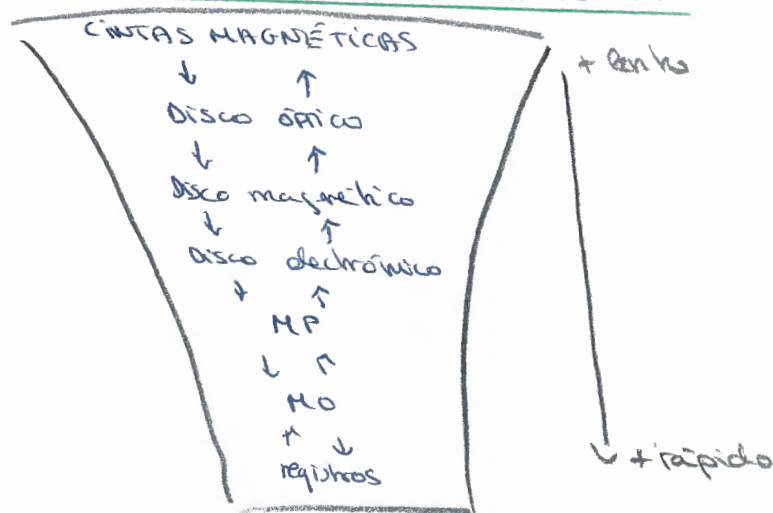
# MEMORIA OCULTA (CACHE)

2

- \* + peque q la MP pero + rápida
- \* Situada entre MP y la CPU
- \* Es volátil
- Memoria RAM de capacidad y tps de acceso pequeños.
- "La info. en uso" se copia de forma temporal de sistemas de almacenamiento lentos a otros más rápidos (M.O.)
- Al acceder, se comprueba 1º si la info. está en la M.O. :
  - Si está, se usa directam<sup>e</sup> (acceso)
  - Si no está (fallos), los datos se copian en la memoria caché y se usan.
- Principio de **localidad de las referencias**: las direc. de la MP a las q se accede durante un período de tps en el proceso suelen estar localizadas en zonas pequeñas de la MP
- \* Mismo principio q ubique la memoria virtual y las páginas de Internet.



## JERARQUÍA DE ALMACENAMIENTO



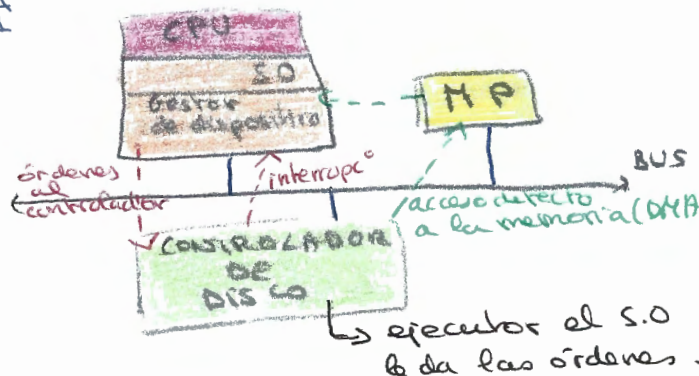
E/S

- Controlador de dispositivo
  - Trata en hipode dispositivo
  - Registros/buffers locales
  - Recibe órdenes del S.O
  - Interrumpe cuando termina (modelo más usado)

- Gestor de dispositivo
  - \* sin específicos al controlador
  - \* da las órdenes
  - componente del S.O q. interacciona con el ordenador

### Esquema habitual

- El usuario solicita una operac<sup>o</sup> de E/S mediante una llamada al sistema y queda esperando
- El S.O. usa el gestor para trasladar la orden al controlador
- El controlador transfiere los datos a la memoria o sin sin pocos a la CPU
- Cuando se completa, el ordenador interrumpe a la CPU.
- Se le pasan datos al usuario y se le despierta.



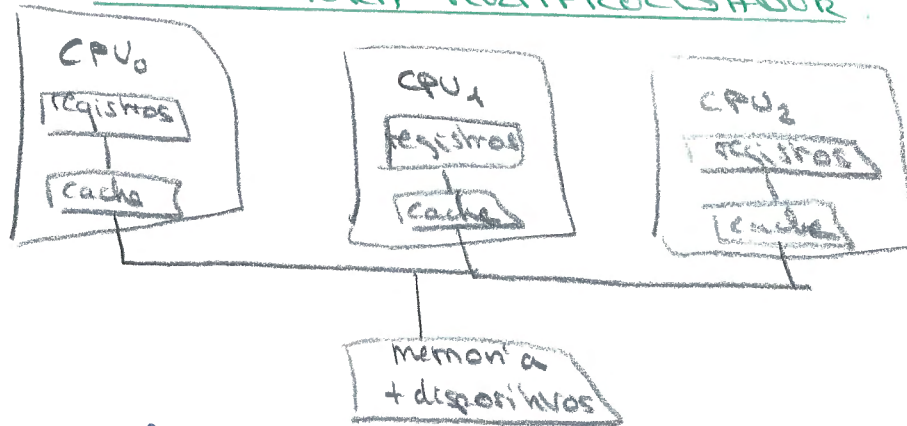
# ARQUITECTURA DE LOS ORDENADORES

- La mayoría de los sistemas usan un procesador de propósito general  
↳ tb. llevan procesadores específicos (ej. gráficos).

## • MULTIPROCESADOR:

- sistemas // o fuertem<sup>t</sup> acoplados.
- Ventajas: mejores prestaciones, mayor fiabilidad, economía de escala.

## ARQUITECTURA MULTIPROCESADOR



- Se complica el S.O. → pueden ejecutar varias actividades en //
- los datos E/S deben llegar al procesador adecuado.

## INTRO A LOS S.O

- Es 1 "programa/s" (**software**) que:
  - administra el hardware
  - proporciona un entorno para ejecutar los programas de aplicac<sup>o</sup>
  - interfaz entre usuarios y hardware de 1 ordenador
- Facilita el uso del ordenador
  - ↳ se adapta al tipo de aplicaciones a ejecutar y al uso
  - ↳ oculta la complejidad del hardware
- Utiliza el ordenador eficientem<sup>t</sup>
  - ↳ Gestión de recursos: procesos/memoria/E/S/ficheros...
  - ↳ Control de intentos de acceso simultáneos

## EL S.O DESDE EL PT DE VISTA DEL USUARIO

- PC Independiente: Fácil de usar + rendimiento para 1 solo usuario
- Mainframe con terminales: maximizar la utilizac<sup>o</sup> de recursos (CPU, memoria, E/S)... con reparto equitativo.
- Estac<sup>o</sup> de T y servidores conectados mediante redes
- Ordenadores de mano: Fácil de usar + rendimiento en base a sus limitaciones.

## EL S.O. DESDE EL PT DE VISTA DEL ORDENADOR

- Asigna y controla los recursos (CPU, memoria, E/S...)
- Mecanismos de gestión de recursos dependientes del tipo de sistema:
  - Interactivos → optimizar el tp. de respuesta, da prioridad a procesos con mucha E/S
  - lots → maximizar el uso del procesador, sin importar el tp. de respuesta.
  - Tp real → prioridad a procesos más urgentes.

## DEF. DE S.O. (No hay una def. común aceptada).

- S.O. es un gestor de recursos
  - gestiona todos los recursos del sistema
  - cuando hay peticiones en conflicto → decide quién usa el recurso
  - aplica criterios de equidad y eficiencia
- S.O. es un programa de control → controla la ejecuc<sup>o</sup> de los programas para prevenir errores o uso inadecuado del ordenador.
- Núcleo (**kernel**) → es el componente principal del S.O.

# COMPONENTES DEL S.O.

## GESTIÓN DE PROCESOS

(3)

Programa: código y datos estáticos

Proceso: programa de ejecuc<sup>o</sup> → necesita recursos: tp de CPU + memoria + ficheros + dispositivos E/S

Servicios proporcionados por el gestor de Procesos del S.O.

→ servicios básicos → creac<sup>o</sup>, destrucc<sup>o</sup> de procesos  
→ suspender y reanudar procesos

→ Mecanismos de interacc<sup>o</sup> entre procesos → sincronización y comunicación

Ejecución interactiva: cambio entre procesos muy rápido.

Si no caben todos los procesos en memoria se pueden llevar a disco.

\* la memoria virtual permite ejecutar programas que están parcialm<sup>t</sup> en disco.

### Operaciones del S.O.

→ Dirigido por interrupciones: 

|            |                                                                      |
|------------|----------------------------------------------------------------------|
| → Hardware | para cada tipo de interrupci <sup>o</sup> hay una rutina de servicio |
| → Software |                                                                      |

→ S.O. se basa en el sistema dual del procesador:

- modo usuario (muy restringido) y núcleo (= privilegiado o supervisor) = 1 bit de modo.

- El código de los usuarios se ejecuta en modo usuario.

- El núcleo en modo privilegiado.

### Transic<sup>o</sup> entre modos

→ Cuando se produce una interrupci<sup>o</sup>, el hardware cambia automáticamente a modo privilegiado.

→ El S.O. trata la interrupci<sup>o</sup>

→ Antes de dar control de nuevo al usuario, cambia a modo usuario.

### Interrupciones

Las interrupciones dirigen el comportamiento del S.O.

→ Generac<sup>o</sup> de las interrupciones: - dispositivos E/S para indicar fin de una operac<sup>o</sup> o señal a CPU  
- Software para q. S.O. realice una operac<sup>o</sup>.

→ Cuando interrupci<sup>o</sup> ⇒ se pasa a modo privilegiado / prioridad entre interrupciones / transfiere control a la rutina de servicio de dicha interrupci<sup>o</sup>.

→ Cuando se termina ⇒ se pasa a modo usuario / se continúa la ejecuc<sup>o</sup> del programa del usuario.

### Llamadas al sistema

Cada operación requiere 1 o varias llamadas a sistema.

### Gest<sup>o</sup> de memoria

S.O. se encarga de gestionar la MP.

decide cuáles son los contenidos de la memoria.

Gest<sup>o</sup> de Almacenamiento secundario → ficheros en discos duros, etc.

Lo Almacena lo q. no cabe en memoria en discos  
copias de seguridad en disco

### SUBSISTEMA DE E/S

S.O. oculta detalles y peculiaridades de los dispositivos de E/S

Responsabilidades { gestión memoria de E/S  
Interfaz entre con los gestores de dispositivos  
gestores para los ≠ dispositivos hardware

## PROTECC<sup>o</sup> Y SEGURIDAD

Protecc<sup>o</sup>: mecanismos para controlar el acceso de procesos o usuarios a recursos del sistema.

Seguridad: defensa del sistema contra ataques internos y externos.

→ Cada usuario tiene un identificador (ID)

Se asigna el ID a cada fichero y proceso del usuario para determinar las operaciones permitidas.

## SERVICIOS DEL S.O.

imp  
• interfaz de usuario / ejempl<sup>o</sup> de programas / operaciones E/S / gestión ficheros / comunicaciones /  
detección de errores.

oficial  
• gestión de recursos / contabilidad / protecc<sup>o</sup> y seguridad.

## TEMA 2 Representac<sup>o</sup> de la info.

①

### 1.1) Datos, info y conocimiento:

- Un agente es un sistema que percibe el entorno y puede actuar sobre él.
- datos: estímulos del entorno que el agente percibe, es decir, meras entradas al sistema.
- información: formada por datos con significado. El significado lo pone el agente receptor, que, aplicando sus mecanismos de razonamiento, los interpreta de acuerdo con sus conocimientos previos.
- conocimiento: informac<sup>o</sup> 1 vez asimilada x el agente en 1 forma sobre la que puede aplicar razonamientos.

→ Bit :

{

- unidad de medida de info.
- contradicción de "binary digit" ↙ 0  
↘ 1

→ Nivel de abstracción lógico: Todos los datos están codificados en binario, es decir, un dato siempre está formado por bits.

### 1.2) Caudal y volumen

- Caudal de un canal de comunicac<sup>o</sup>: es la tasa de bits ( $\equiv$  bitrate), medida en bits por segundo (bps).
- Para un caudal este, el volumen<sup>(v)</sup> transmitido en un tiempo (t) es:

$$V = \text{Bitrate}_{\text{bps}} \times (t)_s$$

→ El volumen de datos está relacionado con la capacidad de almacenamiento de algún sistema capaz de guardarlos ( $\equiv$  memoria). En estos sistemas la unidad básica de almacenamiento es un conjunto de 8 bits ( $\equiv$  octeto  $\equiv$  byte). Y esta es también la unidad para expresar la capacidad<sup>(C)</sup>.

$$(C) = \frac{V}{8}^{\text{bits}}$$

Los multiplicadores de la capacidad son potencias de 2 no de 10.

| Decimales |           | Binarios |           |
|-----------|-----------|----------|-----------|
| Valor     | prefijo   | Valor    | prefijo   |
| $10^3$    | kilo (K)  | $2^{10}$ | kibi (Ki) |
| $10^6$    | mega (M)  | $2^{20}$ | mebi (Mi) |
| $10^9$    | giga (G)  | $2^{30}$ | gibi (Gi) |
| $10^{12}$ | tera (T)  | $2^{40}$ | tebi (Ti) |
| $10^{15}$ | peta (P)  | $2^{50}$ | pebi (Pi) |
| $10^{18}$ | exa (E)   | $2^{60}$ | exbi (Ei) |
| $10^{21}$ | zetta (Z) | $2^{70}$ | zebi (Zi) |
| $10^{24}$ | yotta (Y) | $2^{80}$ | yobi (Yi) |

### 1.3) Soportes de transmisión y almacenamiento

- Un pt de memoria es un mecanismo capaz de almacenar 1 bit. Hay diferentes soportes para almacenar bits. Los soportes tienen características variadas: capacidad, posibilidad de escritura (grabado de 1 conjunto de bits) y/o de lectura (recuperación de un conjunto de bits), tiempo de acceso para la lectura o la escritura, coste por bit, densidad de integración (nº de bits por unidad de superficie) y volatilidad.
- biestable ( $\equiv$  flip-flop): se encuentra en 1 de 2 estados posibles y permanece en él mientras no se le aplique un estímulo para cambiar de estado. Por tanto sirve como pt de mem.  
(N, Z, C,  $\rightarrow$  indicadores biestables)
- Un procesador contiene registros  $\equiv$  conjunto de biestables. En cada registro puede almacenarse una palabra. La longitud de palabra es el nº de bits q. contienen los registros (1, 2, 4, 8 bytes dependiendo del procesador). Los registros son los soportes que tienen mayor coste x bit y menor densidad de integrac.  
$$C = \text{nº registros} \times \frac{\text{longitud palabra (bits)}}{8}$$
- Mem. de acceso aleatorio:  
Para poder acceder a un byte es necesario conocer su dirección que es un nº comprendido entre 0 y  $M-1$  siendo  $M$  la capacidad de la mem.  
Cuando el procesador tiene que escribir/leer un byte genera su direcc y la pone en 1 registro. Si el registro tiene  $n$  bits las direcciones que puede contener  $\in [0, 2^{n-1}]$ .
- "Acceso aleatorio" significa que el tps de acceso para la lectura o escritura es el mismo para todos los bytes, independientemente de la direcc. Estas mem se conocen como RAM.  
mem. no volátil pero solo de lectura (programas de arranque): ROM.
- De la gestº de la mem. secundaria se ocupa el S.O. mientras q la mem. terciaria (insertar/extraer DVD) requiere intervencº humana.  
En general mem. sec y ter. se diferencian sus soportes de los de la mem. principal en: no son volátiles, el acceso a los datos no es aleatorio, no se comunican directamº con el procesador: los datos se leen o escriben en la mem. principal en bloques de tamaño mínimo 512 bytes, mediante la técnica de acceso directo a memoria (DMA).
- El convenio de almacenamiento se basa en estructurar los datos en ficheros o archivos.

## 1.4) Extremistas mayores y menores

②

→ convenio general sobre el almacenamiento en las méms de acceso aleatorio:  
al bit de peso 0 le llamamos bit menos significativo (bms) y al de peso  $n-1$  bit + significativo (bMs)

→ Las direcciones tienen 6 dígitos hexadecimales y la mem 16 ~~direcc~~ MiB  
⇒ El nº total de direcciones es  $= 16^6 = 2^4 \times 2^{20}$

El convenio de que el byte que contiene los bits menos significativos (del 0 al 7) le corresponda la direcc. menor se le llama extremista menor y el opuesto, extremista mayor.

## 1.5) Codificaciones:

Primero ASCII, Estándar ISO/IEC 8859 (computers codificaciones con ASCII) y Unicode (⇒ UTF-8)

↳ compatible con ASCII  
↳ código de longitud variable. ej  $\begin{matrix} \in & 1B \\ & 2B \\ & 3B \end{matrix}$

## 1.6) Conversiones:

• Binario a hexadecimal → por sustitución

hexadecimal a decimal →  $0x4A2F \rightarrow 4 \times 16^3 + 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0$

• Decimal a hexadecimal → Divides por 16 y te quedas con el resto ↑

hexadecimal a binario → por sustitución

## 1.7) Representación de números enteros

Formato de coma fija (=coma situada inmediatamente a la dcha del bms).

$N$ :  $\max_{rep}(2^n - 1)$

$Z$ :  $bMs \equiv S$  (signo 0 > 0 y 1 < 0)  $\max_{rep}(2^{n-1} - 1)$

↓  
Signo y magnitud

Complemento a 1 (cambiamos 0 por 1 y 1 por 0)  $\min_{rep}(-2^{n-1} - 1)$   
↓  
dos reps para 0.

Complemento a 2 (sumamos 1 a C1)  $\min_{rep}(-2^{n-1})$

## 1.8) Operaciones

Para restas sumas el C2 e ignoramos el acarreo de los bits de signo.

✓ desbordamiento

Carraueo

N neg.

Z cero.

| op1 | op2 | Not | And | or | eor | nand | nor |
|-----|-----|-----|-----|----|-----|------|-----|
| 0   | 0   | 1   | 0   | 0  | 0   | 1    | 1   |
| 0   | 1   | 1   | 0   | 1  | 1   | 1    | 0   |
| 1   | 0   | 0   | 0   | 1  | 1   | 1    | 0   |
| 1   | 1   | 0   | 1   | 1  | 0   | 0    | 0   |

## Desplazamientos y rotaciones (no pueden estar representados en signo y magnitud)

A la derecha y A la izd.

• desplazamientos lógicos: introduce un cero y hay acarreo



• desplazamientos/circulares/rotaciones: se coge el bMS y se introduce. (el bMS se pierde)



• desplazamientos circulares con acarreo: se coge el bMS y se introduce (el bMS se acarrea)



• desplazamientos aritméticos ( $\times 2$ )



## Números reales:

Binario a hexadecimal y viceversa: sustitución

Hexadecimal a decimal:  $(8)_{16} = 12 \times 16^{-1} + 8 \times 16^{-2} = 0,78125$

Decimal a hexadecimal: multiplicas por 16 y te quedas con la parte entera.

Para su representación se puede utilizar un formato de coma fija con un convenio sobre la posición de la coma.

Resolución: "salto" o diferencia entre dos números representables.

Pero es mejor el formato de coma flotante.

## Formato de coma flotante:

S: signo    H: nº de dígitos    E: exponente     $N \in \mathbb{R}$     C: característica

$$\begin{aligned} 1. \pi &= 3,1415926 = 0x 3,243F6 \\ &= 0b \ 0011, 0010 \ 0100 \ 0011 \ 1111 \ 0110 \end{aligned}$$

$$\text{normalizaci}^\circ \text{ fraccionaria} = 0,11 \ 0010 \ 0100 \ 0011 \ 1111 \ 0110 \times 2^2$$

$$\text{normalizaci}^\circ \text{ fraccionaria omitiendo el 1}^\circ \text{ bit} = 1,1 \ 0010 \ 0100 \ 0011 \ 1111 \ 0110 \times 2$$

$$\text{normalizaci}^\circ \text{ entera} = 11 \ 0010 \ 0100 \ 0011 \ 1111 \ 0110 \times 2^{-20}$$

Permite representar nº racionales expresados en la forma:

$$N = \pm H \times 6^E \text{ (normalizaci}^\circ \text{ entera)}$$

$$N = \pm 0, H \times 6^E \text{ (normalizaci}^\circ \text{ fraccionaria)}$$

$$N = \pm 1, H \times 6^E \text{ (normalizaci}^\circ \text{ fraccionaria omitiendo el 1}^\circ \text{ bit)}$$

Convención para los n° 60, IE<sup>3</sup> 754

$$N = \pm 1, M \times b^E$$

precisión sencilla  $E=8$   
precisión doble  $E=11$

Negativos: rep. en signo y magnitud

M: normalización fraccionaria, omitiendo el bits

Base:  $b=2$

E con exceso de  $2^{E-1} - 1$ :  $E = C - 2^{E-1} + 1$

pasos a seguir: conversión a binario, normalización y calculamos C.

### Detección y corrección de errores:

Existen muchos algoritmos para la detección de errores, todos están basados en añadir redundancia al mensaje: bits de comprobación.

Algunos algoritmos permiten tb. corregir errores.

Algoritmo + sencillo: bit de paridad (se añade un bit para que el resultado sp sea par o impar).

Otro algoritmo: Suma de comprobación (la info se envía por bloques, entonces se envía tb el C2 de la suma. El receptor suma sus bloques con el C2 y el resultado tiene que dar 0).

### Tratamiento de los errores:

#### Peticion de repetición automática

- Parada y espera (el emisor espera a que el receptor le diga si lo ha recibido bien).
- Transmisión continua (hasta que el receptor notifica que no lo ha recibido bien el emisor continua enviando).

Corrección de errores en destino: se utiliza un código corrector, que añade la suficiente redundancia para reducir el n° de errores en el receptor.

ARQ híbrido: Combina los 2 anteriores.

Compresión: Transformación de una representación con N bytes a otra con  $N_c$  bytes ( $N_c < N$ ).

Factor de compresión (k) y porcentaje (C):  $k: 1 \equiv k = \frac{N}{N_c}$   
 $C = \frac{100}{k} \% = \frac{100 N_c}{N} \%$

Compresión sin pérdidas: de los datos comprimidos se pueden recuperar exactam<sup>t</sup> los originales.

Compresión con pérdidas: no es posible recuperar exactam<sup>t</sup> los originales.

## Algoritmos de compresión:

Se basan en la eliminac<sup>o</sup> de las redundancias innecesarias en la rep.

Es necesario un equilibrio entre: la distorsión aceptable y recursos disponibles para la codificac<sup>o</sup> y decodificac<sup>o</sup> (t<sub>ps</sub> real).

### Ej. algoritmos de compresión sin pérdidas:

- Codificac<sup>o</sup> de secuencias largas:  $1111 \equiv (4)$
- Codificac<sup>o</sup> con longitud variable (Huffman): Menos bits a símbolos + frecuentes.
- Codificac<sup>o</sup> por diccionario: Emisor y receptor crean un diccionario (el mismo).
- Deflate: combinac<sup>o</sup> de  $\uparrow$  y de Huffman.

### Ej. algoritmos de compresión con pérdidas:

Se aplica normalmente a señales analógicas digitalizadas: datos en binario resultante de muestreo y cuantificación.

Percepción: sonido:  $20\text{ Hz} - 20\,000\text{ Hz} \Rightarrow 300\text{ Hz} \text{ a } 3500\text{ Hz}$

imágenes: reducci<sup>o</sup> n<sup>o</sup> de colores

imágenes en mov:  $1/25^{\text{s}}$ . 30 fps

Muchas de las características de la percepción están relacionadas con las frecuencias de las distintas componentes de la señal  $\Rightarrow$  uso de transformac<sup>o</sup> matemáticas como la DCT (transformada discreta del coseno).

### Ej. de compresión de imágenes:

TIFF:  $\rightarrow$  Formato flexible que incluye una cabecera con etiquetas (tamaño, resoluc<sup>o</sup>, compres<sup>o</sup> si la hay...)

$\rightarrow$  Normalmente no compres<sup>o</sup>, o compres<sup>o</sup> sin pérdidas

$\rightarrow$  Uso para archivos de imágenes, o para aplicaciones como imágenes médicas.

JPEG:  $\rightarrow$  Método de compresión con pérdidas

$\rightarrow$  aplica algoritmos que combinan transformaciones matemáticas y particularidades de la vis<sup>o</sup> humana.

$\rightarrow$  Los parámetros de los algoritmos pueden ajustarse según la aplicación. Los herramientas que genera JPEG seleccionan un parámetro global  $Q$  (=calidad)  $\begin{cases} \text{100 max} \\ \text{0 min} \end{cases}$

## Tipos de ficheros

### Ficheros regulares (o "archivo"): bloque de datos $\Rightarrow$

- codificados en binario,
- almacenados en mem. secundaria con un nombre y disponible para su uso por algún programa, que lo conoce por su nombre.

### Ficheros especiales

$\rightarrow$  Dispositivos de E/S, Pseudodispositivos, Directorios, enlaces simbólicos, sockets...

Def fichero: es un conjunto de datos codificados en binario (compuestos o no), almacenados en un soporte no volátil con un nombre asociado, y disponible para los programas que pueden interpretar datos. (Ese conjunto de datos puede ser, en sí mismo, un programa).

Def archivo: es un fichero que contiene varios ficheros junto con datos sobre los ficheros ("metadatos") que permiten extraer los ficheros mediante un programa adecuado.

Metadatos: Datos sobre los datos del fichero:

- Tipo (especial de caracteres, de bloques...)
- Fechas (creac<sup>o</sup> y modificac<sup>o</sup>)
- Permisos
- Si es regular, tamaño...

La mayoría de los sistemas de fichero no tienen metadatos sobre el tipo de contenido de los ficheros regulares.

Contenidos de ficheros regulares:

→ Para el sistema de ficheros todos los ficheros regulares son iguales: un fichero es una sucesión de bytes.

→ Es el programa que usa al fichero el que entiende el significado de su contenido binario.

Los ficheros pueden ser

- ejecutables
- ficheros de texto
  - de documentos
  - multimedia
  - estructurados (diseñados para aplicaciones concretas)...

→ Cada uno tiene asociado un formato: convenios sobre la forma y el orden de almacenamiento.

Identificación del tipo de contenido de 1 fichero

→ por la extensión del nombre (caracteres tras el último pt)

→ por una cabecera que contenga metadatos:  
Número mágico.

→ "Avisando" con un mensaje 1a MIME

Metadatos externos: MIME (Multipurpose Internet Mail Extensions)

En su origen para extender el e-mail con textos no ASCII (imágenes...)

En la cabecera que precede a los datos se indica el tipo y el subtipo

MIME (ej: text/plain, text/html...)

# Formatos de texto plano, documentos, archivos y datos estructurados.

Texto plano: Sucesión de codificaciones de caracteres (ASCII, ISO o Unicode)  
Pequeñas diferencias entre sistemas de ficheros (final de línea)

Documentos: Diversos formatos "propietarios", estándares: ODF, OOXML

Archivos: fichero concatenación de ficheros, eventualmente comprimido.

Datos estructurados: conjunto de "registros" que tienen "campos".

## Representación digital de imágenes

dos maneras

- imágenes matriciales (con píxeles):
  - Se describe en 2 dimensiones.
  - Resolución:  $n^{\circ}$  píxeles  $1 \times n^{\circ}$  píxeles
  - Profundidad de color:  $2^0$  (2 colores)  $2^1$  (4 colores)  $2^2$  (16 colores)  $2^8$  colores (paleta de colores)  $2^4$  colores = true color
  - sin compresión
  - con compresión
- Gráficos vectoriales (objetos geométricos)
  - Describe de las imágenes como conjunto de objetos primitivos definidos por características geométricas.
  - No adecuado para imágenes fotográficas.
  - Se pueden escalas sin perder resolución
  - Normalmente ocupan menos memoria

## Formatos y códecs de audio

→ Formato de audio: convenios para el almacenamiento y la transmisión de la representación binaria de señales sonoras.

→ Códec de audio: - codificador: señal analógica vs datos binarios a fichero o a "streaming".  
- Decodificador: proceso inverso.

Tipos de formatos de audio: → Sin compresión: la mayoría basados en PCM (Pulse-code Modulation): valores codificados proporcionales a las amplitudes de la muestra a sus logaritmos.  
→ Con compresión: Aprovechan redundancias de la señal y características psicoacústicas para reducir el volumen y la tasa de datos.

- Sin pérdidas. Reducen como mucho a la mitad.
- Con pérdidas. Pueden reducir todo lo que se quiera, a costa de perder calidad.

PCM: En el tps que transcurre entre la muestra  $n$  y la muestra  $n+1$  (período de muestreo,  $t_s$ ) se genera un tren de impulsos que corresponde a la codificación binaria de la muestra  $n$ .

⚠ Si  $N$  es el volumen de datos resultante de PCM y  $N_c$  el resultante de compresión,

→ porcentaje de compresión:  $C = \frac{100 N_c}{N} \%$

→ factor de compresión:  $fc = k:1$ , con  $k = \frac{N}{N_c}$

## Formatos de vídeo, códecs y contenedores

Formato de contenedor: Describe un metafichero formado por varios ficheros de  $\neq$  tipos y metadatos con info. sobre el contenido, sobre la sincronización de los flujos, etc.

Un programa de reproducción de ficheros multimedia o de streams reconoce y abre el contenedor e incluye los códecs adecuados para los contenidos.

MPEG (Moving pictures Experts group): Grupo de trabajo de ISO/IEC que elabora normas para compresión y transmisión de multimedia.

Muchos fabricantes las adoptan, pero algunas implementaciones son cerradas o protegidas por patentes, lo que dificulta la interoperabilidad.

Procesador hardware: Sistema físico que ejecuta programas previamente almacenados en una memoria de acceso aleatorio.

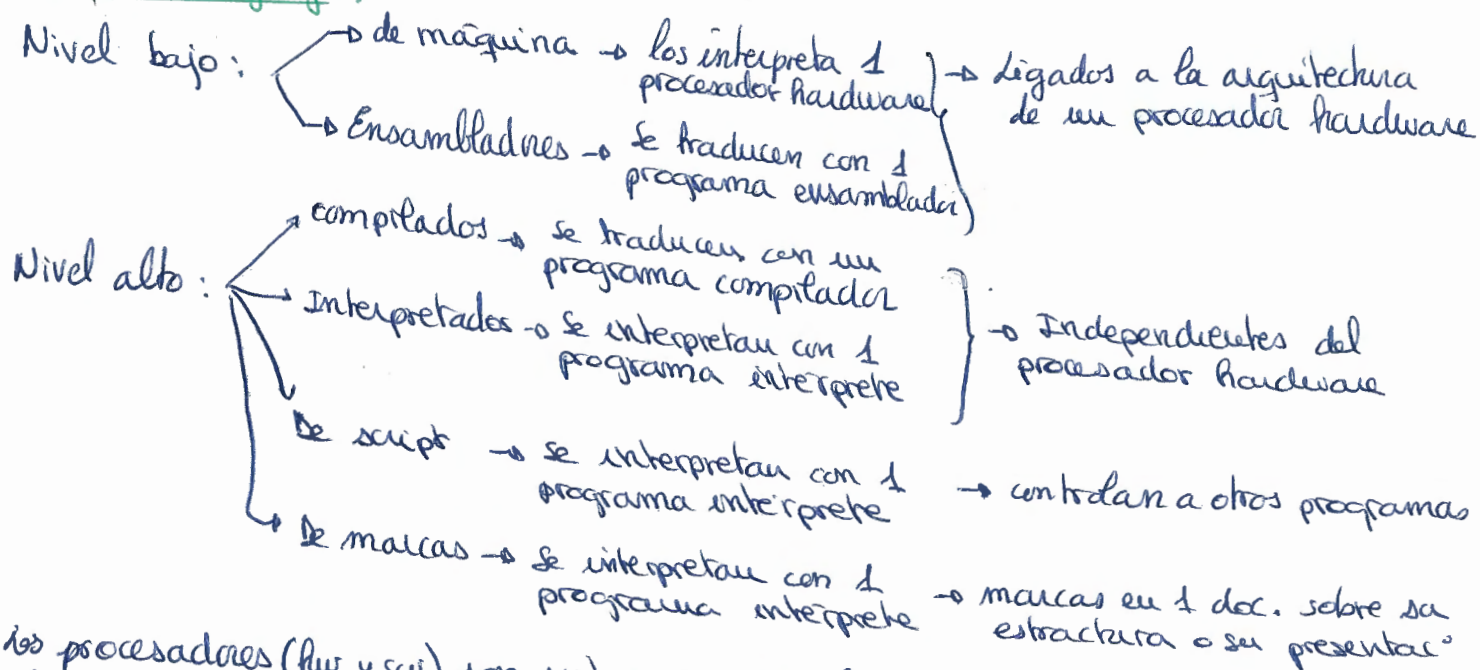
Procesador software: Programa que transforma unos datos de E en unos de S.

Lenguaje de máquina: Lenguaje binario que "entiende" el procesador hardware. Esta formado por el repertorio de instrucciones (conjunto de códigos de operación y de formatos de las instrucciones) y los convenios de representaci<sup>ón</sup> de los datos.

Lenguaje ensamblador: Representación simbólica, inteligible para las personas, del lenguaje de máquina.

Programa ensamblador: Traduce programas escritos en lenguaje ensamblador a lenguaje de máquina. Es un procesador software.

### Tipos de lenguaje:



Los procesadores (hw y sw) son sistemas complejos, solo pueden describirse haciendo abstracc<sup>ión</sup> de detalles. Dependiendo de qué es lo que interesa estudiar hay distintos niveles de abstracción y tipos de modelos.

Nivel de circuito eléctrico: conjunto de transistores, resistencias...

Nivel de dt. lógico: conjuntos de componentes que realizan operaciones como  $(0 \text{ or } 1 = 1)$

Nivel de máquina convencional: Buses, V<sub>AL</sub>, V<sub>D</sub>, R, V<sub>C</sub>... se hace abstracc<sup>ión</sup> de las puertas que componen las unidades.

Nivel de máquina operativa: las instrucciones de máquina más las llamadas al sistema. Se hace abstracción de gestión de procesos, de almacenamiento...

Nivel de máquina simbólica: en lugar de instrucciones de máquina, su expresión en lenguaje ensamblador (o en un lenguaje de alto nivel). Se hace abstracc<sup>ión</sup> de 0 y 1's.

Modelo: Esquema teórico de un sistema o de 1 realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento.

Modelo estructural: describe del sistema como conjunto de partes (subsistemas) y sus interrelaciones.

Modelo funcional: qué hace el sistema (cómo responde a las E, cómo se usa).

Modelo procesal: cómo lo hace (cómo funciona).

La máquina de von Neumann:

modelo estructural:

- La CPU es un procesador hardware (que tiene la VAL y la UC) (Hz)

MP y VE/S conectados por buses a la CPU.

modelo funcional:

- El contenido de 1 palabra (40 bits) puede interpretarse como  $n^{\circ}$  entero entre  $-2^{39} - 1$  y  $+2^{39} - 1$  (CD) o como  $n^{\circ}$  fraccionario con coma fija o como des instrucc. de 20 bits.
- Formato de instrucciones:

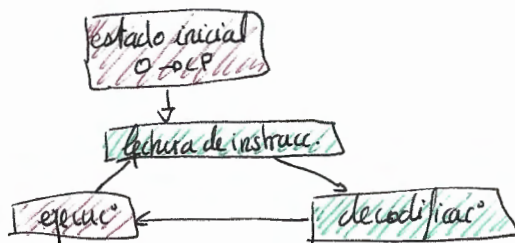


- El programa debe estar almacenado en la MP antes de comenzar su ejecución.
- Hay instrucciones de procesamiento y de bifurcación.

modelo procesal:

En cada paso de la ejecución la unidad de control:

- 1 Lee (extrae) de la MP la instrucción cuya dirección de memoria se encuentra en el R contador de programa.
- 2 Decodifica los bits de la instrucción.
- 3 Ejecuta la instrucción, generando las microórdenes oportunas.



- procesadores RISC:

Componentes: Procesadores (Reloj + acumulador + CPU) + Memorias + Periféricos

Interconexión por buses: Conjunto de conductores eléctricos que conectan varias "estaciones" que pueden ser unidades funcionales en un ordenador o nodos en 1 red. Un bus es un canal de difusión; lo que significa que cada estación recibe las transmisiones de todas las demás, y todas ellas tienen acceso al bus.

Modelo estructural básico: RE: registro de estado (Z, C, N, V)

CP: contador de programa (contiene la dirección de la instrucción que está en la fase de lectura). Es un registro.

Encadenamiento en 1 procesador: Las fases de ejecución de instrucciones se realizan a la vez.

Modelos funcionales:

CISC: varios niveles de instrucciones

- instrucc. para operaciones complejas
- varios formatos de instrucc. (uno o varios bytes).

RISC: solo las instrucciones + frecuentes.

- Formato de instrucciones regular,  $\equiv$  Todas las instrucc. tienen la misma longitud. y el n° de formatos  $\neq$  es reducido. En cada formato todos los bits tienen el mismo significado para todas las instrucc., y esto permite q. la UC sea + sencilla.
- Arquitectura "load/store"  $\equiv$  los únicos accesos a la MP son para extraer instrucc. y datos. Todas las operaciones de procesamiento se realizan en registros del procesador.
- 3 tipo de instrucc.:
  - de "mov." y procesamiento entre registros.
  - de transferencia de datos con la mem. y los periféricos.
  - de transferencia de control: bifurcaciones, llamadas y retornos de subprogramas, interrupción de programa.

### Direccionamiento de la mem.

Espacio de direccionamiento: si el bus <sup>de direcc.</sup> tiene  $n$  bits, el procesador puede generar  $2^n$  direcciones de bytes. Entre la MP y los registros se pueden transferir bytes o palabras de  $k$  bytes.

Espacios de direcc. independiente: las direcciones de E/S no tienen relación con las direcciones de la mem. La dirección  $d$  puede referirse a una posición de la mem. o a un puerto. En estos procesadores son necesarias instrucc. específicas para la E/S.

Espacios de direcc. compactado: ciertas direcciones generadas por el procesador no corresponden a la mem. sino a puertos. Por tanto el acceso a estos puertos no requiere instrucc. especiales, son las mismas que permiten acceso a la mem. Se dice que la E/S está "memory mapped".

Convencios:

- Almacenamiento de 1 palabra en varios bytes (Extremista mayor o menor)
- Direccionamiento de las palabras: En algunos procesadores es obligatorio que las direcciones de palabra sean múltiplos de  $k$ : "direcciones de palabra alineadas".

Modos de direccionamiento: (para obtener DE)

Direcc. directo: El de Von Neumann. La dirección está contenida en la misma instrucc.

Direcc. indirecto: La dirección de mem. no está en la instrucc., sino en otra palabra de la mem., o en un registro del procesador.

Direcc. indexado: La DE se obtiene sumando el contenido de  $1R$  ( $R$  de índice)

Direcc. relativo al CP: La DE se obtiene sumando una distancia ( $\oplus$  o  $\ominus$ ) al valor q. tiene en ese momento el contador de programa.

Direcc. inmediato: La instrucc. contiene el operando.

### Subprogramas:

Programa al q. pueden llamar otros programas para realizar una func. con  $\neq$  parámetros

Paso de parámetros: por registros <sup>(puede ser)</sup> o por zonas de mem. apuntadas por  $R$ . (paso de ref.)

Direcc. de retorno: En un  $R$  especial (registro de enlace,  $LR$ )

- instrucc. de llamada: guarda la dirección de retorno en  $LR$  y bifurca.
  - instrucc. de retorno: copia el contenido de  $LR$  en el contador de programa.
- En una pila de mem.

## Pila en RAM para llamadas a subprogramas:

• Registro puntero de pila, PP.

- Operaciones push y pop:
  - Instrucc. de llamada: guarda en la pila la direcc. de retorno (push) y bifurca al subprograma.
  - Instrucc. de retorno: saca de la pila la direcc. de retorno (pop) y la introduce en el contador de programa.

La pila tb. puede usarse para pasar los parámetros.

## Comunicación con periféricos:

Puertos de los controladores: registros que contienen:

- datos intercambiados con bus R del procesador
- o - info. sobre el estado.
- o - órdenes del procesador.

Cada puerto tiene una direcc. de E/S

Dos convenios sobre las direcciones:

- Espacios de direcc. independientes.
- Espacios de direcc. compartido.

## Interrupciones: combinaciones de hw y sw con muchas variantes:

En general, el procesador (hw) y el sw tienen que:

- pasar a modo supervisor
- inhibir otras interrupciones (para no perderse).
- Salvar el contexto del programa interrumpido: direcc. de retorno...
- Identificar la causa: interna, externa...
- Pasar a ejecutar la rutina de servicio
- Al terminar el servicio, restaurar el contexto
  - Permitir interrupciones
  - Volver a modo usuario.

## El procesador ARM:

ARM (Basic RISC Machine), ARM (Advanced RISC machine).

ARM: registro de estado, CPSR (current program Status register)

Al atender 1 interrupción se copia automáticamente en SPSR (saved PSR)

INZCV | . . . . . | I | 1 0 0 | Mode

0: interrupciones inhibidas  
1: interrupciones permitidas  
Supervisor o usuario

ARM: Modelo procesal: Cadena de profundidad 3 con instrucc. de 4 bytes.

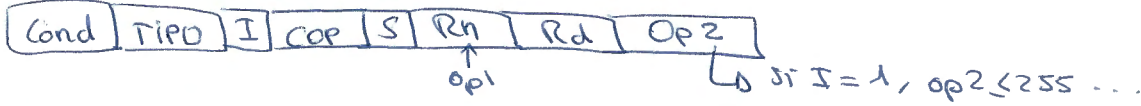
ARM: Formatos de instrucc.

- 00: procesamiento y mov. de datos (16)
- 01: Transferencias con MP y periféricos (11)
- 10: bifurcación (2)
- 11: interrupción de programa (1)

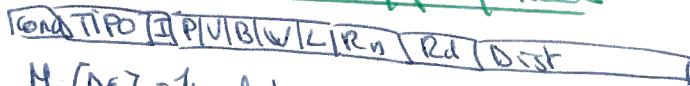
Todas las instrucciones son condicionales.

③

### Procesamiento y mov. op.



### Transferencia con MP y periféricos



M [DE] = 1 palabra o un byte de la mem, o un puerto (espacios compartidos)

DE = f(Rn, S, P, U, Dist)

Rn = registro de base

W = si W=1 DE se escribe en Rn

#### Modo inmediato postindexado (I=0, P=0)

DE = Rn

U=1 + si U=0, - (Rn) ± (Dist) → Rn

#### Modo inmediato preindexado (I=0, P=1)

DE = (Rn) ± (Dist)

U=1, + si U=0, -

si W=1, (Rn) ± (Dist) → Rn

#### Modo registro postindexado (I=1, P=0)

DE = (Rn)

(Rn) ± despl(Rm) → Rn

#### Modo registro preindexado (I=1, P=1)

DE = (Rn) ± despl(Rm)

si W=1, Rn ± despl(Rm) → Rn

### Transferencias de control.

Valor → RIS = PC

Instrucc. de bifurcación DE = dir. instrucc + 8 + 4 × Dist (± 32 MiB)

DE → RIS; si L=1, dir. instrucc + 4 → LR = R14

### Interrupciones en ARM.

Para cada causa una rutina de servicio

### Programación de ARM

Módulo: programa que:

→ puede ensamblarse, dando como resultado un cód. objeto (instrucc. de máquina en binario).

→ este código objeto no puede ejecutarse, y necesita otros módulos.

Todo programa que no sea trivial debe descomponerse en módulos para:

→ Facilitar la depuración: comprobación del buen funcionamiento de cada módulo y corrección de posibles errores.

→ Reutilizar los módulos para otros programas.

El montador es un programa que genera un cód. objeto ejecutable combinando los cód. objeto de los módulos.

En cada módulo tienen q. declararse:

→ los símbolos externos (están definidos en otros módulos) con la directiva

«extern» (para que el ensamblador no genere errores).

→ los símbolos de acceso (los pueden utilizar otros módulos) con la directiva  
«global» (para que el montador no genere errores).

### Procesadores software.

Procesador de lenguaje: Programa que procesa otros programas escritos en algún lenguaje no binario para que el resultado pueda ser interpretado por un procesador hw. (o x otro procesador de lenguaje intermedio).

El proceso de ensamblaje: Normalmente 1 ensamblador traduce en 2 pasos:

① Analiza el programa fuente. Si no hay errores sintácticos construye una tabla interna (etiquetas y sus direcc.)

② Vuelve a explorar las instrucciones del programa fuente y, con ayuda de la tabla de símbolos interna, va traduciendo cada una a binario. Genera un código objeto, una tabla de símbolos de acceso, una tabla de símbolos externos y un dicc. de reubicac.

Dicc. de reubicac.: lista de las direcc. cuyos contenidos deben cambiarse sumándoles d cuando el cód. objeto se cargue a partir de la direcc d y no de 0.  
(se suma d en la direcc. y en el contenido).

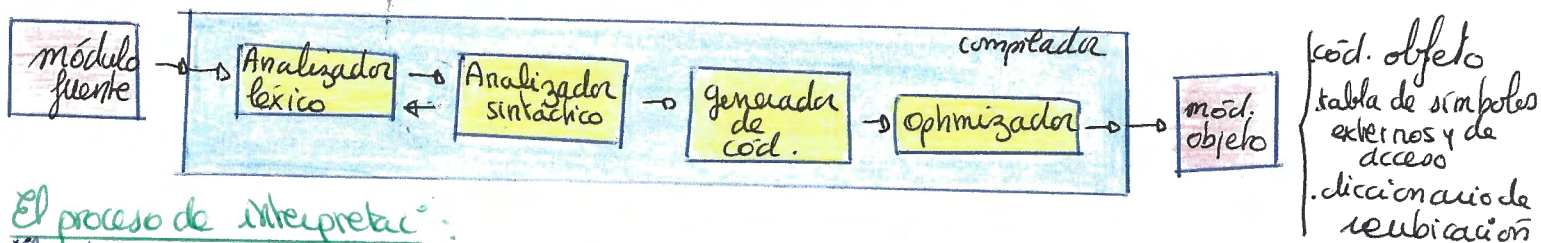
### El proceso de montaje:

El montador (linker): → coloca los módulos objeto generados por el ensamblador uno tras otro.  
→ pone los valores adecuados de acuerdo con las tablas de símbolos externos y de acceso.  
→ genera un código objeto ejecutable y un dicc. de reubicac. global.

### El proceso de compilac.:

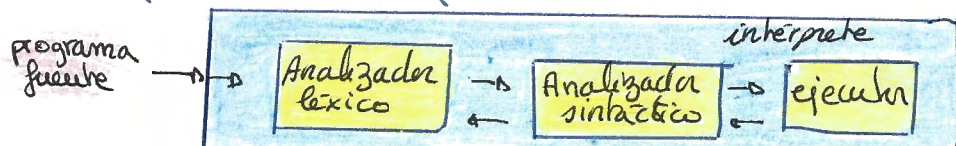
Funcionalm<sup>t</sup> un compilador hace lo mismo que un ensamblador.

Procesalm<sup>t</sup> es + complejo:



### El proceso de interpretación:

El intérprete no genera ningún 'cód. objeto'. Cada vez q. analiza 1 sentencia genera los órdenes oportunos para el procesador hw.



# Lenguajes de marcas y de script.

## Tipos de marcado:

- Marcado para presentac<sup>o</sup>: Las marcas q. se añaden al texto son cód.; frecuentem<sup>t</sup> binarios, que indican la forma en la q. ha de presentarse (tipo, tamaño...). Este marcado queda oculto al usuario, incluso al mismo autor. (ej. Word)
- Marcado procedimental: Las marcas no son cód. binarios sino etiquetas legibles y manipulables directam<sup>t</sup> por el autor. A veces estas marcas señalan tb aspectos de la presentac<sup>o</sup>. (ej. HTML)
- Marcado descriptivo (o semántico): Las marcas o etiquetas no indican directam<sup>t</sup> la forma de presentac<sup>o</sup>. Asignan el texto marcado a una categoría para que el procesador se sepa de qué se trata. (ej. basados en XML).

HTML: elemento raíz <html>  $\begin{cases} \text{<head>} \\ \text{<body>} \end{cases}$

XML: XML está concebido para el almacenamiento, el transporte y el intercambio de datos entre aplicaciones, no para su presentación.

- Las marcas (etiquetas) en XML no tienen un significado predefinido.
- Es un metalinguaje; con doc. adicionales se definen lenguajes específicos.

Un analizador sintáctico (parser) es un procesador que comprueba que el doc. respeta esas reglas.

## Semántica informal de XML

Para interpretar el contenido de un doc. XML particular se utiliza otro doc. + general que define la semántica para todos los doc. XML de ese tipo.

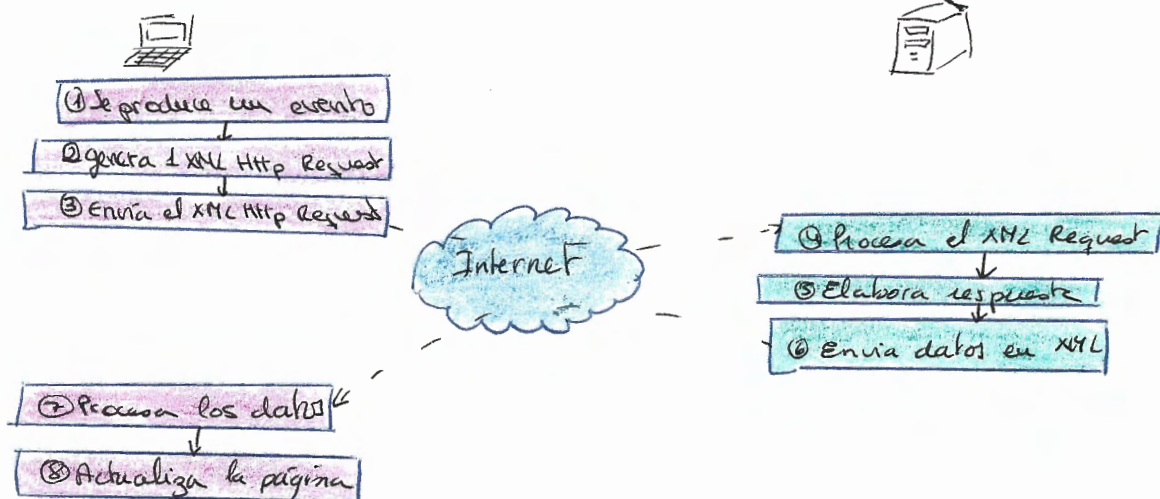
DTD y XML Schema permiten definir lenguajes específicos mediante reglas q. expresan la forma válida de construir doc. en esos lenguajes.

El doc. XML es un doc. válido si, adt de estar bien formado, satisface la especificac<sup>o</sup> de un DTD o un XML schema.

## JavaScript:

- No es Java
- lenguaje interpretado que se ejecuta en un navegador (o en otra aplicac<sup>o</sup>)

AJAX: interacc<sup>o</sup> dinámica entre el servidor y el cliente.



• El n° de bits de 1 palabra es siempre un múltiplo de 8 y coincide con el ancho del bus de datos.

• "Arquitectura de  $x$  bits"  $\equiv$  n° de bits de la palabra (16, 32, 64)

• Si las direcciones son de  $x$  bits, la capacidad de direccionamiento es de  $2^x$ .

• Direcciones generadas por el procesador  $\equiv$  direcciones virtuales.

→ HL → 16 registros ( $R_0 - R_{15}$ )

$R_0 - R_{12}$  → propósito general

$R_{13}$  → (SP) puntero de pila

$R_{14}$  → (LR) registro de enlace }  $\times 2$  modo usuario y modo privilegiado.

$R_{15}$  → (PC) contador de programa

Registro de estado CPSR → estado del procesador (no está en HL)

→ Las instrucc. aritméticas de ARM operan sobre  $\mathbb{Z}$  representados en 32 bits:

- El convenio para  $n^\circ < 0$  es el C2

- El convenio de almacenamiento en memoria es extremista menor.

- Las direcc. de HP deben estar alineadas: un  $n^\circ$  representado en 1 palabra (32 bits) tiene que tener su Byte menos significativo en una direcc. múltiplo de 4.

1.1)

- dispositivos que se conectan a Internet: hosts o sistemas terminales.
- Los sistemas terminales se conectan entre sí mediante una red de enlaces de comunicación y dispositivos de conmutación de paquetes.
- Velocidad de transmisión: bits por segundo (bps)
- Un conmutador de paquetes toma el paquete que llega de uno de sus enlaces de comunicaciones de entrada y lo reenvía a uno de sus enlaces de comunicaciones de salida.
- Dispositivos de conmutación de paquetes:
  - routers (en el núcleo de la red)
  - switches de la capa de enlace (en las redes de acceso).
- Los sistemas terminales acceden a Internet a través de los ISP (Internet Service Provider). Cada ISP es en sí mismo una red de conmutadores de paquetes y enlaces de comunicaciones.
- Internet es todo lo que se conecta a los sistemas terminales entre sí, por lo que los ISP que proporcionan el acceso a los sistemas terminales tb. tienen que estar interconectados entre ellos.
- Los sistemas terminales, los conmutadores de paquetes y otros dispositivos de Internet ejecutan protocolos que controlan el envío y la recepción de la info. dentro de Internet.

Dos de los protocolos + imp.:

TCP (Transmission Control protocol)

IP (Internet Protocol) → especifica el formato de los paquetes que se envían y reciben entre los routers y los sistemas terminales.

- Los estándares de Internet son desarrollados por el IETF (Internet Engineering Task Force). Los documentos asociados a estos estándares IETF: RFC (Request for Comments).

- Internet: una infraestructura que proporciona servicios a las aplicaciones.

- Se dice que las aplicaciones son aplicaciones distribuidas, porque implican a varios sistemas terminales que intercambian datos entre sí. Es importante saber que las aplicaciones de Internet se ejecutan en los sistemas terminales.

- Los sistemas terminales conectados a Internet proporcionan una API (Application Programming Interface) que especifica cómo un programa de src. que se ejecuta en un sistema terminal pide a la infraestructura de Internet que suministre datos a un programa de destino específico que se ejecuta en otro sistema terminal. La API de Internet consta de un conjunto de ~~normas~~ reglas que el programa que transmite los datos debe cumplir para que Internet pueda entregar esos datos al programa de destino.
- Internet es una infraestructura en la que están inventando e implementando constantemente nuevas aplicaciones.
- Cualquier actividad de Internet que implique dos o más entidades remotas que se comunican está gobernada por un protocolo.
- Protocolos implementados por hw en las tarjetas de interfaz de red de dos computadoras conectadas físicamente controlan el flujo de bits a través del "cable" conectado entre las 2 tarjetas de interfaz de red.
- Protocolos de control de congestión de los sistemas terminales controlan la velocidad a la que se transmiten los paquetes.
- Protocolos de los routers determinan la ruta que seguirá un paquete desde el origen al destino.
- Un protocolo define el formato y el orden de los mensajes intercambiados entre 2 o + entidades que se comunican, así como las acciones tomadas en la transmisión y/o la recepción de un mensaje u otro suceso.

1.2)

- host { clientes  $\rightarrow$  computadoras de escritorio y portátiles, PDA ...  
servidores  $\rightarrow$  equipos + potentes que almacenan y distribuyen  
páginas web, flujos de video, correo electrónico ...

- Un programa cliente es un programa que se ejecuta en un sistema terminal que solicita y recibe un servicio de un programa servidor que se ejecuta en otro sistema terminal.
- Un programa cliente normalmente se ejecuta en una computadora y el programa servidor en otra, las aplicaciones Internet cliente-servidor son aplicaciones distribuidas
- No todas las aplicaciones de Internet actuales están construidas por programas cliente puros que interactúan con programas servidor puro. Cada vez + aplicaciones son aplicaciones entre iguales o entre pares (P2P, peer-to-peer), en las que los sistemas terminales interactúan y ejecutan programas que realizan tanto funciones de cliente como de servidor.
- Infraestructura de la tlf cableada tradicional local  $\equiv$  telco local. Cada residencia dispone de un enlace directo de cable de par trenzado a un switch de la compañía tlf, el cual se encuentra en un edificio denominado central telefónica.
- "Acceso telefónico"  $\rightarrow$  el sus. del usuario realmente llama al n.º de tlf de un ISP y establece una conex. tlf tradicional con el mismo.  
 $\rightarrow$  inconvenientes  $\rightarrow$  lento (máx. 56 kbps) y ocupa la línea tlf.
- Acceso residencial de banda ancha  $\rightarrow$  DSL  $\rightarrow$  cable
- Un canal de descarga de alta velocidad opera en la banda de 50 kHz a 1 MHz
- — de carga de velocidad media opera en la banda de 4 kHz a 50 kHz
- — telefónico ordinario bidireccional opera en la banda de 0 a 4 kHz.
- Ventajas DSL  $\rightarrow$  velocidad (descargas: 1-2 Mbps; cargas: 128 kbps-1 Mbps).  
 $\rightarrow$  acceso asimétrico.

Cable: En un sistema de TV por cable tradicional, el terminal de cabecera de cable difunde los canales de TV a través de una red de distribución de cable coaxial y amplificadores hasta los domicilios de los usuarios. En este sistema se utiliza cable coaxial + fibra (HFC  $\rightarrow$  Hybrid Fiber Coax).

- El acceso por cable a Internet requiere el uso de modems especiales, que se conocen como modems por cable. (= dividen la red HFC en 2 canales: cargas y descargas). El acceso es asimétrico (Velocidad de carga > Velocidad de descarga).
- Una característica imp. del acceso a Internet por cable es que se trata de un medio de difusión compartido.

### Fibra óptica (Tecnología FTTH):

La red de distribución óptica + simple se denomina fibra directa, en la que existe una fibra que sale de la central hasta cada domicilio. Este tipo de distribución puede proporcionar un ancho de banda gd. Lo + habitual es que cada fibra saliente de la central sea compartida por muchas viviendas. Hay disponibles 2 arquitecturas:

→ redes ópticas activas (AON) → redes Ethernet conmutadas

→ redes ópticas pasivas (PON) → Cada vivienda dispone de una terminación de red óptica (ONT), que se conecta a un distribuidor del veandario mediante un cable de fibra óptica dedicado.

El distribuidor combina una cierta cantidad de viviendas en único cable de fibra óptica compartido, que se conecta a una terminación de línea óptica (OLT) de la central de compañía de tlf.

La OLT realiza la conversión de señales ópticas en eléctricas, se conecta a través de Internet mediante un router de la compañía de tlf.

En la arquitectura PON, todos los paquetes enviados desde la OLT al distribuidor se replican en este distribuidor.

La fibra óptica permite proporcionar servicios de TV, y el servicio de telefonía normal.

Ethernet: red de área local (LAN) → Ethernet tecnología de acceso predominante en las redes corporativas y universitarias.

Wifi: Existen dos tipos de acceso inalámbrico a Internet. En una LAN inalámbrica, los usuarios inalámbricos transmiten paquetes a un pt de acceso, el cual a su vez está conectado a la red Internet cableada. Los usuarios de una LAN inalámbrica deben encontrarse a unas pocas decenas de metros del pt de acceso. En las redes inalámbricas de área extensa, los paquetes se transmiten a una estación base a través de la misma infraestructura inalámbrica utilizada por la telefonía móvil.

• Acceso inalámbrico de área extensa: (3G) utilizan la infraestructura de la telefonía móvil para acceder a estaciones base que están separadas entre sí (por km).

• WiMAX opera independientemente de la red de telefonía móvil y promete velocidades comprendidas entre 5 y 10 Mbps o superiores para distancias de decenas de km.

Medios físicos: En cada transmisor-receptor, el bit se envía mediante ondas electromagnéticas o pulsos ópticos a lo largo de un medio físico.

(ej: cable de cobre de par trenzado, cable coaxial, el cable de fibra óptica multimodo, el espectro terrestre y el espectro de radio por satélite).

Los medios físicos: 2 categorías

- guiados (las ondas se transportan a través de un medio sólido).
- no guiados (las ondas se propagan por la atmósfera y el espacio exterior).

Cable de par trenzado: medio de transmisión guiado + barato.

- Los hilos se trenzan para reducir las interferencias eléctricas procedentes de pares similares próximos.
- Un par de hilos constituyen un único enlace de comunicaciones.
- El par trenzado no apantallado (UTP) se utiliza <sup>en las</sup> redes LAN.
- La velocidad de transmisión de datos es de 10 Mbps y 1 Gbps. (depende del espesor del cable y de la distancia entre el transmisor y el receptor).

Cable coaxial: • Consta de 2 conductores de cobre, pero dispuestos de forma concéntrica.

- Velocidad de transmisión > 1 Mbps
- Es común en los sistemas de TV por cable, el transmisor desplaza la señal digital a una banda de frecuencia específica y la señal analógica resultante se envía desde el transmisor a uno o más receptores.
- Puede utilizarse como un medio compartido guiado.

Fibra óptica: • medio flexible y de poco espesor que conduce pulsos de luz, representando cada pulso un bit.

- La fibra óptica es inmune a las interferencias electromagnéticas presenta una atenuación de la señal muy baja y es muy

difícil que alguien pueda llevar a cabo un "pinchazo" en una de estas líneas.

- Es muy caro (transmisores, receptores, conmutadores).
- Las velocidades del enlace estándar de portadora óptica:  
 $51'8 \text{ Mbps} \approx 39,8 \text{ Gbps}$ ; suele hacerse referencia a estas especificaciones como OC-n donde la velocidad de enlace es  $n \times 51'8 \text{ Mbps}$

Canales de radio terrestres: • Transportan señales en el espectro electromagnético.  
• Medio no guiado.  
• 2 grupos

- los que operan en áreas locales (redes LAN inalámbricas)
- ————— en áreas extensas (tecnologías celulares).

Canales de radio vía satélite: Las comunicaciones por satélite enlazan 2 o + transmisores/receptores de microondas con base en la Tierra, que se conocen como estaciones terrestres. El satélite recibe las transmisiones en una banda de frecuencia, regenera la señal utilizando un repetidor y transmite la señal a otra frecuencia. En este tipo de comunicaciones se emplean dos tipos de satélite:

- geoestacionarios
- de baja órbita (LEO).

1.3)

(4)

- Existen 2 métodos fundamentales que permiten transportar los datos a través de una red de enlaces y conmutadores: la conmutac<sup>o</sup> de circuitos y la conmutac<sup>o</sup> de paquetes.

↓

El paquete se transmite a través de una serie de enlaces de comunicaciones. Pero con la técnica de conmutación de paquetes, el paquete se envía a la red sin haber reservado ancho de banda. Si uno de los enlaces está congestionado porque otros paquetes tienen que ser transmitidos a través de él al mismo tiempo, entonces nuestro paquete tendría que esperar en un buffer en el lado del emisor del enlace de transmisión y, por tanto, sufrirá un retardo.

### Conmutación de paquetes

- Las aplicaciones distribuidas intercambian mensajes para llevar a cabo sus tareas.
- En las redes de computadoras modernas, el origen divide los sms largos en fragmentos de datos + pequeños  $\equiv$  paquetes. Entre el origen y el destino, cada uno de estos paquetes viaja a través de los enlaces de comunicaciones y de los ~~switches~~ conmutadores de paquetes (routers y switches de capa de enlace).
- La mayoría de los conmutadores de paquetes emplean el método de transmisión de almacenamiento y reenvío en las entradas de los enlaces. Transmis<sup>o</sup> de almacenamiento y reenvío  $\equiv$  que el conmutador tiene que recibir el paquete completo antes de poder transmitir el 1<sup>er</sup> bit del paquete al enlace de salida. Por tanto, los conmutadores de paquetes de almacenamiento y reenvío añaden un retardo de almacenamiento y reenvío<sup>(r)</sup> en la entrada de cada enlace.

$$r = \frac{QL}{R} \quad \text{siendo} \quad \begin{cases} Q = \text{nº de enlaces entre 2 hosts.} \\ L = \text{Un paquete de } L \text{ bits} \\ R = \text{Velocidad de cada enlace } R \text{ (bps).} \end{cases}$$

- Cada conmutador de paquetes tiene varios enlaces conectados a él y para cada enlace conectado, el conmutador de paquetes dispone de un buffer de salida ( $\equiv$  cola de salida) q. almacena los paquetes que el router enviará a través de dicho enlace  $\Rightarrow$  se produce retardo de cola. Si el buffer está completam<sup>t</sup> lleno se produce pérdida de paquetes.

- retardos impredecibles y variados.
- buena compartición del ancho de banda
- sencilla, eficiente, menos cara.

¿Cómo atraviesan los paquetes las redes de conmutac<sup>o</sup> de paquetes?

- En Internet, cada paquete que atraviesa la red contiene en su cabecera la direcc<sup>o</sup> del destino del paquete.
- Cada router dispone de una tabla de reenvío que asigna las direcciones de destino a los enlaces salientes.
- Un protocolo de enrutamiento determina la ruta + corta desde cada router hasta cada destino.

Redes troncales de Internet y proveedores ISP

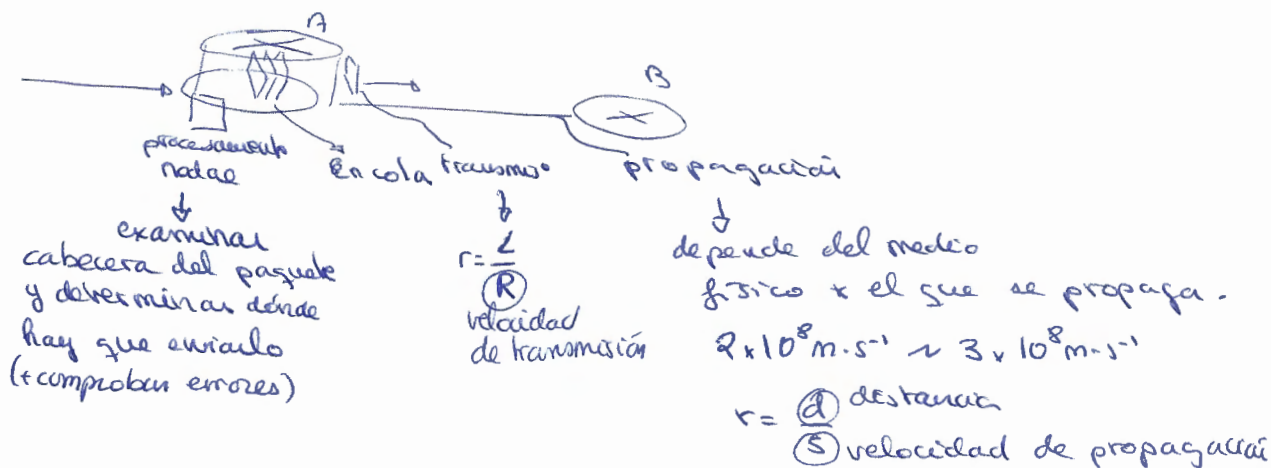
- Internet es una red de redes.
- En la red pública de Internet, los ISP de acceso situados en la frontera de Internet están conectados al resto de Internet a través de una jerarquía de niveles de proveedores ISP.
- ISP de nivel 1  $\equiv$  Redes troncales de Internet.
- ISP de nivel 2  $\equiv$  cliente del ISP de nivel 1 al que está conectado ~~al~~  
 $\Rightarrow$  ISP de nivel 1 es un proveedor de dicho cliente.
- Nivel más bajo de la jerarquía  $\equiv$  ISP de acceso.
- Cuando los ISP están conectados directamente entre sí en el mismo nivel  $\equiv$  igualitarios.
- Dentro de la red de un ISP, los pts en los que el ISP se conecta a otros ISP  $\equiv$  ~~estas~~ Pts de presencia (POP).

1.4)

Retardo nodal  $\equiv$  en un único router.

(5)

Retardos que sufre un paquete: Retardo de procesamiento nodal, retardo de cola, retardo de transmisión y retardo de propagación. La suma es el retardo nodal total.



### • Retardo de cola

• varía de 1 paquete a otro.

intensidad de tráfico:  $\frac{L_a}{R}$  velocidad media a la que llegan los bits a la cola.  
(R) velocidad de transmisión

$$\frac{L_a}{R} > 1 \Rightarrow r \rightarrow \infty$$

$$\frac{L_a}{R} \leq 1 \Rightarrow r = \frac{(n-0)L}{R}$$

### • Pérdida de paquetes

Una cola para acceder a un enlace tiene una capacidad finita. Si no hay sitio para almacenar un paquete, el router lo elimina  $\equiv$  el paquete se pierde.

Retardo total (origen  $\rightarrow$  destino)  $\Rightarrow$  Tracerout lo muestra

$$R_T = N (\text{retardo procesamiento nodal} + \text{Retardo de transmisión} + \text{Retardo propagación})$$

Siendo N-1 routers entre origen y destino.

### • Tasa de transferencia en las redes de computadoras

• Además de los retardos y la pérdida de paquetes, otra medida crítica de rendimiento de las redes de computadoras es la tasa de transferencia de terminal a terminal.

• La tasa de transferencia instantánea en cualquier instante de tps es la velocidad a la que el host recibe el archivo.

Si el archivo consta de F bits y la transferencia dura T segundos  $\Rightarrow$  la tasa media de transferencia del archivo es  $\frac{F}{T}$  bps.

$R_s$   $\rightarrow$  velocidad del enlace entre el servidor y el router.

$R_c$   $\rightarrow$   entre el router y el cliente.

$$\Rightarrow \text{tasa de transferencia} = \min \{R_c, R_s\}$$

$$\Rightarrow \text{velocidad de transmisión del enlace cuello de botella} = \frac{F}{\min \{R_c, R_s\}}$$

- La tasa de transferencia depende de las velocidades de transmisión de los enlaces a través de los que fluyen los datos.
- Si no existe ningún otro tráfico, la tasa de transferencia = velocidad mínima de transmisión a lo largo de la ruta origen-destino.
- La tasa de transferencia depende de velocidades de transmisión de los enlaces y del tráfico existente.

En particular, un enlace con  $\pm$  velocidad de transmisión alta puede ser el enlace cuello de botella para la transferencia de un archivo si hay muchos otros flujos de datos atravesando to ese enlace.

1.5)

- Los protocolos son organizados en capas.
- Servicios que ofrece una capa a la capa que tiene por encima  $\equiv$  modelo de servicio de capa
- Una capa de protocolo puede implementarse por sw, por hw o mediante una combinac<sup>o</sup> de ambos.
- Cuando los protocolos de las distintas capas se toman en conjunto se habla de pila de protocolos. La pila de protocolos de Internet consta de 5 capas:

→ La capa de aplicac<sup>o</sup> es donde residen las aplicaciones de red y sus protocolos.  
Un protocolo de la capa de aplicac<sup>o</sup> está distribuido a lo largo de varios sistemas terminales, estando la aplicac<sup>o</sup> en el sist. terminal que utiliza el protocolo para intercambiar paquetes de info. con la aplicac<sup>o</sup> de otro sist. terminal.  $\downarrow$  (=mensaje)

→ La capa de transporte de Internet transporta los sms de la capa de aplicac<sup>o</sup> entre los pts terminales de la aplicac<sup>o</sup>.  
Protocolos de transporte: TCP (=servicio de <sup>transferencia de datos fiable</sup> conex<sup>o</sup>) y UDP (=mecanismo de control de congest<sup>o</sup>).  
Los paquetes de la capa de transporte  $\equiv$  segmentos.

→ La capa de red de Internet es responsable de trasladar los paquetes de la capa de red ( $\equiv$  datagramas) de un host a otro.  
Proporciona el servicio de suministrar el segmento de la capa de transporte del host destino.  
Incluye el protocolo IP

→ La capa de enlace traslada un paquete de un nodo al siguiente nodo.  
paquetes capa de enlace  $\equiv$  tramas.

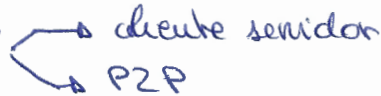
→ Capa física mueve los bits individuales dentro de la trama de un nodo a otro.

- Los striches de la capa de enlace implementan las capas 1 y 2 y los routers 1 a 3.
- encapsulac<sup>o</sup>  $\equiv$  añadir info a la cabecera para que el paquete pueda seguir su camino.



# Arquitectura de la aplicac°

(17)



→ Cliente-servidor: se existe un host activo denominado servidor. Los host clientes pueden estar activos de forma o intermitentemente.

En una aplicac° cliente-servidor un único host servidor es incapaz de responder a todas las solicitudes de sus clientes → se utiliza agrupación de hosts = centro de datos para crear un servidor virtual de gran capacidad.

Tiene infraestructura intensiva (es caro.)

→ P2P → aplicac° que explota la comunicac° directa entre parejas de hosts conectados de forma intermitente conocidos como peers.

auto-escalabilidad: aunque cada peer genera una carga de W solicitando archivos, tb. añaden capacidad de servicio al sistema distribuyendo archivos a otros peers.

• Los procesos se comunican.

proceso = programa q. se ejecuta dentro de 1 sist. terminal.

Los procesos de 2 sist. terminales se comunican entre ellos intercambiando sms a través de la red de computadoras.

• El proceso que inicia la comunicac° = cliente

• El proceso que espera a ser conectado = servidor.

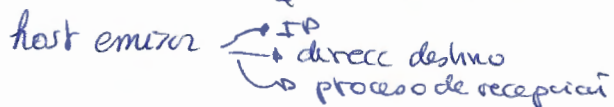
• Un proceso envía sms a la red y los recibe de la red a través de una interfaz su denominada socket.

• Un socket es la interfaz entre la capa de aplicac° y la capa de transporte de un host. (= Interfaz de programación de aplicaciones API) -

• servicios que puede ofrecer un protocolo de la capa de transporte:

transferencia de datos fiable, tasa de transferencia (las aplicaciones con requisitos de tasa de transferencia = aplicaciones sensibles al ancho de banda; Aplicaciones elásticas pueden hacer uso de la tasa de transferencia que haya disponible), temporización y seguridad.

• El host se identifica con su direcc. IP.



• Un protocolo de la capa de aplicac° define cómo los procesos de 1 aplicac° q. se ejecutan en 2 sist. terminales se pasan los sms entre sí.

Define → Tipo de sms intercambiados

Sintaxis de los sms

Semántica de los campos.

reglas (cómo se envían y reciben sms).



• WWW  $\equiv$  World Wide Web

• ~~HTTP~~ El protocolo de transferencia de hipertexto es el protocolo de la capa de aplicación de la Web y se encuentra en el corazón de la Web.

• HTTP se implementa en 2 programas  $\left. \begin{array}{l} \rightarrow \text{servidor} \\ \rightarrow \text{cliente} \end{array} \right\}$  se ejecutan en  $\neq$  terminales

se comunican entre sí intercambiando sms HTTP.

• Una pág. Web consta de objetos ( $\equiv$  archivos  $\leq$  en base HTML).

• HTTP es un protocolo sin memoria de estado.

•  $\text{tiempo ida y vuelta} = 2RTT(\text{tiempo ida y vuelta}) + t_{\text{transmisión}}$

• conexiones no persistentes: cada par solicitud/respuesta se envía a través de una conexión TCP separada.

• conexiones persistentes: todas las solicitudes y respuestas se envían a través de la misma conexión TCP

• mensajes de solicitud HTTP: La 1ª línea de un sms de solicitud HTTP se denomina línea de solicitud y las siguientes líneas son las líneas de cabecera. La línea de solicitud consta de 3 campos: el campo de método, el de URL y el de la versión HTTP.

• mensajes de respuesta HTTP: línea de estado inicial, líneas de cabecera y cuerpo de entidad (todo lo que contiene el objeto solicitado).

• Un navegador genera líneas de cabecera en función del tipo y la versión del navegador, de la configuración del navegador que tenga el usuario y de si el navegador actualmente tiene en cache una versión del objeto, posiblemente desactualizada.

Los servidores web se comportan de forma similar: existen productos, versiones y configuraciones  $\neq$ , que influyen en las líneas de cabecera que se incluirán en los sms de respuesta.

• HTTP utiliza cookies para identificar a los usuarios.

cookies  $\rightarrow$  una línea de cabecera de la cookie en el sms de respuesta HTTP.  
 • una línea de cabecera de la cookie en el sms de solicitud HTTP.  
 • el archivo de cookies almacenado en el sist. terminal del usuario y gestionado por el navegador del usuario  
 • una base de datos back-end en el sitio web.

Una cache-web, tñ. denominada servidor proxy, es una entidad de red que satisface solicitudes HTTP en nombre de 1 servidor web de origen. La cache web dispone de su propio almacenamiento en disco y mantiene en él copias de los objetos solicitados recientemente.

- Una cache es a la vez un servidor y un cliente.
- La cache web se ha implantado en Internet por dos razones:
  - reduce el tps de respuesta a la solicitud del cliente
  - reducen el tráfico de enlace de acceso a Internet de una institución.
- Aunque el almacenamiento en cache puede reducir los tiempos de respuesta percibidos por el usuario, introduce un nuevo pb: la copia de un objeto que reside en la cache puede estar desactualizada. HTTP dispone de un mecanismo que permite a la cache verificar que sus objetos están actualizados. Este mecanismo se denomina GET condicional.

## 2.5)

- Un identificador para los hosts es el nombre de host (= no proporciona info. acerca de la ubicación del host dentro de Internet).
- Los hosts tb. se identifican mediante direcc. IP (De izda a dcha se va obteniendo info + presión de dónde está el host dentro de Internet).
- Las personas prefieren utilizar como identificador el nombre de host mientras que los routers prefieren emplear las direcc. IP.
- ⇒ Servicio de directorio que traduzca nombres de host a direcc. IP ⇒ **SISTEMA DE NOMBRES DE DOMINIO (DNS)**.

- DNS es una base de datos distribuida implementada en una jerarquía de servidores DNS y un protocolo de capa de aplicac<sup>o</sup> que permite a los host consultar la base de datos distribuida.
- Alias de host: Un host con nombre complicado (⇒ host canónico) puede tener 1 o más alias.
- Alias del servidor de correo (mismo para las direcc. de correo electrónico).
- Distribución de carga: DNS tb. se emplea para realizar la distribuc<sup>o</sup> de carga entre servidores replicados.

Para los servidores web replicados hay asociado un conjunto de direcc. IP con 1 único nombre de host canónico. La base de datos DNS contiene este conjunto de direcc., el servidor responde con el conjunto completo de direcc. IP, pero rota el orden de las direcc. de cada respuesta.

- DNS: base de datos distribuida en Internet.  
Si solo hubiese 1 DNS ⇒ (si falla ⇒ Internet falla)  
⇒ Volumen de tráfico ⇒ congestión de redes  
⇒ Bases de datos centralizadas ⇒  
⇒ Mantenimiento ⇒ muy difícil.

DNS :  $\left\{ \begin{array}{l} - \text{raíz} \\ - \text{dominio de nivel superior} : \text{responsables dominios superiores (.com...)} \\ - \text{autoritativos} : \text{registros DNS que relacionan host con direcc. IP.} \end{array} \right.$

Servidor DNS local (x ejemplo en una uni.).

- Almacenamiento en caché DNS: mejorar retardos y reducir tráfico.  
Le permite no tener que consultar continuamente otros servidores DNS.

2 clases de ms DNS  $\left\{ \begin{array}{l} \rightarrow \text{respuesta} \\ \rightarrow \text{consulta} \end{array} \right\}$  utilizan el mismo formato.

• Un registrador es una entidad comercial que verifica la unicidad del nombre de dominio y lo añade a la base de datos.