

Tema 1. Introducció

Joan Manuel Parcerisa



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



Presentació de EC

Presentació de EC

- Professor de teoria
 - Joan Manuel Parcerisa
 - jmanel@ac.upc.edu
 - Despatx C6-116

Presentació de EC

- Professor de teoria
 - Joan Manuel Parcerisa
 - jmanel@ac.upc.edu
 - Despatx C6-116
- Web de l'assignatura
 - <http://docència.ac.upc.edu/FIB/grau/EC/>
 - Conté
 - Apunts de l'assignatura
 - Col·lecció de Problemes
 - Enunciats, plantilles i simulador MARS per a fer les pràctiques a casa
 - Exàmens anteriors
 - Calendari de sessions de laboratori
 - etc.
 - Descàrrega de documents
 - Username: privatEC
 - Password: Secure2010

Presentació de EC

- Professor de teoria
 - Joan Manuel Parcerisa
 - jmanel@ac.upc.edu
 - Despatx C6-116
- Web de l'assignatura
 - <http://docència.ac.upc.edu/FIB/grau/EC/>
 - Conté
 - Apunts de l'assignatura
 - Col·lecció de Problemes
 - Enunciats, plantilles i simulador MARS per a fer les pràctiques a casa
 - Exàmens anteriors
 - Calendari de sessions de laboratori
 - etc.
 - Descàrrega de documents
 - Username: privatEC
 - Password: Secure2010
- Racó
 - <https://raco.fib.upc.edu/>

EC dins el pla d'estudis

S1	<div>FM Fonaments Matemàtics 7,5 ECTS</div>	<div>F Física 7,5 ECTS</div>	<div>PRO1 Programació 1 7,5 ECTS</div>	<div>IC Introducció als Computadors 7,5 ECTS</div>	
S2	<div>M1 Matemàtiques 1 7,5 ECTS</div>	<div>M2 Matemàtiques 2 7,5 ECTS</div>	<div>PRO2 Programació 2 7,5 ECTS</div>	<div>EC Estructura de Computadors 7,5 ECTS</div>	
S3	<div>PE Probabilitat i Estadística 6 ECTS</div>	<div>BD Bases de Dades 6 ECTS</div>	<div>SO Sistemes Operatius 6 ECTS</div>	<div>EDA Estructura de Dades i Algorismes 6 ECTS</div>	<div>CI Interfícies de Computadors 6 ECTS</div>
S4	<div>EEE Empresa i Entorn Econòmic 6 ECTS</div>	<div>IES Introducció a l'Enginyeria del Software 6 ECTS</div>	<div>XC Xarxes de Computadors 6 ECTS</div>	<div>PROP Projectes de Programació 6 ECTS</div>	<div>AC Arquitectura de Computadors 6 ECTS</div>

Presentació de EC

- Teoria
 - Tema 1 (1a part). Presentació
 - Tema 2. Instruccions i tipus de dades bàsics
 - Tema 3. Traducció de programes
 - Tema 4. Matrius
 - Tema 1 (2a part). Rendiment i Consum
 - Tema 5. Aritmètica d'enters i coma flotant
 - Tema 6. memòria cache
 - Tema 7. Memòria virtual
 - Tema 8. Excepcions i interrupcions

Presentació de EC

- Teoria

- Tema 1 (1a part). Presentació
- Tema 2. Instruccions i tipus de dades bàsics
- Tema 3. Traducció de programes
- Tema 4. Matrius
- Tema 1 (2a part). Rendiment i Consum
- Tema 5. Aritmètica d'enters i coma flotant
- Tema 6. memòria cache
- Tema 7. Memòria virtual
- Tema 8. Excepcions i interrupcions

- Bibliografia

- Bàsica: **Apunts** a la web (no cobreixen tot el temari)
- Complementària: veure correspondència de temes a la web
 - D. Patterson and J. L. Hennessy. “**Estructura y Diseño de Computadores: La Interfaz Hardware/ Software**”, 4a ed. Reverté, 2011.
 - ídem. "Computer Organization and Design: The Hardware/ Software Interface, 5th. ed. Morgan Kaufmann, 2013.

Presentació de EC

- Laboratori
 - 1 Sessió introductòria + 5 sessions avaluables
 - Calendari a la web

Presentació de EC

- Laboratori
 - 1 Sessió introductòria + 5 sessions avaluables
 - Calendari a la web
 - Avaluació continuada: estudi previ + treball a l'aula
 - Estudi previ: INDIVIDUAL, a presentar a l'inici de cada sessió.
Imprimiu-vos el quadern que trobareu a la web
 - Treball a l'aula: ~~per PARELLES~~ INDIVIDUAL

Presentació de EC

- Laboratori

- 1 Sessió introductòria + 5 sessions avaluables
 - Calendari a la web
- Avaluació continuada: estudi previ + treball a l'aula
 - Estudi previ: INDIVIDUAL, a presentar a l'inici de cada sessió.
Imprimiu-vos el quadern que trobareu a la web
 - Treball a l'aula: ~~per PARELLES~~ INDIVIDUAL
- Examen de Laboratori
 - INDIVIDUAL, als PCs de les aules, cap al final del curs

Presentació de EC

- Laboratori

- 1 Sessió introductòria + 5 sessions avaluables
 - Calendari a la web
- Avaluació continuada: estudi previ + treball a l'aula
 - Estudi previ: INDIVIDUAL, a presentar a l'inici de cada sessió.
Imprimiu-vos el quadern que trobareu a la web
 - Treball a l'aula: ~~per PARELLES~~ INDIVIDUAL
- Examen de Laboratori
 - INDIVIDUAL, als PCs de les aules, cap al final del curs
- Recomanació
 - Molt recomanable que us instal·leu el simulador MARS a casa
 - Feu servir la versió que hi ha a la web de l'assignatura
 - Seguiu les instruccions de la web (cal instal·lar el fitxer `startup.s`)
 - Està escrit en Java, funciona en totes les plataformes

Presentació de EC

- Avaluació

- Examen Parcial (EP)
- Examen Final (EF), inclou tots els temes 1 al 8
- Examen de Laboratori (EL)
- Avaluació Continuada de laboratori (AC)

$$\text{Nota} = 0,2 \cdot \max(\text{EP}, \text{EF}) + 0,6 \cdot \text{EF} + 0,2 \cdot (0,85 \cdot \text{EL} + 0,15 \cdot \text{AC})$$

- Competència Transversal

- Sostenibilitat i Compromís Social (SiCS)
- Nota A, B, C o D en l'expedient de l'alumne
- Breu treball escrit en grups de 3 alumnes
- S'informarà dels detalls més endavant

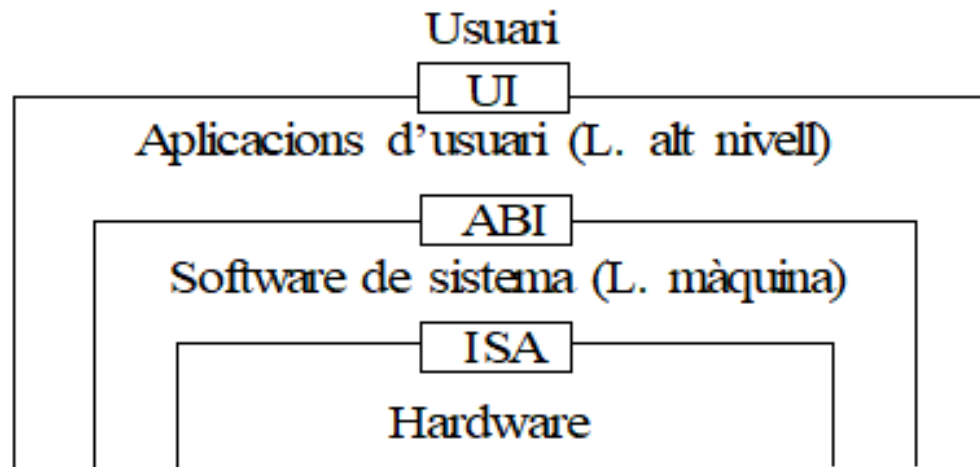
Introducció

Nivells d'abstracció del computador

- El computador com una jerarquia de nivells d'abstracció
 - **Hardware**: portes lògiques, multiplexors, biestables (IC)
 - És el que maneja un Arquitecte de Computadors
 - **Llenguatge màquina/assembler**: MIPS, x86, ARM, RISC-V (EC)
 - És el que maneja un programador de sistemes
 - **Llenguatge d'alt nivell**: C/C++, Java, Fortran, Python (Pro2)
 - És el que maneja un programador d'aplicacions
 - **Interfície d'usuari**: Menús, icones, Tallar-i-enganxar
 - És el que maneja un usuari final

Nivells d'abstracció del computador

- Cada nivell
 - Conté una **implementació** que és una "caixa negra"
 - Estableix una **interfície** amb la qual interactua el nivell superior



- L'abstracció ens ajuda a tractar la complexitat
 - Amaga els detalls de baix nivell

Nivells d'abstracció del computador

- Instruction Set Architecture (ISA)

- **Especificació** que descriu els aspectes del **processador** visibles al programador de llenguatge màquina/assembleador

- Instruccions
- Registres
- Model de memòria
- E/S
- Excepcions
- etc.

→ EC Temes 2, 3, 4, 8

- Exemples: MIPS, ARM, RISC-V, x86, PowerPC

Nivells d'abstracció del computador

- Instruction Set Architecture (ISA)

- **Especificació** que descriu els aspectes del **processador** visibles al programador de llenguatge màquina/assembleador

- Instruccions
 - Registres
 - Model de memòria
 - E/S
 - Excepcions
 - etc.

→ EC Temes 2, 3, 4, 8

- Exemples: MIPS, ARM, RISC-V, x86, PowerPC

- Una mateixa ISA admet múltiples implementacions

- p.ex. successives CPUs d'Intel i AMD implementen l'ISA x86

Nivells d'abstracció del computador

- Instruction Set Architecture (ISA)

- **Especificació** que descriu els aspectes del **processador** visibles al programador de llenguatge màquina/assembleador

- Instruccions
 - Registres
 - Model de memòria
 - E/S
 - Excepcions
 - etc.

→ EC Temes 2, 3, 4, 8

- Exemples: MIPS, ARM, RISC-V, x86, PowerPC

- Una mateixa ISA admet múltiples implementacions

- p.ex. successives CPUs d'Intel i AMD implementen l'ISA x86

- L'ISA és un *compromís* del fabricant amb el programador

- El fabricant garanteix compatibilitat de la nova implementació amb els programes escrits per a aquest ISA

Nivells d'abstracció del computador

- Application Binary Interface (ABI)
 - **Especificació** que descriu la interfície de baix nivell entre un programa i el **software de sistema**

Nivells d'abstracció del computador

- Application Binary Interface (ABI)
 - **Especificació** que descriu la interfície de baix nivell entre un programa i el **software de sistema**
- Inclou les **crides al sistema** i a **llibreries**
 - E/S de dispositius
 - Gestió de memòria i disc
 - Planificació de tasques
 - Compartició de recursos
 - Etc.

Nivells d'abstracció del computador

- Application Binary Interface (ABI)
 - **Especificació** que descriu la interfície de baix nivell entre un programa i el **software de sistema**
- Inclou les **crides al sistema** i a **llibries**
 - E/S de dispositius
 - Gestió de memòria i disc
 - Planificació de tasques
 - Compartició de recursos
 - Etc.
- Inclou també
 - **Convenis de crida i retorn** de funcions → **EC Tema 3**

Nivells d'abstracció del computador

- Application Binary Interface (ABI)
 - **Especificació** que descriu la interfície de baix nivell entre un programa i el **software de sistema**
- Inclou les **crides al sistema** i a **llibries**
 - E/S de dispositius
 - Gestió de memòria i disc
 - Planificació de tasques
 - Compartició de recursos
 - Etc.
- Inclou també
 - **Convenis de crida i retorn** de funcions → **EC Tema 3**
- Tot programa s'ha de recompilar per a cada ISA/ABI

Traducció entre nivells

- Llenguatge d'alt nivell

- Nivell d'abstracció pròxim al domini del problema
- Portable
- Productiu (ràpid d'escriure)

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```


Traducció entre nivells

- Llenguatge d'alt nivell

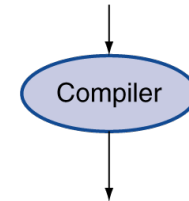
- Nivell d'abstracció pròxim al domini del problema
- Portable
- Productiu (ràpid d'escriure)

- Llenguatge ensamblador

- Llenguatge màquina en format textual "llegible"

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



Assembly
language
program
(for MIPS)

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

Traducció entre nivells

- Llenguatge d'alt nivell
 - Nivell d'abstracció pròxim al domini del problema
 - Portable
 - Productiu (ràpid d'escriure)
- Llenguatge ensamblador
 - Llenguatge màquina en format textual "llegible"
- Llenguatge màquina
 - Representació binària
 - Codifica dades i instruccions

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for MIPS)

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

Assembler

Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100
000000111110000000000000000001000
```

Compilació vs Interpretació

- Compilació

- Traducció del programa sencer tot d'una vegada (estàtic)
- Per executar-lo no es necessita ni el compilador ni el codi font
- Programa generat **molt ràpid**
- Cal recompilar-lo per a cada ISA i/o ABI, procés complex
- Exemples: C/C++, Fortran, Pascal...

Compilació vs Interpretació

- Compilació

- Traducció del programa sencer tot d'una vegada (estàtic)
- Per executar-lo no es necessita ni el compilador ni el codi font
- Programa generat **molt ràpid**
- Cal recompilar-lo per a cada ISA i/o ABI, procés complex
- Exemples: C/C++, Fortran, Pascal...

- Interpretació

- Es tradueixen les accions una per una i es van executant
- Cal tenir un intèrpret i el codi font per executar-lo
- Execució més **lenta**
- Fàcil portabilitat del codi sense modificar a qualsevol entorn
- Exemples: Java, Python...

Compilació vs Interpretació

- Compilació

- Traducció del programa sencer tot d'una vegada (estàtic)
- Per executar-lo no es necessita ni el compilador ni el codi font
- Programa generat **molt ràpid**
- Cal recompilar-lo per a cada ISA i/o ABI, procés complex
- Exemples: C/C++, Fortran, Pascal...

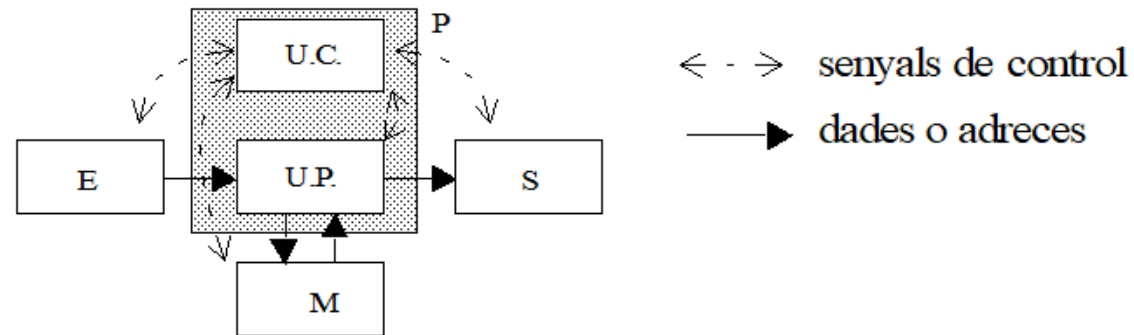
- Interpretació

- Es tradueixen les accions una per una i es van executant
- Cal tenir un intèrpret i el codi font per executar-lo
- Execució més **lenta**
- Fàcil portabilitat del codi sense modificar a qualsevol entorn
- Exemples: Java, Python...

- Els llenguatges interpretats són més productius, portables i fàcils de distribuir i d'actualitzar... però són massa lents per a aplicacions on el rendiment és crític

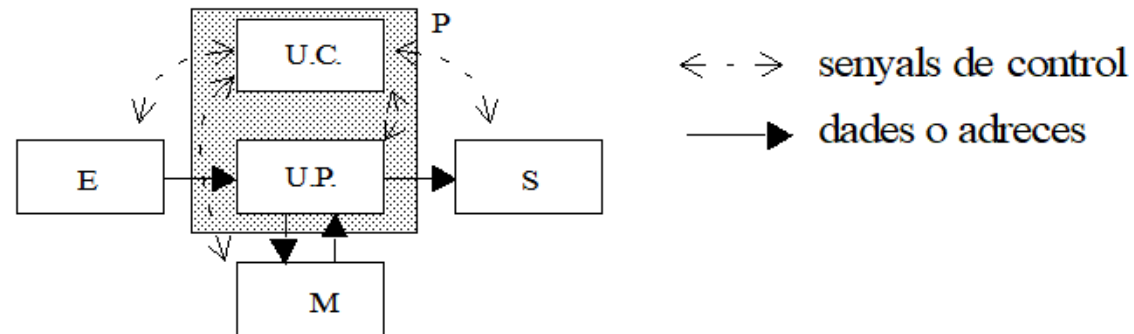
Arquitectura Von Neumann (1945)

- Concepte: Instruccions i dades en el mateix espai d'adreces de memòria



Arquitectura Von Neumann (1945)

- Concepte: Instruccions i dades en el mateix espai d'adreces de memòria



- Sovint, el model es presenta de forma simplificada:

