Tema 1. Introducció

Joan Manuel Parcerisa

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Presentació de EC

Presentació de EC

- Professor de teoria
  - Joan Manuel Parcerisa (despatx C6-116)
  - jmanel@ac.upc.edu

Presentació de EC

- Professor de teoria
  - Joan Manuel Parcerisa (despatx C6-116)
  - jmanel@ac.upc.edu
- Web d'EC
  - http://docència.ac.upc.edu/FIB/grau/EC/
    - Apunts
    - Col·lecció de Problemes
    - Enunciats, plantilles i simulador MARS per a fer pràctiques a casa
    - Exàmens anteriors
    - Calendari de sessions de laboratori

Presentació de EC

- Professor de teoria
  - Joan Manuel Parcerisa (despatx C6-116)
  - jmanel@ac.upc.edu
- Web d'EC
  - http://docència.ac.upc.edu/FIB/grau/EC/
    - Apunts
    - Col·lecció de Problemes
    - Enunciats, plantilles i simulador MARS per a fer pràctiques a casa
    - Exàmens anteriors
    - Calendari de sessions de laboratori
  - Descàrrega de documents

    Username: privatEC

    Password: Secure2010

Presentació de EC

- Professor de teoria
  - Joan Manuel Parcerisa (despatx C6-116)
  - jmanel@ac.upc.edu
- Web d'EC
  - http://docència.ac.upc.edu/FIB/grau/EC/
    - Apunts
    - Col·lecció de Problemes
    - Enunciats, plantilles i simulador MARS per a fer pràctiques a casa
    - Exàmens anteriors
    - Calendari de sessions de laboratori
  - Descàrrega de documents

    Username: privatEC

    Password: Secure2010
- Racó
  - https://raco.fib.upc.edu/

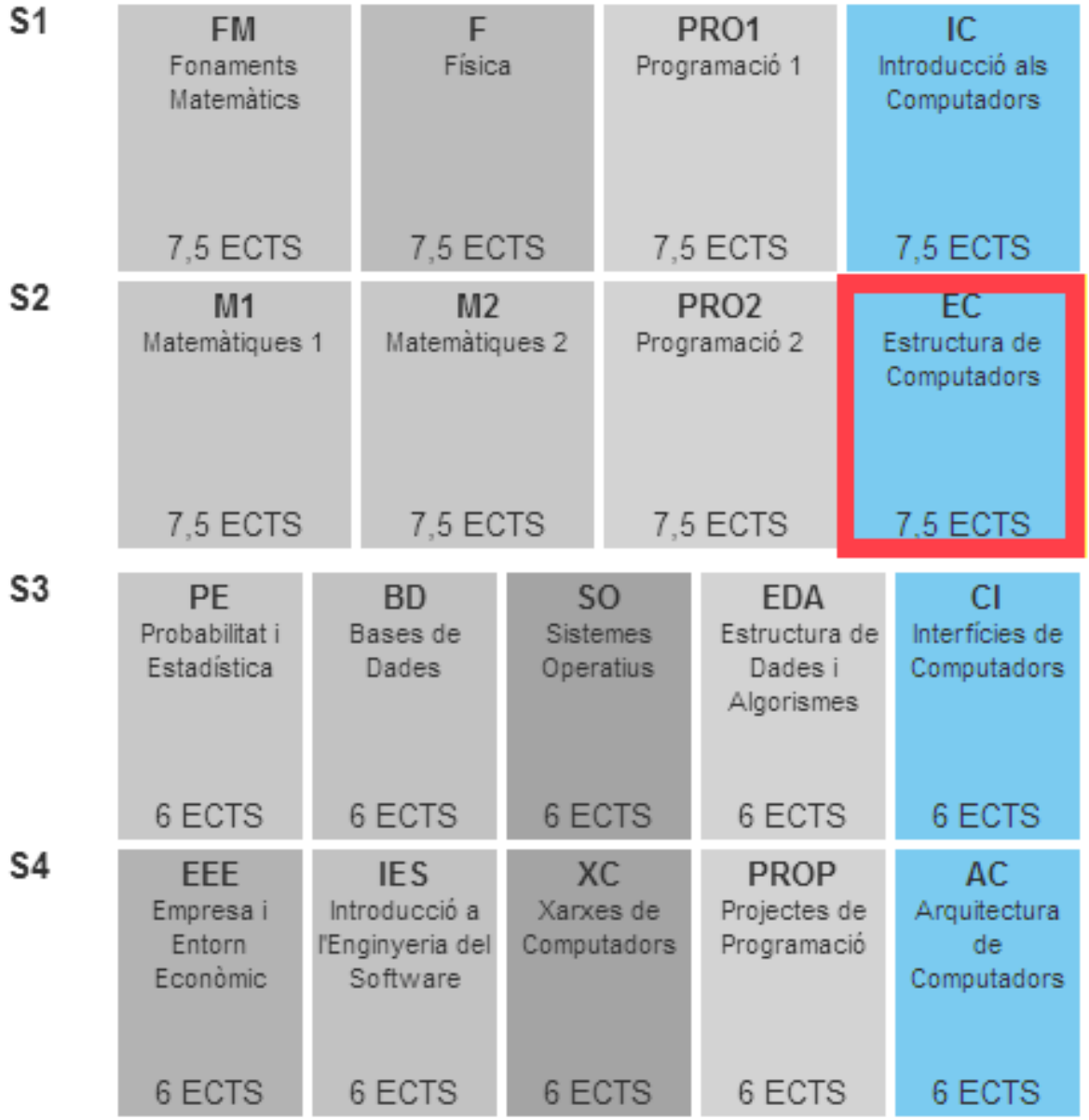
Presentació de EC

- Professor de teoria
  - Joan Manuel Parcerisa (despatx C6-116)
  - jmanel@ac.upc.edu
- Web d'EC
  - http://docència.ac.upc.edu/FIB/grau/EC/
    - Apunts
    - Col·lecció de Problemes
    - Enunciats, plantilles i simulador MARS per a fer pràctiques a casa
    - Exàmens anteriors
    - Calendari de sessions de laboratori
  - Descàrrega de documents

    Username: privatEC

    Password: Secure2010
- Racó
  - https://raco.fib.upc.edu/
- Slides d'aquest grup (teoria i problemes)
  - https://personals.ac.upc.edu/jmanel/material-ec.html

EC dins el pla d'estudis



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 (detalls a la web)
  - Bàsica: **Apunts**
  - Complementària: Llibre de text
    - D. Patterson and J. L. Hennessy. "**Estructura y Diseño de Computadores: La Interfaz Hardware/ Software**", 4a ed. Reverté, 2011.

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: verificar la validesa de l'Estudi Previ, corregint-ne els errors

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: verificar la validesa de l'Estudi Previ, corregint-ne els errors
  - 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: verificar la validesa de l'Estudi Previ, corregint-ne els errors
  - Examen de Laboratori
    - INDIVIDUAL, als PCs de les aules, cap al final del curs
  - Recomanacions
    - Instal·leu-vos 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)

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

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)

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

- Competència Transversal
  - Sostenibilitat i Compromís Social (SiCS)
  - Nota A, B, C o D en l'expedient de l'alumne
  - 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

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

Nivells d'abstracció del computador

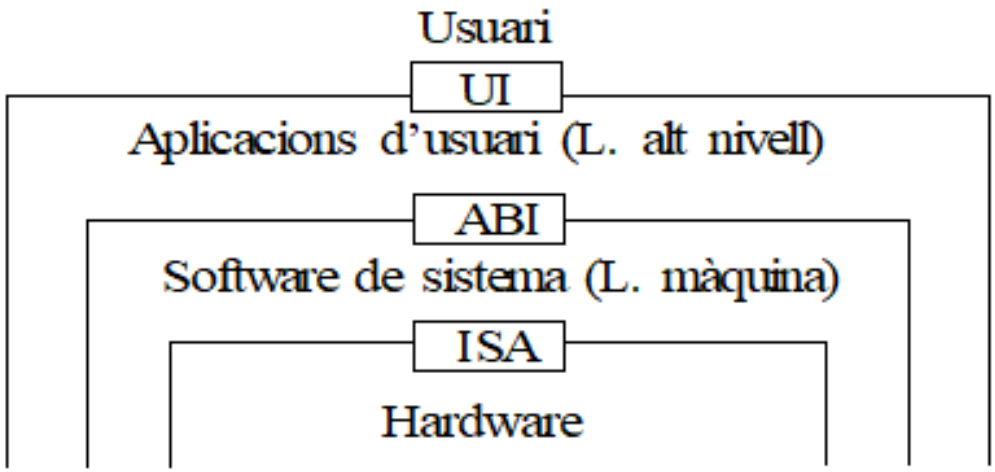
- El computador com una jerarquia de nivells d'abstracció
  - Hardware: portes lògiques, multiplexors, biestables (**IC**)
    - És el que maneja un Arquitecte de Computadors
  - Llenguatge màquina/assembler: MIPS, x86, ARM, RISC-V (**EC**)
    - És el que maneja un programador de sistemes
  - Llenguatge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

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

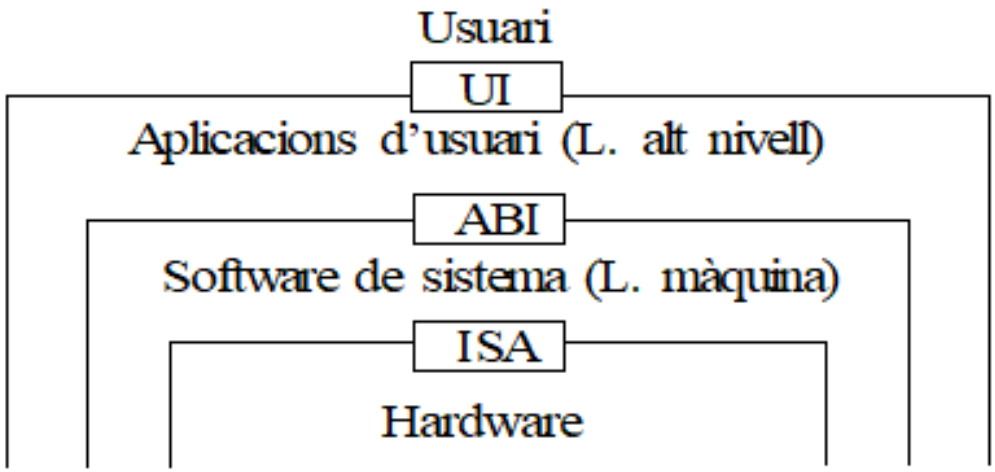
Nivells d'abstracció del computador

- Cada nivell estableix una abstracció
  - Conté una implementació
    - Amb detalls invisibles per al nivell superior ("caixa negra")



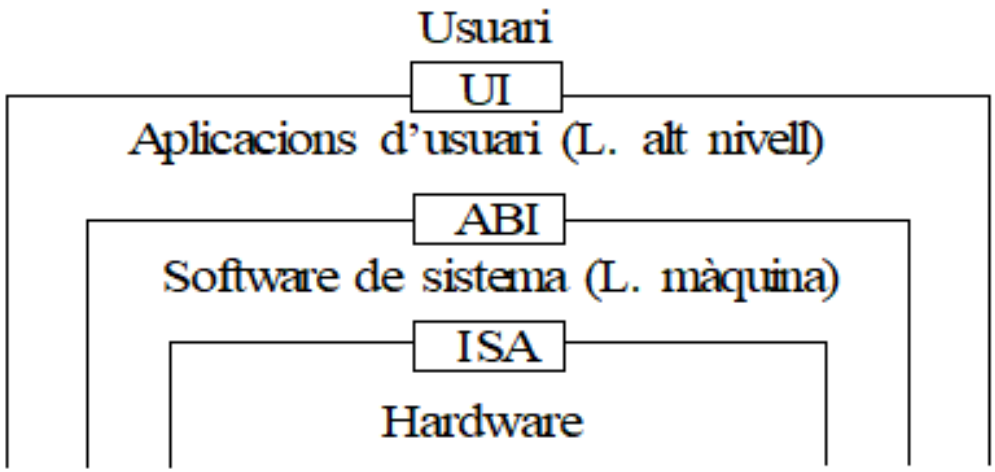
Nivells d'abstracció del computador

- Cada nivell estableix una abstracció
  - Conté una implementació
    - Amb detalls invisibles per al nivell superior ("caixa negra")
  - Estableix una interfície amb la qual interactua el nivell superior



Nivells d'abstracció del computador

- Cada nivell estableix una abstracció
  - Conté una implementació
    - Amb detalls invisibles per al nivell superior ("caixa negra")
  - Estableix una interfície amb la qual interactua el nivell superior



L'abstracció ens ajuda a tractar la complexitat

Interfícies entre nivells: ISA

- Instruction Set Architecture (ISA)
  - **Especificació** que descriu els aspectes del **processador** visibles al programador de llenguatge màquina/assemblador

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

    → **EC Temes 2, 3, 4, 8**
  - Exemples: MIPS, ARM, RISC-V, x86, PowerPC

Interfícies entre nivells: ISA

- Instruction Set Architecture (ISA)
  - **Especificació** que descriu els aspectes del **processador** visibles al programador de llenguatge màquina/assemblador

    - 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. succesives CPUs d'Intel i AMD implementen l'ISA x86

Interfícies entre nivells: ISA

- Instruction Set Architecture (ISA)
  - **Especificació** que descriu els aspectes del **processador** visibles al programador de llenguatge màquina/assemblador

    - 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. succesives CPUs d'Intel i AMD implementen l'ISA x86

- L'ISA és un *compromís* del fabricant amb el programador
  - Les successives noves implementacions mantindran compatibilitat amb els programes ja escrits per a aquest ISA

Interfícies entre nivells: ABI

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

Interfícies entre nivells: ABI

- Aplication 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.

Interfícies entre nivells: ABI

- Aplication 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.
- Inclou també
  - Convenis de crida i retorn de funcions → **EC Tema 3**

Interfícies entre nivells: ABI

- Aplication 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.
- 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)

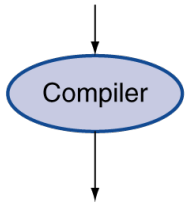
```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Traducció entre nivells

- Llenguatge d'alt nivell
  - Nivell d'abstracció pròxim al domini del problema
  - Portable
  - Productiu (ràpid d'escriure)
- Llenguatge assemblador
  - 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;
}
```

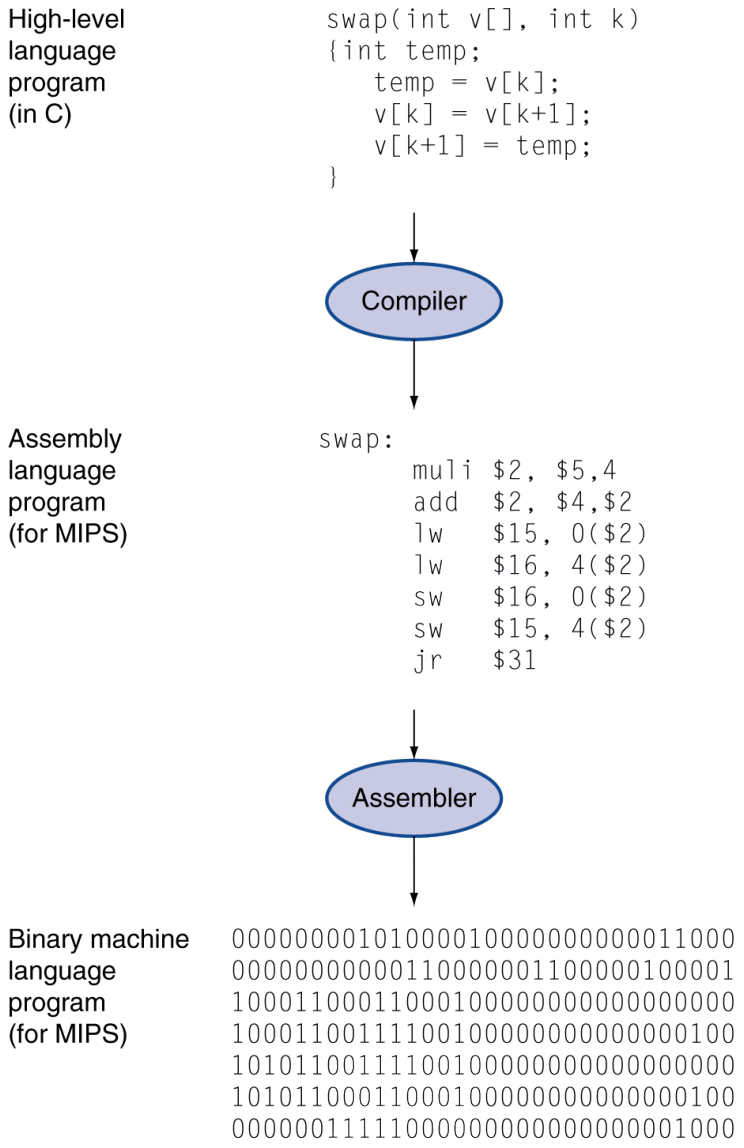


Assembly language program (for MIPS)

```
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 assemblador
  - Llenguatge màquina en format textual "llegible"
- Llenguatge màquina
  - Representació binària
  - Codifica dades i instruccions



Compilació vs Interpretació

- Compilació
  - La traducció ès estàtica: genera un *executable* tot d'un cop
    - L'executable no necessitarà ni compilador ni codi font
    - L'executable és molt ràpid
  - Exemples: C/C++, Fortran, Pascal...
  - Cal recompilar-lo per a cada ISA i/o ABI: procés complex

Compilació vs Interpretació

- Compilació
  - La traducció ès estàtica: genera un *executable* tot d'un cop
    - L'executable no necessitarà ni compilador ni codi font
    - L'executable és molt ràpid
  - Exemples: C/C++, Fortran, Pascal...
  - Cal recompilar-lo per a cada ISA i/o ABI: procés complex
- Interpretació
  - Es tradueixen les accions una per una i es van executant
    - Cal tenir un intèrpret i el codi font per executar-lo
    - L'execució és més lenta
  - Exemples: Java, Python...
  - Fàcil portabilitat del codi sense modificar a qualsevol entorn

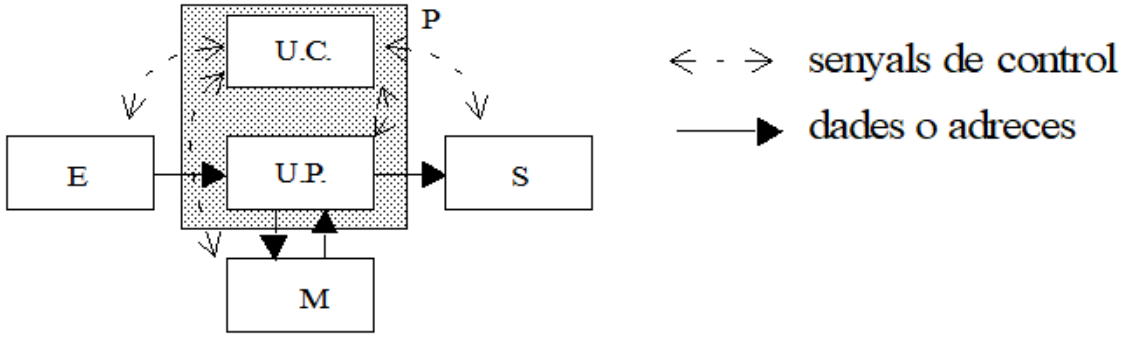
Compilació vs Interpretació

- Compilació
  - La traducció ès estàtica: genera un *executable* tot d'un cop
    - L'executable no necessitarà ni compilador ni codi font
    - L'executable és molt ràpid
  - Exemples: C/C++, Fortran, Pascal...
  - Cal recompilar-lo per a cada ISA i/o ABI: procés complex
- Interpretació
  - Es tradueixen les accions una per una i es van executant
    - Cal tenir un intèrpret i el codi font per executar-lo
    - L'execució és més lenta
  - Exemples: Java, Python...
  - Fàcil portabilitat del codi sense modificar a qualsevol entorn

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

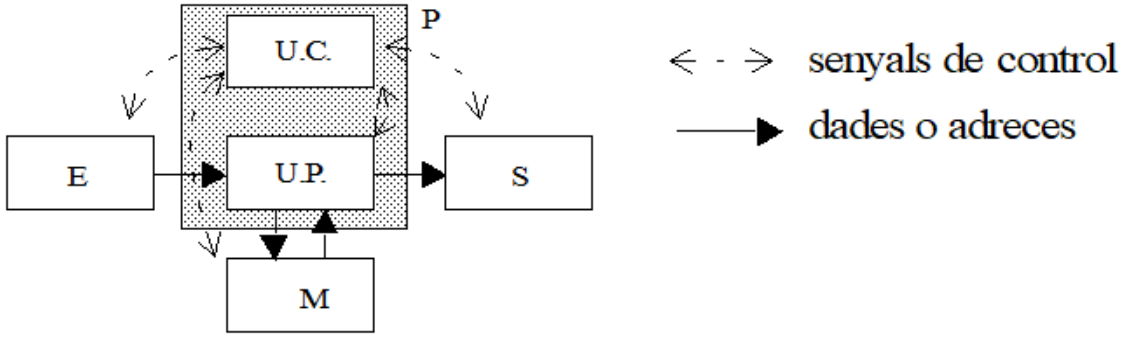
Arquitectura Von Neumann (1945)

- 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: