



IJ

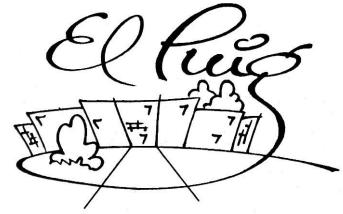
Entorns de Desenvolupament: M05

Tipologia dels llenguatges de programació

Fernando Porrino Serrano

Presentació elaborada en base al document original d'en Marcel García Vacas
Il·lustració de portada en base a imatges originals extretes de commons.wikimedia.org



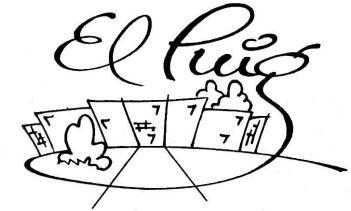


- **El concepte de llenguatge de programació**
- **Generacions de llenguatges de programació**
 - 1^a i 2^a generacions
 - 3^a generació
 - . Compilador
 - . Intèrpret
 - . Màquina virtual
 - 4^a i 5^a generacions



Concepte de llenguatge de programació

- Un llenguatge de programació es pot entendre com un **mecanisme de comunicació** entre un ésser humà i una màquina.
- Amb un llenguatge de programació es pot donar **ordres estrictes** a una màquina per a que actuï segons els nostres interessos.
- Avui dia existeixen una gran quantitat de llenguatges de programació, que es poden classificar en **generacions**.



1^a generació: llenguatge màquina

- Es donen instruccions **directament al microprocessador** fent servir codi màquina o binari.
- Un programador entrenat pot generar codi altament eficient.
- Difícil d'entendre i de mantenir, altament complex.
- **Estretament lligat el hardware.**

Program Fragment:	$Y = Y + X$	
Machine Language Code (Binary Code)		
Opcode	Address	
1100 0000	0010 0000 0000 0000	
1011 0000	0001 0000 0000 0000	
1001 0000	0010 0000 0000 0000	
Memory Cell Definitions:		
Addr.	Name	Cell Contents
1000	X	32
2000	Y	16

Exemple de codi màquina.
Font: urgenthomework.com



2^a generació: llenguatge assemblador

- Es donen instruccions **directament al microprocessador** fent servir codi assemblador.
- Cada instrucció en assemblador correspon a una instrucció en llenguatge màquina.
- Un programador entrenat pot generar codi altament eficient.
- Difícil d'entendre i de mantenir.
- Estretament lligat el hardware.

Program Fragment: Y = Y + X		
Machine Language Code (Binary Code)		
Opcode	Address	Assembly Language Code
1100 0000	0010 0000 0000 0000	LOAD Y
1011 0000	0001 0000 0000 0000	ADD X
1001 0000	0010 0000 0000 0000	STORE Y

Memory Cell Definitions:		
Addr.	Name	Cell Contents
1000	X	32
2000	Y	16

Figure 1.4: Machine and Assembly Language Program Fragment

Exemple de codi assemblador vs codi màquina.

Font: ee.hawaii.edu



2^a generació: l'assemblador

- L'assemblador es una peça de codi que s'encarrega de fer la traducció de codi ensamblador a codi màquina.
- Facilita la tasca al programador, que treballa amb instruccions més senzilles.

```
401001: 016 nop [cs:rax+rax+0x0]
401010: sub rsp, 0x38
401014: mov eax, [rip+0x35b5a6]
40101a: mov r9d, [rip+0x30521f]
401021: lea r8, [rip+0x35afe8]
401028: lea rdx, [rip+0x35afdf9]
40102f: lea rcx, [rip+0x35afce]
401036: mov [rip+0x35afe4], eax
40103c: lea rax, [rip+0x35afdd]
401043: mov [rsp+0x20], rax
401048: call dword 0x57f9e0
40104d: mov [rip+0x35afdf1], eax
401053: add rsp, 0x38
401057: ret
```

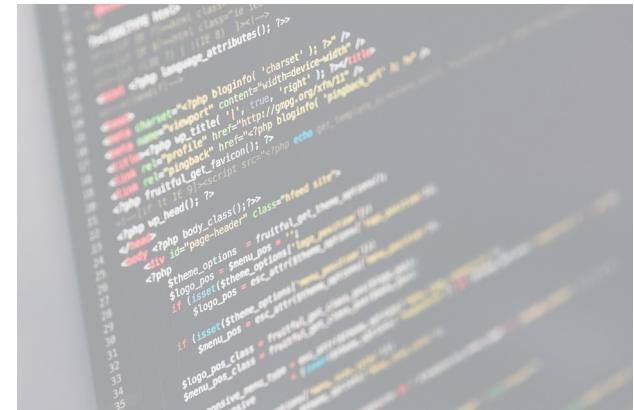
Exemple del codi font de "Snowman" descompilat.





3^a generació: llenguatges d'alt nivell

- Es fan servir **frases senzilles** per programar, resultant en un codi més lleible i entenedor.
- Es presenten **estructures de control simples** i intuïtives.
- Prioritat al manteniment per sobre del rendiment.
- Permet fer aplicacions altament grans i complexes.
- Fàcils d'aprendre.



Font: pixabay.com



3^a generació: llenguatges compilats

- **Alguns llenguatges de 3^a generació necessiten compilarse per a poder funcionar.**
- **El compilador (junt amb l'enllaçador) tradueix el codi font a codi màquina.**
- **Alguns llenguatges compilats:**
 - C
 - BASIC
 - COBOL
 - Lisp





3^a generació: llenguatges interpretats

- Alguns llenguatges **no necessiten compilarse** per funcionar.
- S'interpreten **alhora que s'executen** instrucció per instrucció.
- Aquesta funció d'interpretació la fa **l'intèrpret**.
- Els llenguatge interpretats son **més lents** que els compilats però **més portables**.
- **Alguns llenguatges interpretats:**

- Javascript

- PHP

- ASP Entorns de Desenvolupament (DAM: M05)



if(i < 9 {

Traducció

10110000 01100001



3^a generació: màquines virtuals

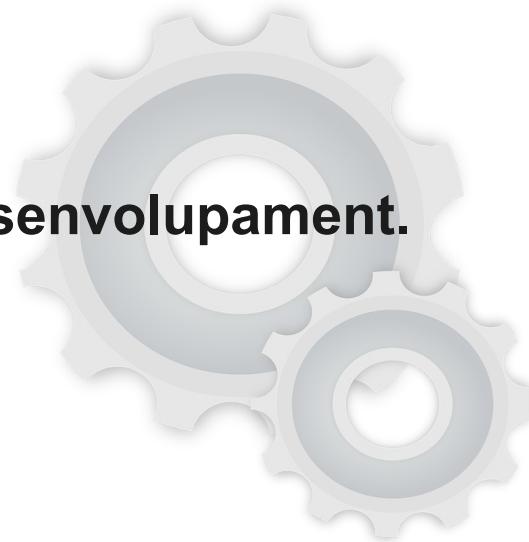
- Les màquines virtuals permeten **hibridar** els llenguatges compilats i els interpretats.
- El **compilador** transforma el codi font a codi intermedi o **bytecode**, que la màquina virtual executa.
- El codi compilat es **portable**, només es necessita la màquina virtual compatible amb la plataforma (Windows, Linux, Mac...).
- Alguns exemples:
 - Java
 - C# .NET
 - VB .NET



4^a generació: llenguatges de propòsit específic



- Orientats a treballar amb grans quantitats **d'informació**.
- S'allunyen del procediment i es centren en la **consulta, tractament i presentació de la informació**.
 - Generació d'informes.
 - Representació gràfica de fórmules matemàtiques.
 - Consultes sobre bases de dades (SQL, XML).
- Poden incloure els seus propis entorns de desenvolupament.
- **Alguns exemples:**
 - SQL
 - MATLAB
 - XQuery



Font: pixabay.com



5^a generació: llenguatges lògics (R.I.P.)

- Pensats per a treballar amb Intel·ligència Artificial.
- Han quedat desbancats per els llenguatges moderns de 3^a i 4^a generació.
- Gairebé no es fan servir, ni tan sols al camp de la I.A.
- Alguns exemples:
 - Lisp
 - Prolog



Font: pixabay.com



IJ

Gràcies per la vostra atenció!

