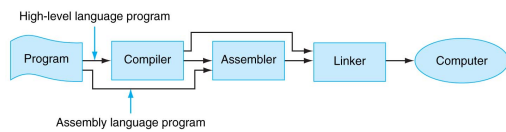


## Arquitetura de Computadores I

### O processo de assemblagem

António de Brito Ferrari  
[ferrari@ua.pt](mailto:ferrari@ua.pt)

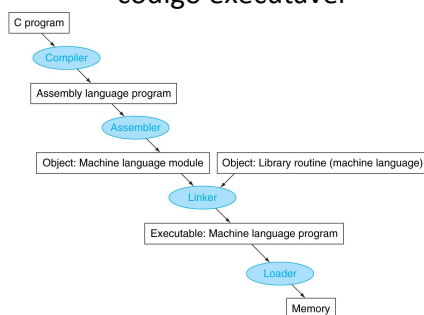


**Assembly:** diretamente escrito pelo programador ou resultado da tradução feita pelo compilador de uma linguagem de "alto nível"

ABF - AC1 - MIPS assembler

2

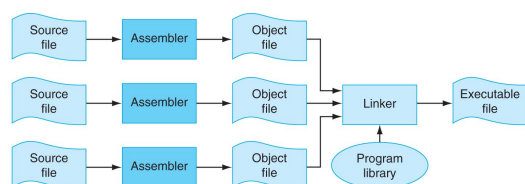
## Tradução linguagem de alto nível para código executável



ABF - AC1 - MIPS assembler

3

## O Processo de “assemblagem”



ABF - AC1 - MIPS assembler

4

## Tarefas do assembler

- Seguir Diretivas
- Traduzir Pseudoinstruções em instruções nativas
- Construir a lista de labels e endereços (***Symbol table***)
- Gerar linguagem máquina
- Criar ***Object File***

ABF - AC1 - MIPS assembler

5

## Diretivas

- As diretivas indicam ao assembler como traduzir o programa mas não geram instruções máquina
- ***.text (addr)*** : indica que o que segue é para ser colocado no segmento de texto (a partir de *addr*)
- ***.align n***: alinhar o dado que segue com 2<sup>n</sup> *byte boundary*; *align 2* ⇒ word boundary seguinte
- ***.globl sym***: declara *sym* global (*sym* pode ser referenciado noutros ficheiros)
- ***.data (addr)*** : os itens subsequentes são colocados no user data segment (a partir de *addr*)
- ***.asciiz str***: armazenar a string *str* na memória inserindo o carácter null no final

ABF - AC1 - MIPS assembler

6

## Pseudoinstruções

### Pseudoinstrução

- move \$t0, \$t1
- blt \$t0, \$t1, L1
- li \$t0, 0x12345
- add \$t1, 0x12345

### Instrução nativa

- add \$t0, \$zero, \$t1
- slt \$at, \$t0, \$t1
- bne \$at, \$zero, L1
- lui \$t0, 0x1
- ori \$t0, \$t0, 0x2345
- lui \$at, 0x1
- ori \$at, \$at, 0x2345
- add \$t1, \$t1, \$at

ABF - AC1 - MIPS assembler

7

## Assembler de duas passagens

- O assembler percorre duas vezes o texto do programa (duas passagens):

1. Constrói a *Symbol Table*

Label	Endereço

2. Gera o código máquina

- Labels
  - Local – o objeto apenas pode ser usado no ficheiro
  - Global (Externo) – pode ser referenciado por outros ficheiros para além daquele em que é definido

ABF - AC1 - MIPS assembler

8

## Programa assembly

```

.text
.align 2
.global main
main:
    subu    $sp, $sp, 32
    sw      $ra, 20($sp)
    sd      $a0, 32($sp)
    sw      $0, 24($sp)
    sw      $0, 28($sp)
loop:
    lw      $t6, 28($sp)
    mul     $t7, $t6, $t6
    lw      $t8, 24($sp)
    addu    $t9, $t8, $t7
    sw      $t9, 24($sp)
    addu    $t0, $t6, 1
    sw      $t0, 28($sp)
    ble     $t0, 100, loop
    la      $a0, str
    lw      $a1, 24($sp)
    jal     printf
    move    $v0, $0
    lw      $ra, 20($sp)
    addu    $sp, $sp, 32
    jr      $ra

```

```

.data
.align 0
str:
    .asciiz "The sum from 0 .. 100 is %d\n"

```

ABF - AC1 - MIPS assembler

9

Symbol Table

Símbolo	Endereço
main	
loop	
str	

## Código Executável

```
001001111011110111111111111100000
10101111101111110000000000010100
10101111101001000000000000100000
101011111010010100000000000100100
10101111101000000000000000011000
10101111101000000000000000001100
1000111110101110000000000001100
10001111101110000000000000011000
0000000110011100000000000011001
00100101110010000000000000000001
00101001000000010000000001100101
1010111110101000000000000001100
0000000000000000011110000010010
00000011000011111100100000100001
000101000010000011111111110111
10101111101110010000000000011000
001110000001000001000000000000
1000111110100101000000000011000
00001100000100000000000001101100
0010010010000100000001000010000
10001111101111100000000000010100
0010011110111010000000000100000
000000111100000000000000001000
00000000000000000001000000100001
```

ABF - AC1 - MIPS assembler

10

## Formato do ficheiro objeto (UNIX)

Object file header	Text segment	Data segment	Relocation information	Symbol table	Debugging information
--------------------	--------------	--------------	------------------------	--------------	-----------------------

**Object File Header** – descreve o tamanho e posição dos outros campos do ficheiro objeto

**Text segment** – contem o código máquina das rotinas no ficheiro fonte.  
Este código pode não ser executável devido a referências externas.

**Data segment** – contem a representação em binário dos dados no ficheiro fonte.  
Os dados podem estar incompletos devido a referências externas.

**Relocation information** – identifica as instruções e dados que dependem dos endereços absolutos quando o programa é carregado em memória.  
Esta informação é necessária porque o assembler não sabe que posições de memória o código e os dados vão ocupar quando forem ligados com o resto do programa.

**Symbol table** – os labels que ainda não estão definidos (external references,...) e aqueles que estando definidos são globais (i.e. referenciados noutros ficheiros objeto)

**Debugging information** – informação que permite associar instruções máquina com as instruções no ficheiro fonte (source file)

ABF - AC1 - MIPS assembler

11

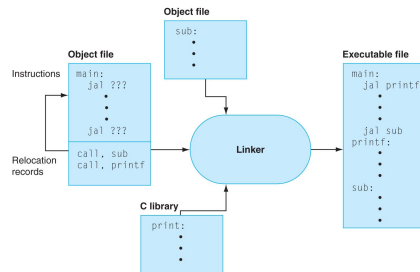
## Link editor ou linker

- Pega em diferentes programas objeto (p.ex. Programa e rotinas da biblioteca que invoca) e liga-os
- 3 passos de processamento:
  1. Procurar nas bibliotecas as rotinas usadas pelo programa
  2. Determinar as posições de memória que o código de cada módulo ocupará e relocalizar as respetivas instruções ajustando as referências absolutas
  3. Resolver as referências entre ficheiros

ABF - AC1 - MIPS assembler

12

## O *linker*: resolução das referências externas



ABF - AC1 - MIPS assembler

13

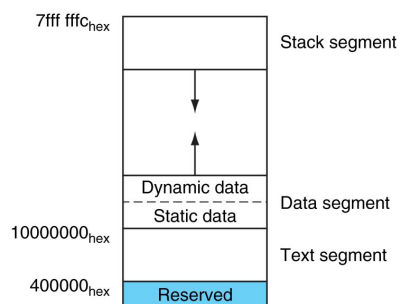
## Loader (UNIX)

1. Lê o header do ficheiro executável para determinar o tamanho dos segmentos de texto e dados
2. Cria um espaço de endereçamento para o programa (texto + dados + stack)
3. Copia instruções e dados para o espaço de endereçamento atribuído
4. Copia os argumentos passados ao programa para o stack
5. Inicializa os registos (\$sp = topo do stack)
6. Salta para a rotina de inicialização que copia os argumentos do stack para os registos e invoca o programa. Quando este termina e retorna a rotina de inicialização termina o programa com a system call *exit*

ABF - AC1 - MIPS assembler

14

## Mapa da memória



ABF - AC1 - MIPS assembler

15