

# Arquitetura e Organização de Computadores

## Mips: Modos de Endereçamento e Ponteiros

**Prof. André D'Amato**

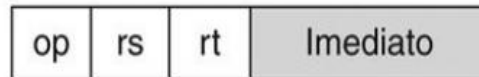
[andredamato@utfpr.edu.br](mailto:andredamato@utfpr.edu.br)

# Modos de Endereçamento no MIPS

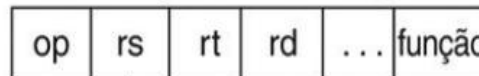
- O MIPS apresenta os seguintes modos de endereçamento:
  - Endereçamento Imediato;
  - Endereçamento em registrador;
  - Endereçamento de base ou deslocamento;
  - Endereçamento relativo a PC;
  - Endereçamento pseudodireto.

# Modos de Endereçamento no MIPS

## 1. Endereçamento imediato



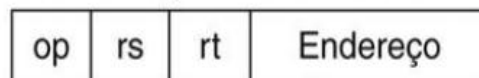
## 2. Endereçamento em registrador



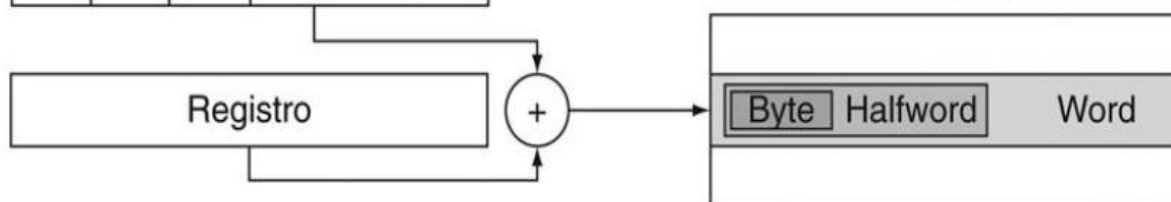
Registradores

Registro

## 3. Endereçamento de base



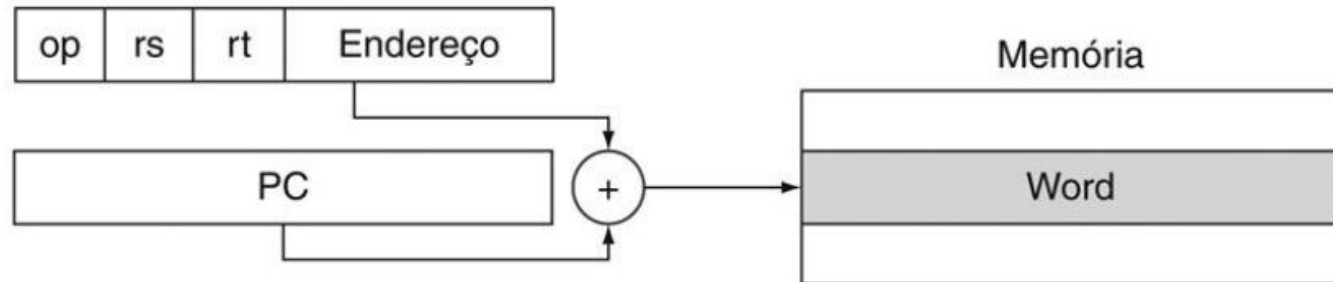
Memória



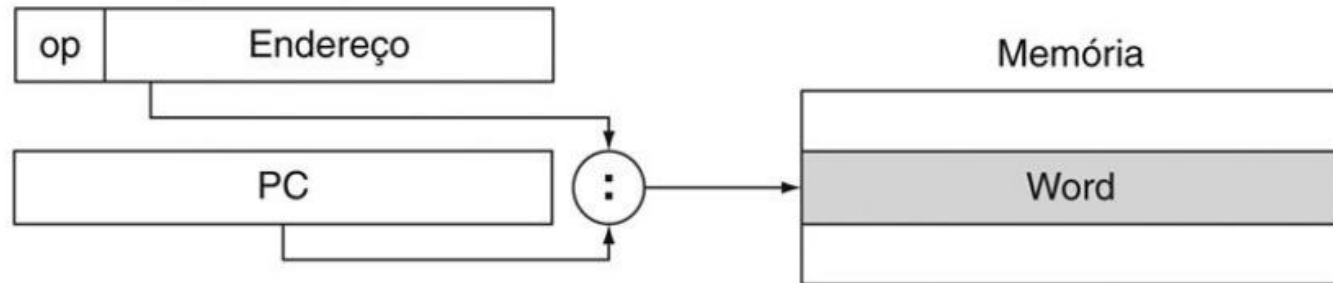
Modos de Endereçamento do MIPS

# Modos de Endereçamento no MIPS

## 4. Endereçamento relativo ao PC



## 5. Endereçamento pseudodireto



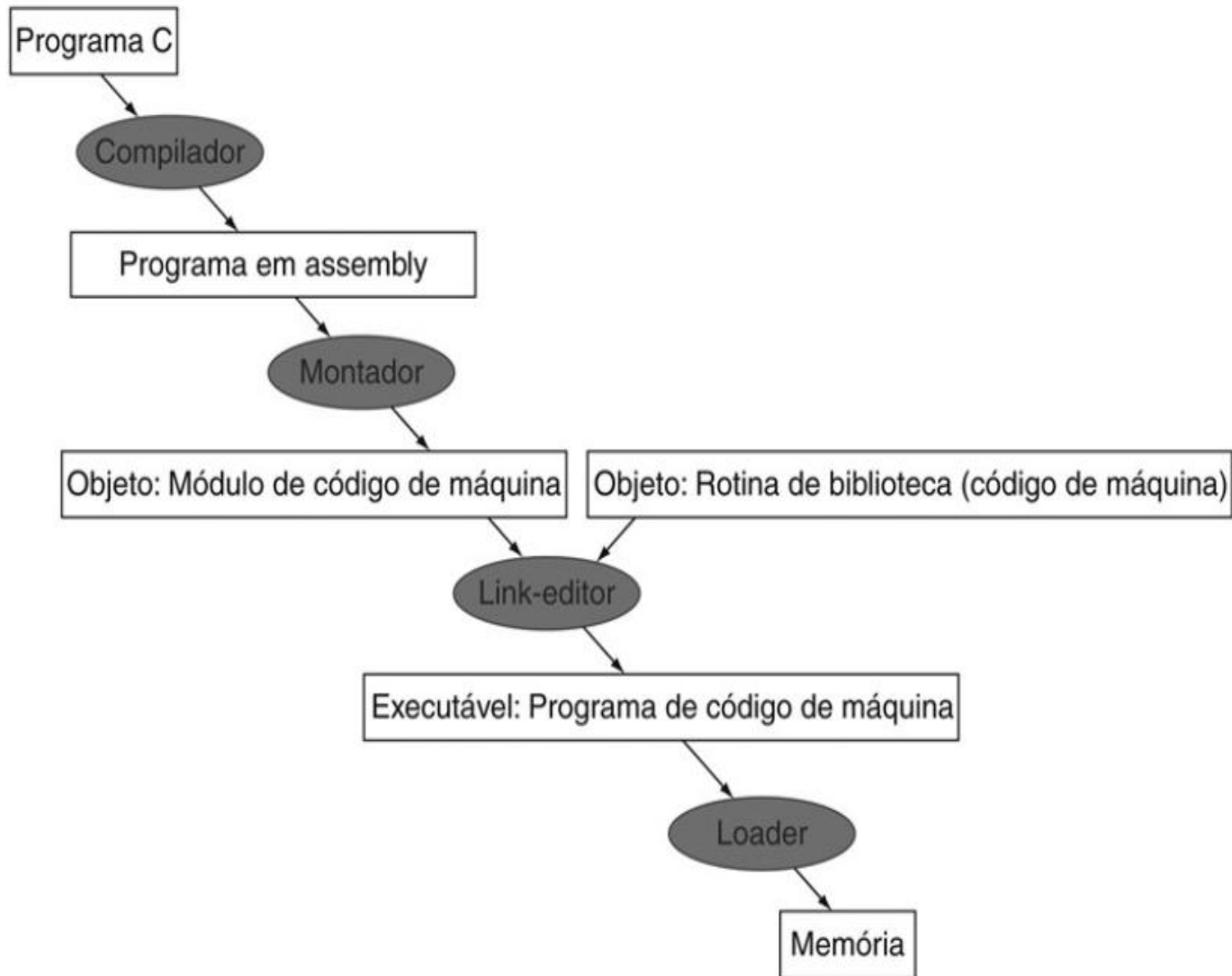
Modos de Endereçamento do MIPS



# Traduzindo e iniciando programas

- **Compilador**
  - Código C em assembly
- **Montador**
  - Assembly para código objeto
    - Endereços relativos para as labels
- **Linker**
  - Objeto em executável
    - Endereços finais com deslocamentos devidamente calculados a partir dos vários módulos do programa
- **Loader**
  - Carrega o programa na memória principal

# Traduzindo e iniciando programas



# Arquivo objeto

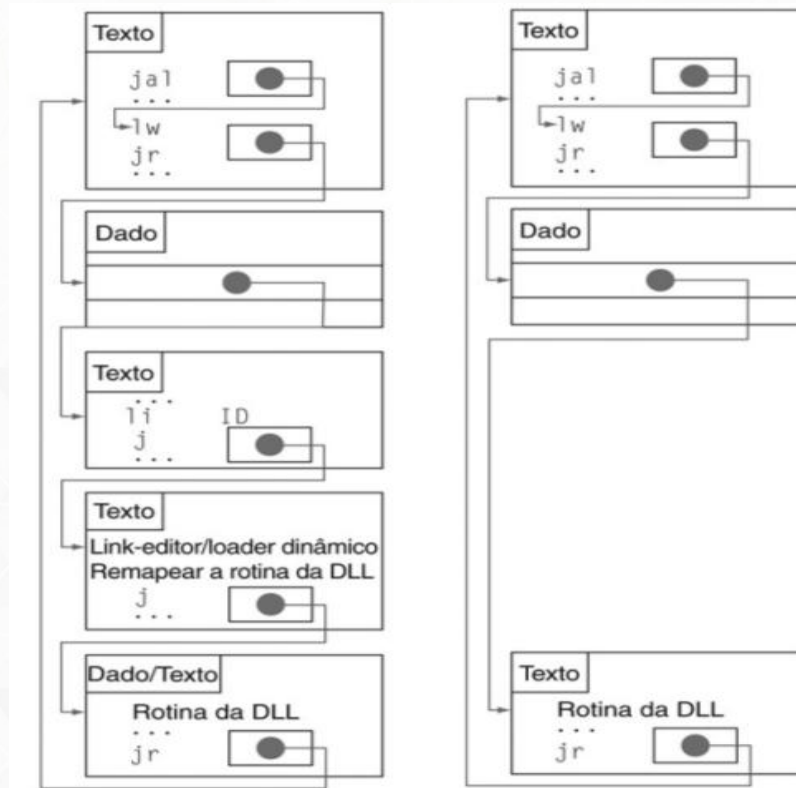
- Arquivo Objeto Unix
  - Cabeçalho;
  - Seguimento de texto;
  - Seguimento de dados estáticos;
  - Informações de relocação;
  - Tabela de símbolos;
  - Informações de depuração.
- Para vincular os arquivos objetos um linker:
  - Aloca simbolicamente na memória os seguimentos de dados e código;
  - Determina os endereços das labels de dados e código
  - “Junta” as referências externas e internas

# Loader

- Lê cabeçalho do executável para saber tamanho dos segmentos de texto e dados;
- Cria espaço de endereçamento necessário;
- Copia os segmentos de texto e dados para a memória;
- Copia os parâmetros de programa para a pilha;
- Define o \$sp para a primeira célula livre;
- “Pula” para rotina de inicialização, copia parâmetros para os regs de argumentos e desvia para o ponto de entrada do programa. A rotina termina com uma chamada EXIT.



# DLLs (Dinamicamente Linked Libraries)



**DLL por meio de link-edição tardio**

# Arrays vs Ponteiros

- Exemplo de uma rotina para zerar uma sequência de palavras na memória utilizando **ponteiro**

```
void clear(int *array, int size){  
    int *p;  
    for (p = &array[0]; p < &array[size]; p = p + 1){  
        *p = 0;  
    }
```

# Arrays vs Ponteiros

- Exemplo de uma rotina para zerar uma sequência de palavras na memória utilizando **array**

```
void clear(int array[], int size){  
    int i;  
    for (i = 0; i < size; i++)  
        array[i] = 0;  
}
```

# Referências

- PATTERSON, David A.; HENNESSY, John L. Organização e projeto de computadores: a interface hardware/software, 5. ed. Rio de Janeiro: Elsevier, 2017. ISBN 9788535287936.
- STALLINGS, William.; Arquitetura e Organização de Computadores. 10. ed. São Paulo: Pearson Education do Brasil, 2017. ISBN 9788543020532.
- TANENBAUM, A. S.; AUSTIN, T. Organização Estruturada de Computadores. 6. ed. São Paulo: Pearson Education do Brasil, 2013. ISBN 9788581435398.