

**UNIVERSIDADE ESTADUAL DO RIO GRANDE DO  
SUL**

**UNIDADE GUAÍBA**

**Programação de Sistemas**  
**Ligadores/Carregadores**

**Celso Maciel da Costa**

Guaíba, Novembro 2013.

# Sumário

- Carregadores
- Carregadores absolutos
- Carregadores Gerais

# Loaders

- Um loader é um programa do sistema que executa a função de carga.
  - **Alguns também suportam relocação e ligação.**
  - **Outros possuem um ligador e um carregador separado.**
- Funções básicas
  - **Levar um programa objeto para a memória.**
  - **Começar sua execução.**

# Input

- Programa objeto:
  - **Contém instruções e dados do programa fonte.**
  - **Especifica os endereços de memória onde esses itens deverão ser carregados.**

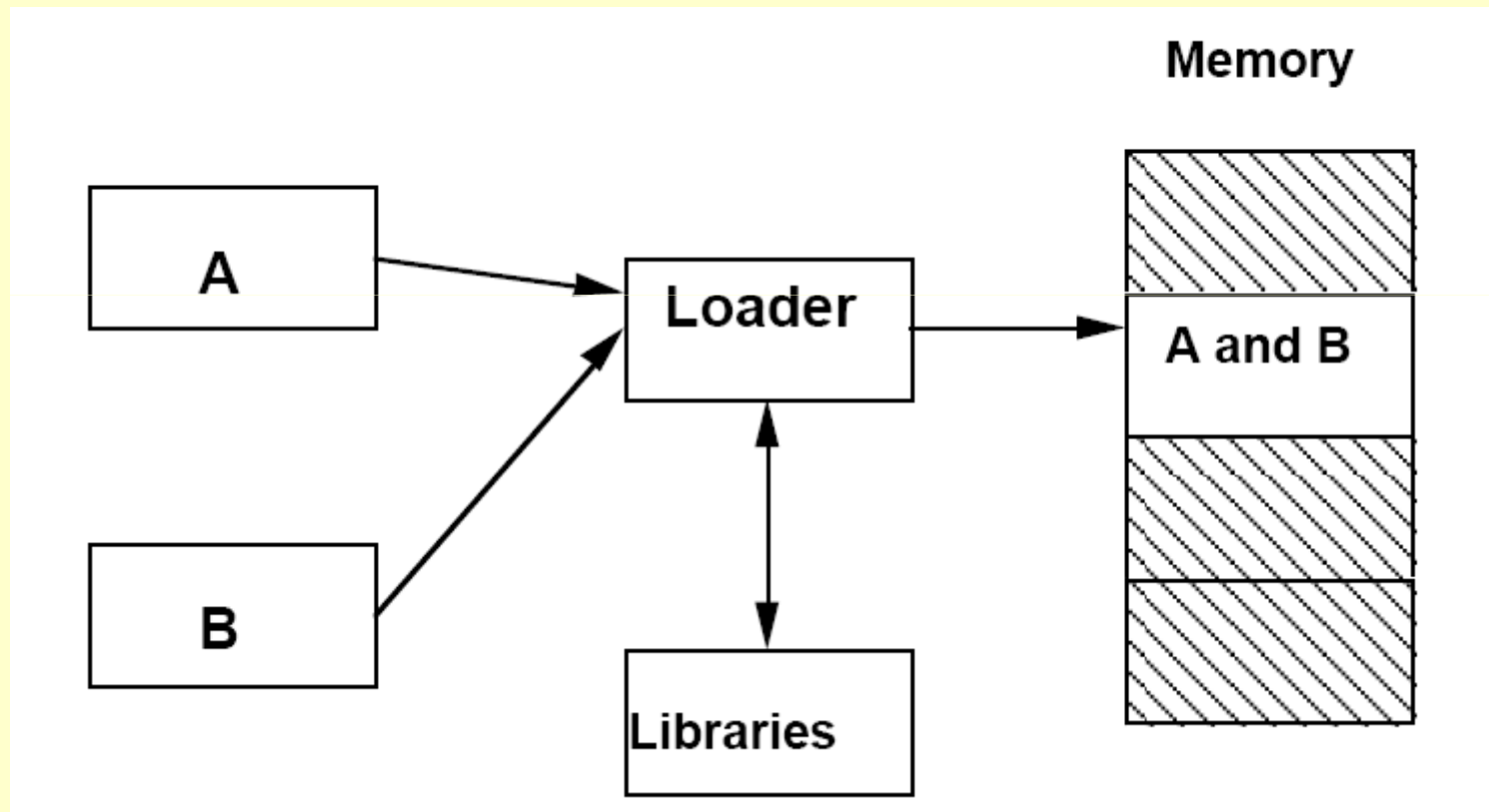
# Funções básicas

- **Alocação:** alocar espaço na memória para os programas.
- **Ligação:** Resolver referências simbólicas entre programas objetos.
  - **Combinar dois ou mais programas objetos separados.**
  - **Fornecer informações necessárias para permitir referências entre eles.**

# Funções Básicas

- **Relocação:** Ajustar todos os endereços dependentes de localização, tais como endereços de constantes, de maneira a corresponder ao endereço alocado.
  - **Modifica o programa objeto de maneira a que possa ser carregado em endereços diferentes da locação originalmente especificada.**
- **Carga:** Colocar as instruções de máquina e os dados na memória física.

# Funções básicas



# Projeto de um Carregador Absoluto

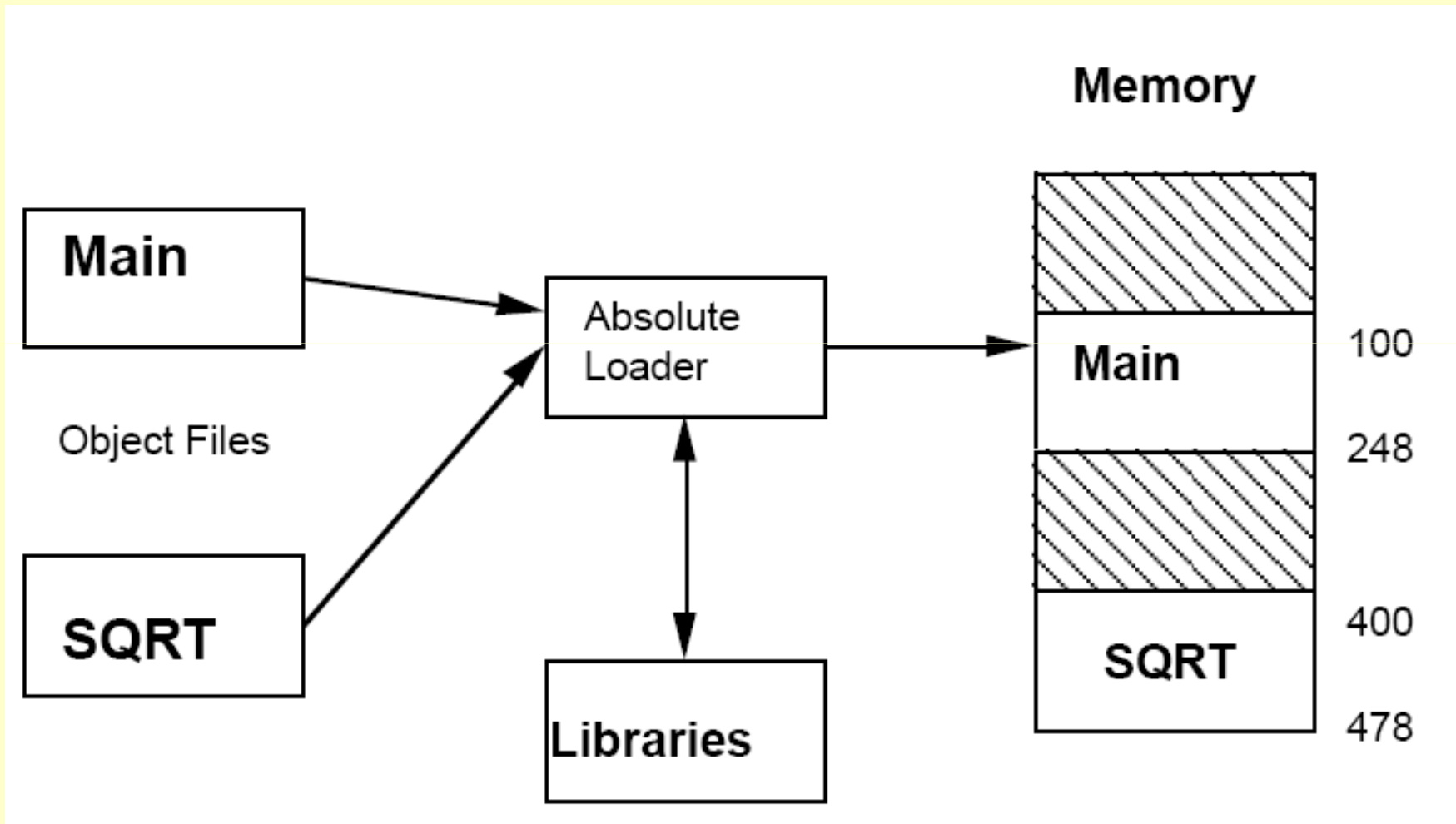
- Sua operação é muito simples
  - **Sem ligação ou relocação**
- Operação em um passo
  - Verificar o header (H record) para verificar se trata-se de um programa executável.
  - Ler cada tipo de registro T record (código e dados), e mover cada objeto para o endereço de memória indicado.
  - No final da leitura (E record), saltar para o endereço especificado para começar a execução do programa carregado.



# Modelos de Loader

- **Compile and Go**
  - O montador roda em uma parte da memória.
  - Coloca as instruções de máquina (traduzidas) e os dados diretamente nas posições de memória alocadas.
  - Quando a tradução termina, o montador transfere a execução para a primeira instrução do programa traduzido.

# Loader Absoluto



# Desvantagens

- Uma porção de memória é perdida pois a memória ocupada pelo montador não está disponível para programas objetos.
- É necessário retraduzir os programas de usuários toda vez que for executar.
- É muito difícil tratar múltiplos segmentos, especialmente se os dados estão em programas fontes diferentes.

# Desvantagens

- Se ocorrer aumento no tamanho do MAIN, por exemplo, em 300 bytes.
  - **O final do MAIN ( $100 + 300 = 400$ ) irá sobrepor o início de SQRT (400)**
  - **Será necessário colocar SQRT em um novo endereço.**
- Além disso, será necessário que todas as outras subrotinas que se referem ao endereço de SQRT sejam modificadas.

# Boot Loader

- Automaticamente executado quando o computador é ligado.
- Executa o primeiro código (usualmente a execução começa no endereço zero de memória)
  - **Carrega o OS a partir do endereço 80**
  - **Após a carga de todo o código, boot salta para o endereço 80 (ou o endereço de início do main do OS)**

# Desvantagens dos Carregadores Absolutos

- Endereço de carga deve ser especificado.
- O programador deve ter cuidado em não atribuir duas funções ao mesmo endereço ou em endereços sobrepostos.
- Difícil utilização de bibliotecas eficientemente.
  - **É importante selecionar e carregar as bibliotecas quando necessárias.**

# Desvantagens dos Carregadores Absolutos

- Alocação - pelo programador
- Ligação - pelo programador
- Relocação – Não existente. Carregado onde o montador designar.
- Transferência para a memória - realizada pelo carregador.

# Carregador Geral

- Ligação
- Relocação
- Carga



# Ligação de Subrotinas

- O main do programa A deseja executar o subprograma B.
- O programador, no programa A, poderia escrever uma instrução de transferencia (e g, BSR B) para o subprograma B.
- O montador não conhece o valor desse símbolo (B) e irá gerar um erro.

# Externals e Global

- A pseudo-instrução EXTERN, seguida de uma lista de símbolos, indica que os símbolos são definidos em outros programas, mas referenciados no presente programa
- Se o símbolo é definido no programa e referenciado em outros:
  - **Inserir o símbolo em uma lista, seguindo a pseudo-instrução GLOBAL.**

```
MAIN  ORG  $10
      EXTERN SUBROUT
      BSR   SUBROUT
DONE  HLT
```

# Relocação

- Carregador Relocador or Carregador Relativo:
- Dois métodos para especificar a relocação como parte do programa objeto:
- **1. Registro de Modificação**
  - **Descreve cada parte do código objeto que deve ser trocada quando o programa é relocado.**

# Segundo Método

- **2. Máscara de Bit's:** Um bit/byte de relocação associado com cada palavra do código objeto.
  - **A para Absoluto: não necessita modificação.**
  - **R para Relativo: necessita relocação.**
  - **X for external.**
- Exemplo
- T0010A6119AFE00A4003A0E01R

# Ligador Carregador de Dois Passos

- Passo 1
  - Aloca e atribui a cada programa endereços de memória.
  - Cria a tabela de símbolos, preenchendo os valores dos símbolos externos.
- Passo 2
  - Carrega o código do programa.
  - Executa modificações de relocação nos endereços que necessitam ser alterados.
  - Resolve referências externas. (ligação).

# Estruturas de Dados

- Tabela de Símbolos Externos (EXTAB)
  - Armazena o nome e o endereço de cada símbolo externo definidos no conjunto de programas sendo carregados.
  - Indica em qual programa o símbolo é definido.
  - Uma tabel hash é geralmente usada.
- Endereço de Carga do Programa (PROGADDR)
  - Endereço inicial de memória onde o programa ligado é para ser carregado.
  - Fornecido pelo O/S

# Estruturas de Dados

- Control Section Address (CSADDR)
  - **Endereço inicial atribuído ao CS correntemente sendo analisado pelo carregador.**
  - **Seu valor é adicionado a todo endereço relativo, para convertê-lo no endereço atual.**



# Algoritmo

- Passo 1:
  - **PROGADDR** é obtido do O/S.
  - **CSADDR** é setado com esse endereço
  - **Todos os símbolos externos são incluídos na Tabela de Símbolos Externos (EXTAB)**
    - » Os endereços dos símbolos externos são obtidos adicionando o valor do CSADDR
  - **O endereço de cada CS são determinados:**  
$$\text{CSADDR} = \text{CSADDR} + \text{CSLEN}$$

# EXTAB

Program/CS	Symbol	Address	Length
Test		0040	0046
	EXE	0060	
ProgA		0086	0010
	LISTA	0090	
ProgB		0096	6

## Passo II

- Faz a carga, relocação e ligação.
- Cada registro de Texto é lido:
  - O código objeto é movido para o endereço especificado (mais o valor corrente de CSADDR)
  - Quando “R” é encontrado, o valor é adicionado ao valor da locação de memória indicada.
  - Quando “X” is encontrado o símbolo de EXTAB é resolvido.
  - Último passo: transfere o controle para o programa carregado para começar a execução.