

PCS-2302 / PCS-2024 Lab. de Fundamentos de Eng. de Computação

Aula 09 **Montador Relocável**

Professores:

Jaime Simão Sichman (PCS 2302) Ricardo Luis de Azevedo da Rocha (PCS 2024)



Roteiro

- 1. Necessidade de programas relocáveis
- 2. Implicações na linguagem simbólica
 - Novas pseudo-instruções
 - · Novo formato de instrução
- 3. Montador relocável
 - · Diagrama de classes
 - Exemplo de funcionamento
- 4. Parte Experimental
 - Implementação de um montador relocável para o simulador MVN



equipe

- Exigem gerência cuidadosa das áreas de memória ocupadas e dos endereços de cada parte do Toda vez que um programa é modificado, pode ser necessário recodificá-lo parcial ou totalmente

Programas absolutos são executáveis

- Se a área ocupada pelo novo código for maior que a antiga, é preciso alojar o programa em outra parte da memória

Necessidade de Programas Relocáveis (1)

estritamente nas posições de memória em que

Tornam difícil a manutenção e o trabalho em

TST

Necessidade de Programas Relocáveis (2)



- Programas relocáveis permitem sua execução em qualquer posição de memória
 - As referências à memória devem ser previamente ajustadas
 - Um gerenciador da ocupação da memória deve ser utilizado
- Tornam possível utilizar partes de código projetadas externamente
- Uso de bibliotecas
- Exigem que se possa montar parcialmente um programa, sem todos os endereços resolvidos!

1..2 out 2007

Implicações na linguagem simbólica

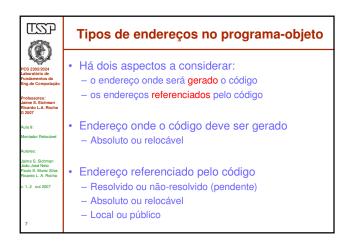
- Para que se possa exprimir um programa relocável e com possibilidade de construção em módulos, separadamente desenvolvidos, é necessário que:
 - Haja a possibilidade de representar e identificar endereços absolutos e endereços relativos
 - Um programa possa ser montado sem que os seus endereços simbólicos estejam todos resolvidos
 - Seja possível identificar, em um módulo, símbolos que possam ser referenciados simbolicamente em outros módulos

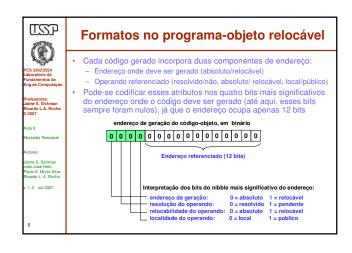
TST

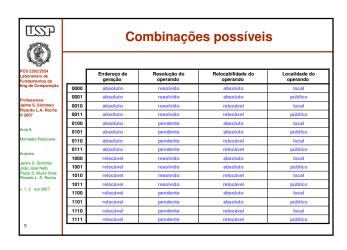
1..2 out 2007

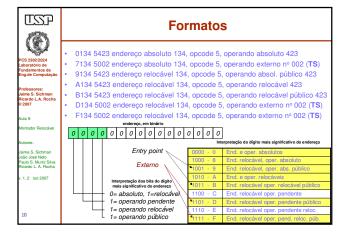
Implicações no montador

- No montador, tornam-se necessários:
- endereços relativos uma pseudo-instrução especial deve indicar que se trata de origem relativa
- importar símbolos para que um símbolo X de outro programa possa ser referenciado no programa
- exportar símbolos para que um ponto X do programa possa ser referenciado em outros
- anexar, ao final da montagem, todos os símbolos não-resolvidos ao programa-objeto, para que essa informação possa ser passada posteriormente ao programa ligador (linker).
- Gerar código-objeto no formato compatível com o loader hexadecimal (função P do simulador MVN)

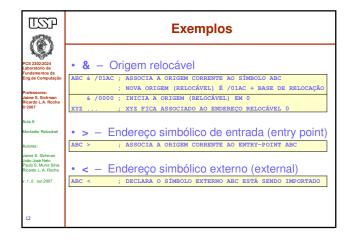














Alterações complementares

- Para atingir toda a sua funcionalidade, as seguintes adições posteriores serão necessárias:
 - Geração de código-objeto no novo formato, incluindo:
 - Operando simbólico
 - Endereços simbólicos de entrada e externos
 - · Outras referências simbólicas não-resolvidas
 - Alteração do dumper hexadecimal: incluir referências simbólicas
 - Algoritmo de relocação a partir de uma base estabelecida
 - Alteração do loader hexadecimal: incluir relocação

