

Capítulo 14: Conceitos básicos

1. Explique em que consiste a resolução de endereços nos seguintes momentos: codificação, compilação, ligação, carga e execução.

Resposta:

Codificação: Nesse momento, a atribuição de endereços de cada variável e do programa é realizada pelo próprio programador. Entretanto, essa abordagem só é utilizada programando em Assembly.

Compilação: Os endereços das variáveis na memória é definido pelo compilador durante a tradução do código-fonte. Para a execução desse processo, o compilador deve ter acesso a todos os códigos-fonte no ato da compilação, evitando assim conflitos entre os endereços atribuídos as variáveis.

Ligação: Na atribuição de endereços na ligação, o ligador (linker) atribui os endereços de memória com base na tabela de símbolos gerada pelo compilador. A tabela de símbolos possui a descrição das variáveis, das funções utilizadas, dos tipos, dos locais de definição e dos locais de utilização.

Carga: A atribuição de endereços durante a carga é um processo utilizando na carga de bibliotecas dinâmicas, onde o carregador (loader) é responsável por carregar o código do processo na memória como também atribuir os endereços a serem utilizados.

Execução: Na atribuição de endereços durante a execução, os endereços são emitidos pelo processador em tempo de execução do processo, sendo esses endereços emitidos convertidos para valores efetivos de endereço na memória.

2.Como é organizado o espaço de memória de um processo?

Resposta:

O espaço de memória de um processo é organizado em seções, sendo estas:

TEXT: Seção da memória responsável por armazenar o código binário que será executado pelo processo. Sendo alocado no início do espaço de endereçamento, esta seção tem tamanho fixo e permite acesso de leitura e execução.

DATA: Seção da memória responsável por armazenar as variáveis estáticas inicializadas. Os valores dessas variáveis são armazenados no arquivo do programa, sendo carregado quando o processo começa. As variáveis estáticas podem ser tanto globais como locais.

BSS: O BSS (Block Started by Symbol) é a seção da memória responsável por armazenar as variáveis estáticas não inicializadas. A separação das variáveis estáticas inicializadas e não-inicializadas ocorre devido ao fato das inicializadas serem armazenadas no arquivo executável, sendo carregadas logo no começo do processo.

HEAP: Seção da memória responsável por armazenar as variáveis alocadas dinamicamente. Essa seção é finalizada pelo ponteiro Program Break e pode ter seu tamanho alterado através de chamadas de sistema.

STACK: A Stack (Pilha) é a seção responsável por armazenar o fluxo de execução nas chamadas de função, os parâmetros, o valor de retorno das funções e as variáveis locais. Uma particularidade dessa seção é que ela inicia em um endereço maior e cresce para baixo, alocando novos endereços em endereços menores.