

PCS2056 – Linguagens e Compiladores

Assunto: Especificação do ambiente de execução.

Enunciado da **parte 4** do trabalho: Especificação do ambiente de execução da linguagem de programação definida

Data de entrega: **11 de novembro de 2008**

Objetivo: Estudo do ambiente de execução de uma linguagem imperativa.

Palavras-chave:

Preparação do ambiente

Sistemas de programação

Máquina virtual

Chamadas de rotinas (call/return)

Uso de pilha

Inicialização de variáveis/constantes

Alocação de espaço para o programa: código, dados, pilha e *heap*

Definição do processo de chamada e retorno de sub-rotinas da linguagem

Rotinas da biblioteca: Entrada/saída, acesso ao SO

Programas de sistema: carga programas (loader), programa montador, relocador, editor de ligações

Questões:

- 1) Explique como deve ser efetuada a sequência de chamada e retorno de uma sub-rotina:
 - a) Em que momento é possível determinar o espaço necessário a ser alocado para um registro de ativação?
 - b) Como calcular os endereços dos parâmetros e variáveis locais?
 - c) Como criar um registro de ativação?
 - d) O que deve ser armazenado para permitir o controle da chamada/retorno dentro do registro de ativação?
- 2) Deduza uma fórmula geral para calcular, em relação ao endereço do apontador de registro de ativação, os endereços virtuais dos parâmetros, cujos elementos possam ser bytes, palavras de 32 bits ou agrupamentos de 64 bits conforme o tipo do elemento (char, int ou double, respectivamente).
- 3) Deduza, de forma semelhante à efetuada no exercício anterior, uma fórmula geral para calcular os endereços virtuais das variáveis locais, cujos elementos possam ser bytes, palavras de 32 bits ou agrupamentos de 64 bits conforme o tipo do elemento (char, int ou double, respectivamente).
- 4) Para o caso de sub-rotinas aninhadas, explique como acessar variáveis locais do bloco aninhador dentro do bloco aninhado.
- 5) Deduza uma fórmula geral para calcular, em relação ao endereço do apontador de registro de ativação, os endereços virtuais das variáveis locais do bloco aninhador. Onde armazenar a informação do apontador do registro de ativação do bloco aninhador?
- 6) Escreva em uma linguagem do tipo *assembly*, para a máquina objeto definida, rotinas para tratamento de estrutura de pilha (push/pop/reset).
- 7) Complete a definição da estrutura de chamada de rotinas em *assembly*, usando a estrutura de pilha definida (estabeleça o que deve ser empilhado/desempilhado).
- 8) Escreva em uma linguagem do tipo *assembly*, para a máquina objeto definida, rotinas para tratamento de entrada e saída (para/de uma região de memória).
- 9) Complete a definição do acesso ao SO.
- 10) Defina a estrutura dos programas montador e carregador. Como serão executados os seus programas na máquina virtual?

PCS2056 – Linguagens e Compiladores

Assunto: Tradução das declarações

Objetivo: Com base na especificação do ambiente de execução, desenvolvido anteriormente, projetar o mecanismo de tradução para as declarações de uma linguagem imperativa.

Palavras-chave:

tabelas de símbolos e de atributos

Atividades de Projeto:

- 1) Projetar uma rotina responsável pela montagem e pela busca de informação em uma **tabela de símbolos (= tabela de nomes)**, a ser usada para guardar os nomes das variáveis, e de uma **tabela de atributos**, na qual são informados o seu tipo, uma indicação sobre o fato de já ter sido declarada ou não, referenciada ou não, e um indicador da posição de memória onde as mesmas foram alocadas.
- 2) Projetar um algoritmo para a **tradução das declarações de variáveis simples: inteira, booleana, real**. Este algoritmo deve promover a alocação de espaço para as variáveis, e estabelecer a correspondência entre as variáveis indicadas na declaração e sua respectiva área alocada na memória, contabilizando o uso da memória e registrando as variáveis e seus atributos nas tabelas de símbolos e de atributos.
- 3) Estender o algoritmo anterior de forma que traduza também **declarações de agregados** homogêneos (vetores e matrizes). Esse procedimento deverá alocar espaço para conter na memória os elementos do agregado, e também construir e depositar na memória, para cada matriz ou vetor declarado, o correspondente descritor, registrando tudo isso nas tabelas de símbolos e de atributos.
- 4) Projetar um algoritmo que, dados os índices e o descritor de um agregado homogêneo, **localize** na memória o elemento referenciado, **extraindo** daí o valor corrente ou então nele **depositando** um novo valor fornecido.
- 5) Repetir as duas últimas atividades para **agregados heterogêneos**.