

Enunciado Laboratório 04 de Compiladores.
Entrega 14/09/2024

Você deve:

- **Fazer o trabalho individualmente;**
- **Enviar um arquivo .zip contendo todos os arquivos necessários para rodar o gerador de descrição de uso de Frame por Função em cima da fatia definida da sua gramática;**
 - **Sua descrição de uso do Frame final deve ser um objeto do tipo FrameFuncao preenchido com os campos do objeto**
 - **As ocorrências de variáveis no seu programa também devem ter a adição de um objeto do tipo FrameAcesso**
- **Realizar os envios SOMENTE pelo Classroom, até 14/09/2024.**

Estenda o seu lab 3 para que ele gere uma descrição de uso do Frame por uma função. *Caso você tenha usado a gramática do Lab2 no Lab3, sua gramática do Lab3 deve ser suficiente para este lab. Caso você tenha usado uma versão restrita no Lab3, você deve estender sua gramática do lab3 para uma fatia maior da gramática da sua linguagem sorteada para adicionar a possibilidade de chamadas de funções serem parte de expressões.* Tais chamadas de funções devem poder ter como parâmetros listas de expressões. (Veja arquivos gramatica.conf do .zip, no diretorio gramatica-6, especificamente os símbolos LE e LEX).

Seu programa deve gerar um objeto FrameFuncao com os campos preenchidos

- int tamanho_frame;
 - O número total de bytes necessários ao seu frame.
- int n_param_entrada;
 - número de parâmetros de entrada para o seu frame
- int n_maximo_param_saida;
 - número máximo de parâmetros de saída do seu frame
- int n_pseudo_registradores;
 - número total de pseudo-registradores da sua função
- int n_variaveis_no_frame;
 - número total de variáveis guardadas no seu frame.

Seu programa também deve preencher para cada uso de Variável o modo objeto do tipo de modo de acesso à variável (no frame ou em pseudo-registrador), com os campos correspondentes preenchidos.

Seu uso do Frame deve seguir as seguintes regras para termos uma padronização.

- Todas as variáveis ocuparão 8 bytes seja em memória ou registrador;
- O Frame utilizará os 40 primeiros bytes (5 palavras) para sua organização interna (endereço de retorno, frame pointer anterior, etc)
- Todos os parâmetros devem ficar no frame anterior (o que chamou o frame atual). Suas posições devem ser aquelas logo acima do FramePointer. O primeira parâmetro deve ficar na posição FramePointer+8, o segundo parâmetro na +16, etc. Os parâmetros não ocupam espaço do frame atual;

- As variáveis que ficarem no frame utilizarão posições após os 40 primeiros bytes. Portanto a primeira variável a utilizar o frame utilizará a posição FP-40, a segunda FP-48, etc. Mantém-se assim o padrão de se utilizar posições decrescentes ao se utilizar o espaço do frame.
- As variáveis internas da função podem ficar no Frame ou em Pseudo-Registradores. Artificialmente, definimos aqui que aquelas variáveis que aparecerem dentro de alguma expressão que é parâmetro de alguma função chamada deve ficar no Frame. Já aquelas variáveis que não aparecem em tal situação devem ficar em Pseudo-Registradores.
- Os demais tipos de variáveis internas devem ficar em pseudo-registradores. Esta regra é artificial e desnecessária, tendo sido criada para exercitar os dois tipos de alocações. As variáveis internas alocadas no frame ocupam posições abaixo do FramePointer, decrescentemente a partir da posição -40;
- Para especificar os 2 diferentes tipos de acesso, você deve preencher os campos FrameAcesso da classe Variavel com um dos dois possíveis tipos de FrameAcesso. No caso de FrameAcessoNoFrame, o campo posicao_no_frame indica o delta em bytes a partir do frame pointer. (Note que duas ocorrências da mesma variável devem ter campos acesso_frame equivalentes ou iguais (mesmo apontador). No caso de FrameAcessoTemp, o campo id identifica unicamente o Pseudo-Registrador e os números de id devem ser definidos sequencialmente a partir de 1, na ordem de aparecimento das variáveis que serão implementadas como pseudo-registradores.