

Compiladores II

Fabio Mascarenhas – 2018.1

<http://www.dcc.ufrj.br/~fabiom/comp2>

Ambientes

- Se uma função acessa variáveis locais de fora dela precisamos levar isso em conta
- Em uma linguagem com funções de primeira classe uma função pode acessar uma variável mesmo depois do escopo onde ela está não estar mais na pilha
- Uma maneira comum de resolver isso é transformar a pilha em uma *árvore*, onde cada nó da árvore seria o pedaço de pilha correspondente a uma chamada de função
- Quando criamos um valor função, esse valor carrega um ponteiro para o nó da chamada corrente
- Um acesso a variável tem a *profundidade* dela – quantos links seguir na árvore

Índices de Bruijn

Exercício

- Modifique a máquina virtual para implementar ambientes, dividindo os 8 bits do índice da variável nas instruções LOAD e STORE em 3 bits de profundidade e 5 de índice

Displays

- Uma ambiente em árvore funciona bem, mas desperdiça muita memória, já que mesmo variáveis que não são acessadas são mantidas acessíveis
- Podemos resolver isso criando um *display* para cada valor de função criado; esse display possui ponteiros para os valores das variáveis externas que a função efetivamente acessa
- Os displays de duas funções que acessam a mesma variável compartilham o mesmo valor
- Podemos manter os valores na pilha enquanto eles estão em escopo ao custo de uma indireção a mais

Exercício

- Modifique a máquina virtual de ambientes para displays; o offset da variável agora fica codificado nos 7 bits menos significativos do argumento de LOAD/STORE, e o bit mais significativo é 0 para uma variável local à função e 1 para uma variável do display

