

Compiladores II

Fabio Mascarenhas – 2018.1

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

Funções Simples

- Funções que acessam apenas suas próprias variáveis locais são simples
- Cada função tem seu próprio código, e sua própria tabela de constantes
- Na chamada de uma função também precisamos guardar a posição atual do program counter
- Podemos reunir essas três coisas em um *frame* de chamada de função, e prealocar uma pilha deles
- Nossa pilha de operandos agora pode crescer toda vez que chamamos uma função

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