Geração de Código - Controle

- Expressões relacionais e condicionais normalmente são usadas para *controlar* a execução, através de saltos condicionais

- As arquiteturas quase sempre associam operações relacionais a saltos, e podemos aproveitar essa característica fazendo expressões condicionais (relacionais e lógicas) gerarem código que tem efeito de saltar para determinado label (com efeito final na pilha neutro)

- Geralmente temos código mais compacto se uma expressão condicional gera código que salte para determinado label se ela for *falsa* ao invés de verdadeira

Geração de Código - Controle

- Saltar para o bloco else, saltar para a saída do laço while, saltar para o início do corpo no laço repeat... Todos esses saltos acontecem se a condição correspondente for *falsa*

```
if x < 5 then
  write 0
else
  write 1
end
```



```
  getglobal x
  icload 5
  if_icmpge $else
  icload 0
  write
  jmp $fim
$else:
  icload 1
  write
$fim:
```

Geração de Código - Procedimentos

- O valor de retorno de um procedimento é a primeira variável local (após os parâmetros)
- Talvez precisemos ajustar a pilha antes de começar a empilhar os parâmetros, por questões de alinhamento – preinvoke
- Os argumentos são empilhados do último para o primeiro
- Uma chamada de procedimento pode ser tanto um comando quanto uma expressão – se for um comando precisamos dar um “pop” após a chamada
- Se ajustamos a pilha antes de empilhar os parâmetros, precisamos fazer isso novamente após a chamada – postinvoke