

COMPILADORES

Prof. Marcus Ramos — P2 — 08/07/2026

1. (1,0 ponto) Quantos bits (no mínimo) são necessários para representar um tipo de dados com n valores? Quantos valores é possível representar com m bits?
Resposta: $\log_2(n)$ e 2^m , respectivamente.
2. (1,0 ponto) Suponha a seguinte declaração em Pascal: `var m:array[2..5, 3..7] of integer;` Se a base é 1.000 e $\text{size}(\text{integer}) = 2$, qual é a origem virtual de m ?
Resposta: $\text{address}(m[i][j]) = 1.000 + (i - 2) * 10 + (j - 3) * 2 = 1.000 + 10 * i - 20 + 2 * j - 6 = 1.000 - 20 - 6 + 10 * i + 2 * j = 974 + 10 * i + 2 * j$. A origem virtual é 974.
3. (1,0 ponto) Por que a avaliação de expressões em máquina de registradores costuma ser mais eficiente do que a geração de código para máquina com pilha?
Resposta: Porque os dados são operados dentro da máquina e não fora dela. Além disso, o conteúdo dos registradores pode ser reaproveitado se o compilador mantiver controle sobre o conteúdo de cada um.
4. (1,0 ponto) Por que os parâmetros são acessados com deslocamento negativo em relação ao LB durante a execução de um subprograma parametrizado?
Resposta: Porque eles são avaliados e deixados na pilha antes da criação do frame do subprograma em questão.
5. (1,0 ponto) Qual a função do LE (link estático) e do LD (link dinâmico)?
O LE é o LB (local base) da instância mais recente do frame correspondente ao bloco que envolve estaticamente o bloco corrente. O LD é o LB do subprograma executado imediatamente antes da execução do subprograma corrente.
O LE é usado para acessar variáveis não locais ao bloco em execução. O LD preserva o LB do frame do bloco que chamou o bloco em execução.
6. (1,0 ponto) Como deve ser endereçada uma variável não-local declarada no bloco que envolve o bloco que envolve o bloco corrente?
Resposta: Percorrendo-se o LE duas vezes ou então usando o L2 do display.
7. (1,0 ponto) Em que consiste a “coalescência” no gerenciamento do heap?
Resposta: Trata-se de agrupar blocos livres consecutivos visando reduzir a fragmentação do heap.
8. (1,0 ponto) De que maneira os templates de código capturam a semântica da linguagem fonte?
Resposta: Associando cada construção da linguagem fonte a uma série de instruções da linguagem objeto (ou outros templates de código) de tal forma que a semântica seja preservada.
9. (1,0 ponto) Por que a execução por um interpretador é normalmente menos eficiente do que a execução de um código compilado?
Resposta: Porque toda vez que código interpretado é executado, é necessário fazer análise léxica, sintática e de contexto antes da execução do código. No caso do código compilado, estas etapas são executadas durante a compilação, não onerando a execução do código compilado. Além disso, o interpretador deve ser carregado na memória ao passo que o código compilado dispensa a presença do compilador durante a sua execução.
10. (1,0 ponto) Por que se diz que um compilador é um programa que “efetua uma transformação sintática preservando o conteúdo semântico”?
Resposta: Porque o objetivo do compilador é transformar o código escrito na linguagem fonte para um outro código (com uma outra sintaxe) na linguagem objeto, mas que preserve o significado (semântica) da linguagem fonte.