

26. [Sebesta, 2000] Porque os ponteiros da maioria das linguagens restringem-se a apontar para um objeto de tipo único?

Um tipo ponteiro é aquele em que as variáveis têm uma faixa de valores que consiste em endereços de memória e um valor especial - `nil` ou `null` - para indicar que um ponteiro não está atualmente referenciando qualquer célula de memória.

Os ponteiros diferem das variáveis escalares porque são mais frequentemente usados para referenciar alguma outra variável, em vez de armazenarem dados de alguma espécie. Assim, os ponteiros necessitam conhecer as características da variável a qual estão referenciando.

Por exemplo, caso um ponteiro `ptr` aponte para uma célula de memória de um tipo de dados que ocupe quatro unidades de memória de tamanho, a instrução `ptr++` deverá fazer com que o valor de `ptr` seja incrementado em quatro unidades, se posicionando no próximo valor válido, ao invés de incrementar seu valor em apenas uma unidade, se posicionando numa posição inválida.

Ou seja, o compilador associa cada variável a uma posição única em memória, capaz de suportar as informações do tipo dessa variável. Sempre que um programa faz referência a uma variável estamos, na realidade, nos referindo ao endereço ou conjunto de endereços que essa variável ocupa. Assim, as linguagens de programação restringem os ponteiros a apontarem para um tipo único, de forma que o compilador saiba a representação da informação apontada pelo ponteiro.