

# TABELAS DE SÍMBOLOS

As tabelas de símbolos são estruturas de dados usadas pelos compiladores para conter informações sobre as construções do programa fonte. As informações são coletadas de modo incremental pelas fases de análise e usadas pelas fases de síntese para gerar o código objeto.

As entradas na tabela de símbolos são criadas e usadas durante a fase de análise pelo analisador léxico, analisador sintático e pelo analisador semântico.

As entradas na tabela de símbolos contêm informações sobre um identificador, como por exemplo: nome, tipo, forma, tamanho, endereço de memória, etc.

As tabelas de símbolos normalmente precisam dar suporte a múltiplas declarações do mesmo identificador dentro de um programa.

As tabelas de símbolos permitem capturar a sensibilidade ao contexto e as ações executadas no decorrer do programa.

O escopo de uma declaração é a parte de um programa à qual a declaração se aplica. As tabelas de símbolos podem ser separadas para cada escopo.

O papel de uma tabela de símbolos é passar informações de declarações para diversos usos.

Uma ação semântica entra com informações sobre o identificador  $x$  na tabela de símbolos, quando a declaração de  $x$  é analisada.

Uma ação semântica associada a uma produção como  $\text{factor} \rightarrow \text{id}$  recupera a informação sobre o identificador da tabela de símbolos.

As tabelas de símbolos permitem saber durante a compilação de um programa o tipo e o valor de seus elementos (números e identificadores), escopo destes, número e tipo dos parâmetros de um procedimento, etc.

Exemplo de uma tabela de símbolos:

Cadeia	Token	Categoria	Tipo	Valor	Escopo	Utilizada
i	id	var	integer	1		S
fat	id	proc	-	-		...
2	num	-	integer	2		N
...						

A tabela de símbolos é acessada pelo compilador sempre que um elemento é mencionado no programa, para:

- Verificar ou incluir sua declaração.

- Verificar seu tipo, escopo ou alguma outra informação.
- Atualizar alguma informação associada ao identificador (por exemplo, valor e escopo).
- Remover um elemento quando este não se faz mais necessário ao programa. Por exemplo: variáveis locais a uma função.

Algumas questões sobre a implementação da tabela de símbolos:

- A estrutura da tabela de símbolos determinada pela eficiência das operações de inserir, verificar e remover.
- Implementação  
Estática.  
Dinâmica: melhor opção.
- Estrutura  
Listas, matrizes.  
Árvores de busca (por exemplo, B e AVL).
- Acesso  
Sequencial, busca binária, etc.  
Hashing: opção mais eficiente.

Com relação a implementação da tabela de símbolos:

- Pode ser implementada uma única tabela para todas as declarações ou várias tabelas, sendo uma para cada tipo de declaração (constantes, variáveis, tipos, procedimentos e funções).
- Podem ser implementadas várias tabelas: diferentes declarações têm diferentes informações/atributos.