

Procesadores de lenguaje

→ Tema 6 – La tabla de símbolos



Salvador Sánchez, Daniel Rodríguez
Departamento de Ciencias de la Computación
Universidad de Alcalá

→ Resumen

- La tabla de símbolos.
- Requerimientos de información.
- Diseño de la tabla de símbolos.
- Gestión en un lenguaje con estructura de bloques.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Introducción

- La tabla de símbolos es una estructura global utilizada por distintos módulos del compilador
 - Es el principal atributo heredado.
- Contiene una entrada para cada uno de los símbolos definidos en el programa fuente.
 - Sobre los identificadores, y opcionalmente sobre las palabras reservadas y las constantes.
 - Información sobre el lexema, tipo de datos, ámbito y dirección en memoria.
- Operaciones principales:
 - Insertar: introduce un símbolo tras una declaración.
 - Buscar: recupera información asociada a un símbolo.
 - Eliminar: borra la información de un símbolo cuando ya no se utiliza.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Introducción

- Datos que se almacenan:

- Para un array:

- Tipo de los elementos.
 - Número de elementos.
 - Límites inferior y superior.

- Para una función:

- Número de parámetros.
 - Tipo de los parámetros.
 - Forma de paso de parámetros.
 - Tipo de retorno.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Estructura de la tabla de símbolos

- Estructura de datos global de tipo diccionario.

- Principalmente hay tres estructuras de implementación:

- **Listas lineales** simple o doblemente enlazadas, normalmente de tamaño variable. Es un sistema sencillo, pero lento cuando hay muchas entradas.
 - **Tablas de dispersión** (*hash*), su eficacia depende de la tabla de dispersión elegida. Es la más utilizada.
 - **Árboles** de búsqueda binarios, AVL y árboles B. No demasiado útiles por la complejidad de ciertas operaciones como la eliminación.

- Las entradas de la tabla de símbolos deben ser suficientemente flexibles como para almacenar información heterogénea.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ TDS como tabla de dispersión

- Una tabla de dispersión es un array con entradas indexadas:

- Una función de dispersión convierte el nombre del identificador en un valor entero que corresponde con un índice.
 - La resolución de colisiones se lleva a cabo mediante **encadenamiento por separado** (una lista enlazada en cada índice).
 - La función de dispersión debe producir pocas colisiones.



Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Declaraciones

- Dos tipos de declaraciones: explícitas e implícitas.
- Hay cuatro clases de declaraciones **explícitas**:
 - De constante: `const double PI = 3.1415;`
 - De tipo: `struct Complejo{ float re, im; }`
 - De variable: `bool encontrado;`
 - De función: `void f(int,float);`
- Declaraciones **implícitas**: en Fortran un identificador no declarado que comience por una letra entre la *i* y la *n* es un entero.
- Los atributos asociados a un nombre en la tabla de símbolos dependen del tipo de declaración.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Reglas de ámbito

- Aunque varían mucho entre los diferentes lenguajes de programación, es posible identificar reglas comunes.
- **Declaración antes del uso**: regla que obliga a declarar un identificador antes de hacer referencia a él en el programa fuente.
 - Facilita la compilación, pues permite construir la TDS a medida que se analiza sintácticamente el código fuente.
 - Si al hacer una búsqueda de un símbolo, ésta falla, se detecta una violación de la regla.
 - Fomenta la compilación en un sólo paso.
- **Estructura de bloques**: cada bloque puede contener declaraciones, algunas de las cuales pueden incluir nombres ya utilizados.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Las estructura de bloques

- Un lenguaje está estructurado en bloques:
 - Si permite la anidación de unos bloques dentro de otros, y
 - Si el ámbito de una declaración se limita al bloque y sus bloques anidados.

- Ejemplo:

```
float g(int a, floatb){  
    int x, y;  
    for (x=0;x<10;x++){  
        bool encontrado;  
        //...  
        {  
            char* x; //...  
        }  
    }  
}
```

regla de anidación
más próxima

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Las estructura de bloques

- Regla de **anidación más próxima**: las referencias a un identificador se refieren siempre a la declaración realizada en el bloque más próximo.
- Implementación de esta regla en la tabla de símbolos:
 - La operación *Insertar* no debe sobrescribir declaraciones anteriores.
 - La operación *Buscar* debe encontrar siempre la declaración más reciente.
 - La operación *Eliminar* sólo elimina la declaración más reciente de un nombre.
- Al procesar ámbitos anidados, la tabla de símbolos se comporta como una pila.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Las estructura de bloques

- Ejemplo basado en una tabla hash.

- Código a analizar:

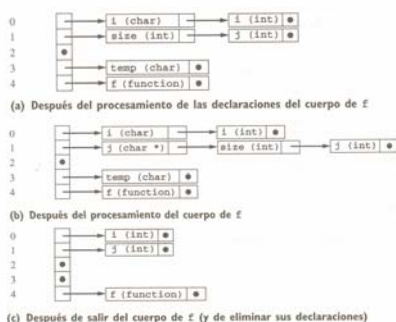
```
int i,j;
int f(int size){
    char i,temp;
    //..
    {
        char* j;
        //..
    }
}
```

- Se supone que la aplicación de la función de dispersión sobre "j" y "size" retorna el mismo valor: 1.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Las estructura de bloques



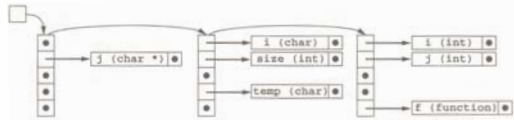
Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Las estructura de bloques

- Con varias tablas anidadas:

```
int i,j;  
int f(int size){  
    char i,temp;  
    //...  
    {  
        char* j;  
        //...  
    }  
}
```



Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Alternativas de implementación

- Construir una nueva tabla de símbolos para cada ámbito, vinculando las tablas desde los ámbitos internos a los ámbitos externos.
 - Una operación de búsqueda continuaría en la tabla "padre" si no obtiene resultados en la tabla actual.
 - Al abandonar un ámbito, se elimina la tabla de símbolos completa.
- En lenguajes en los que los ámbitos están contenidos unos en otros (Pascal) puede bastar con numerar los ámbitos por niveles
 - La inserción de un elemento se realiza en el nivel actual
 - Al cambiar de ámbito, se incrementa o decrementa el nivel actual (al llamar a una función se incrementa y al retornar se decrementa)
 - Es un método mas sencillo pero menos eficiente.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez



→ Bibliografía

- **Básica:**
 - *Construcción de compiladores. Principios y práctica.* Kenneth C. Louden. Thomson-Paraninfo. 2004.
 - *Compiladores: principios, técnicas y herramientas.* A.V. Aho, R. Sethi, J.D. Ullman. Addison-Wesley Iberoamerica. 1990.

Procesadores de lenguaje - Tema 6: La tabla de símbolos
Salvador Sánchez, Daniel Rodríguez