Symbol Table Management

Presentation by,
Rohin Kotagiri (22911A05N4)

What is Symbol Table?

A symbol table is a data structure that stores information about program identifiers - basically everything the compiler needs to remember about your code.

Name	Type	Size	Scope	Memory Location
count	int	4	-	-
x	str	10	-	-
y	float	4	-	-

Structure of Symbol Table Entries

Typical fields in a symbol table entry:

- Name (identifier string)
- Type (int, float, user-defined, etc.)
- Scope information
- Memory allocation details
- Additional attributes (const, static, etc.)

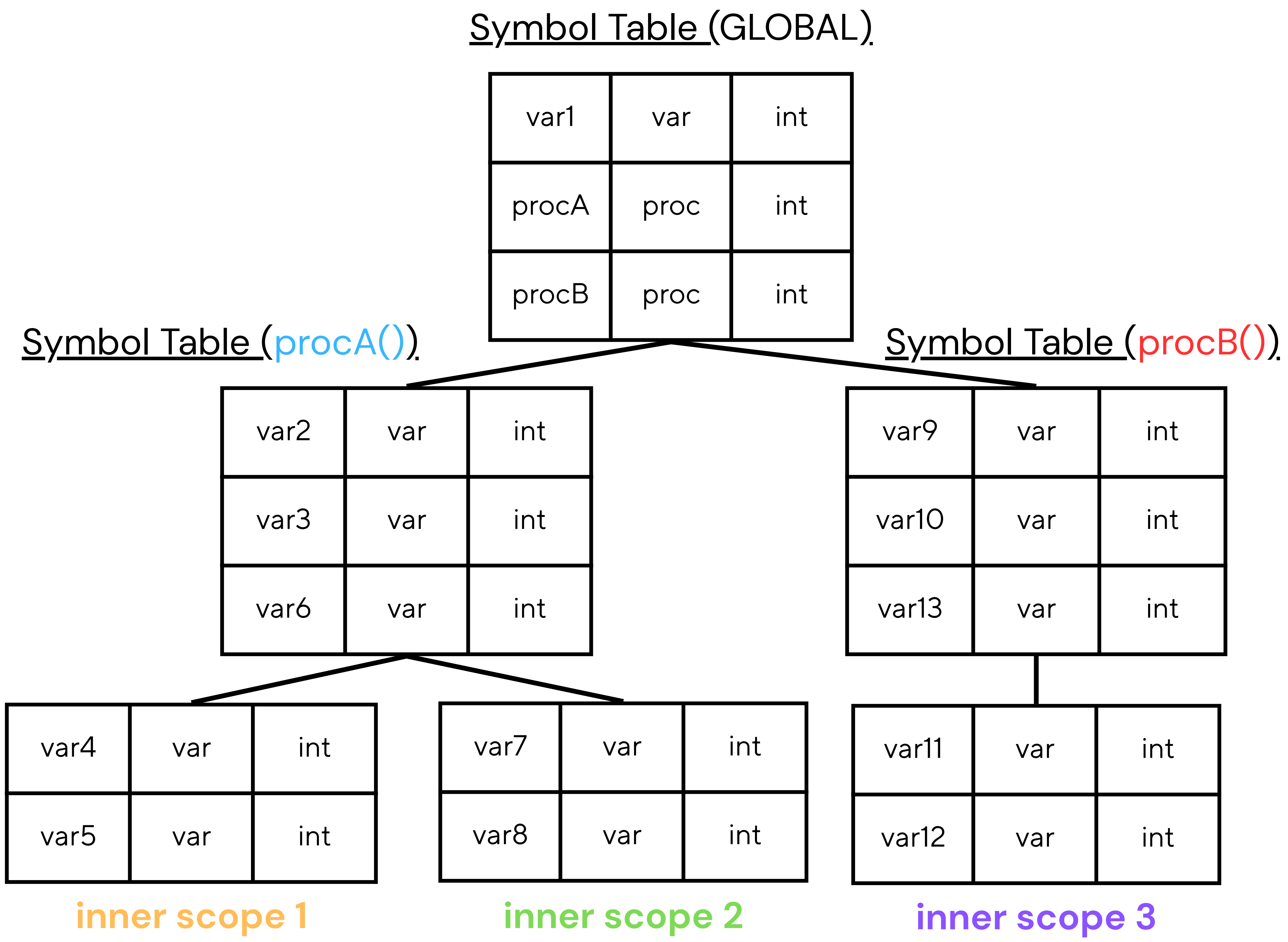
Name	**Type**	**Size**	**Scope**	**Memory Location**
count	int	4	-	-
x	str	10	-	-
y	float	4	-	-

Core Symbol Table Functions

- Insert: Add new identifiers during declarations
- Lookup: Check existence and retrieve attributes
- Update: Modify entries as compilation progresses
- Delete: Remove entries when no longer needed
- Scope management: Handle nested/block scopes

Example

```
int var1;
int procA() {
    int var2, var3;
    {
        int var4, var5;
    }
    int var6;
    {
        int var7, var8;
    }
}
int procB() {
    int var9, var10;
    {
        int var11, var12;
    }
    int var13;
}
```



Implementation Strategies

Chained Hash Tables

Open Addressing Techniques

Dynamic Resizing and Rehashing

Memory Pool Allocation

Thank you

Presentation by,
Rohin Kotagiri (22911A05N4)