Jiangxi University of Science and Technology

Chapter 12 Structures

Lecture1203 Union

12.5 Unions

- A union is a data type that reserves the same area in memory for two or more variables
  - **union**{
  - **char** key; **int** num;
  - **double** price;
  - } **val**;
  - Each of these types, but **only one at a time, can actually be assigned to the union variable**
  - A union reserves sufficient memory locations to accommodate its **largest member's data type**

12.5 Unions

- Individual union members are accessed using the same notation as structure members
- Typically, **a second variable keeps track of the current data type stored in the union**
  - **switch** (**uType**)
  - {
  - **case** ‘c’: printf("%c", val.key); **break**;
  - **case** ‘i’: printf("%d", val.num); **break**;
  - **case** ‘d’: printf("%f", val.price);**break**;
  - **default** : printf("Invalid type : %c", uType);
  - }

12.5 Unions

- **A type name** can be associated with a union to create templates
  - **union DateTime**
  - { **long** days; **double** time;};
  - **union DateTime** first, second, *pt;
- **Pointers** to unions use the same notation as pointers to structures

12.5 Unions

- Unions may be **members** of structures and arrays; structures, arrays, and pointers may be members of unions
  - struct{
  - char uType;
  - **union { char *text; double rate;}** uTax;
  - } flag;
  - rate is referenced as **flag.uTax.rate**
  - ***flag.uTax.text = ?**

Common Programming Errors

- Attempting to use structures and unions, as complete entities, in relational expressions
- Assigning an incorrect address to a pointer that is a member of a structure or union
- Storing one data type in a union and accessing it by the wrong variable name can result in an error that is particularly troublesome to locate

Common Compiler Errors

Error	Typical Unix-based Compiler Error Message	Typical Windows-based Compiler Error Message

Using the wrong type of braces when declaring a structure.
For example:
`struct`
`[`
`  int month;`
`  int day;`
`  int year;`
`] birth;`
	The following error will be reported on each line containing a brace:
	`(S) Syntax error.`
		`:error: syntax error : missing ';' before '[' : error: syntax error : missing ']' before ';'`

Attempting to initialize the elements of a structure inside the declaration.
For example:
`struct`
`{`
`  int month = 6;`
`  int day;`
`  int year;`
`} birth;`
	`S) Syntax error: possible missing ';' or ','?`	`:error: 'month' : only const static integral data members can be initialized inside a class or struct`

Assigning a pointer to a structure rather than the address of the structure.
For example:
`int main()`
`{`
`  struct Date *ptr;`
`  struct Date birth;`
`  ptr = birth;`
`}`
	`(S) Operation between types "struct Date*" and "struct Date" is not allowed.`	`:error: '=' : cannot convert from 'Date' to 'Date *'`

Summary

- A structure allows individual variables to be grouped under a common variable name
- A structure type name can be used to create a generalized structure type describing the form and arrangement of elements in a structure
- Structures are particularly useful as elements of arrays
- Individual members of a structure are passed to a function in the manner appropriate to the data type of the member being passed
- Structure members can be any valid C data type, including structures, unions, arrays, and pointers
- Unions are declared in the same manner as structures

Reference

- https://www.codesdope.com/blog/article/int-main-vs-void-main-vs-int-mainvoid-in-c-c/