

Go4Expert > Articles > Programming > C > Unions in C : A less used but a powerful feature of C

Unions in C : A less used but a powerful feature of C



Contributed by [Amit Ray](#) On 20th July, 2004

This is an article on *Unions in C : A less used but a powerful feature of C* in C.

Rated 5.00 By 3 users

WHAT IS A UNION

A union, is a collection of variables of different types, just like a structure. However, with unions, you can only store information in one field at any one time.

You can picture a union as like a chunk of memory that is used to store variables of different types. Once a new value is assigned to a field, the existing data is wiped over with the new data.

A union can also be viewed as a variable type that can contain many different variables (like a structure), but only actually holds one of them at a time (not like a structure). This can save memory if you have a group of data where only one of the types is used at a time. The size of a union is equal to the size of it's largest data member. In other words, the C compiler allocates just enough space for the largest member. This is because only one member can be used at a time, so the size of the largest, is the most you will need. Here is an example:

```

...
union time
{
    long simpleDate;
    double perciseDate;
}mytime;
...
  
```

The union above could be used to either store the current time (in seconds) to hold time accurate to a second. Or it could be used to hold time accurate to a millisecond. Presumably there are times when you would want one or the other, but not both. This declaration should look familiar. It is the same as a struct definition, but with the keyword union instead of struct.

ACCESSING UNION FIELDS

To access the fields of a union, use the dot operator(.) just as you would for a structure. When a value is assigned to one member, the other member(s) get whipped out since they share the same memory. Using the example above, the precise time can be accessed like this:

```

...
printTime( mytime.perciseDate );
...
  
```

In larger programs it may be difficult to keep track of which field is the currently used field. This is usually handled by using another variable to keep track of that. For example, you might use an integer called mode. When mode equals one, the regular date (simpleDate) is used. If mode is two, then perciseDate is used. This mode variable needs to be set every time a different member in the union is used. Here is a sample program to illustrate the use of unions.

```
#include <stdio.h>
```

```

int main()
{
    union data
    {
        char a;
        int x;
        float f;
    } myData;
  
```

```

int mode = 1;

myData.a = 'A';
printf("Here is the Data:\n%c\n%i\n%.3f\n", myData.a, myData.x, myData.f );

myData.x = 42;
mode = 2;
printf("Here is the Data:\n%c\n%i\n%.3f\n", myData.a, myData.x, myData.f );

myData.f = 101.357;
mode = 3;
printf("Here is the Data:\n%c\n%i\n%.3f\n", myData.a, myData.x, myData.f );

if( mode == 1 )
    printf("The char is being used\n");

else if( mode == 2 )
    printf("The int is being used\n");

else if( mode == 3 )
    printf("The float is being used\n");

return 0;

}

```

This little program declares a union with an int, float, and char. It uses each field, and after each use prints out all the fields (with one ugly printf statement). Here is some sample output:

Here is the Data:

A

577

0.000

Here is the Data:

*

42

0.000

Here is the Data:

1120581321

101.357

The float is being used

Your output might be different. Clearly the data in the unused fields is just garbage. This happens because different types are treated differently by the computer. So if one type is set, the memory is not going to be in the format of the other types. Also mode is used to keep track of the type. In this program it is pretty useless, but in larger programs it would be a great help in keeping track of the unions use.

Summary

Union allows same storage to be referenced in different ways

Only one way is valid at any given time

Usage

- access individual bytes of larger type
- variable format input records (coded records)
- sharing an area to save storage usage
- unions not used nearly as much as structures

Example

```

union HAH
{
    int igor;

```

```
char chuck;
float felix;

};
...
union HAH Jody;
...
```

```
Jody.igor = 2175; /* felix and chuck not valid */
Jody.chuck = 'X'; /* igor and felix not valid */
Jody.felix = 2.5; /* igor and chuck not valid */
```

```
t = Jody.igor;
```

- sizeof union is size of its biggest member
- Unions most often contain different types of structures
- Can only initialize first member of union
- Can assign (copy) one union variable to another
- Can pass union or pointer to union as function arg
- Function can return union type
- Can define pointers to union type object
- Members accessed as unionvar.member or unionptr->member
- Syntax, format and use of tags and declarators like struct, but members overlay each other, rather than following each other in memory

```
union HAH
{
    short svar;
    long lvar;
    char cvar;
    long double dvar;
};
```

□

Thus we see Unions are declared in the same fashion as structs, but have a fundamental difference. Only one item within the union can be used at any time, because the memory allocated for each item inside the union is in a shared memory location.

```
struct conditions
{
    float temp;
    union feels_like
    {
        float wind_chill;
        float heat_index;
    }
}
```

```
} today;
```

As you know, wind_chill is only calculated when it is "cold" and heat_index when it is "hot". There is no need for both. So when you specify the temp in today, feels_like only has one value, either a float for wind_chill or a float for heat_index.

Types inside of unions are unrestricted, you can even use structs within unions.

Other Resources :

http://www.iota-six.co.uk/c/h5_unions.asp
<http://www.sysprog.net/cunions.html>
<http://publications.gbdirect.co.uk/c...r6/unions.html>

Amit Ray. 😊

The trouble with programmers is that you can never tell what a programmer is doing until it's too late. (Seymour Cray) 😊

ankit.gotawala, shabbir, theMachineUser likes this

Related Articles

- [100 Multiple choice questions in C](#), by **coderzone** in C
- [Overview of C#](#), by **Sanskriti** in C#
- [40 SEO Interview Questions](#), by **coderzone** in Internet Marketing

[Learn Xcode and iOS](#)
AboutObjects.com



[asadullah.ansari](#)
TechCake

24Jan2008, 17:49



#2

Who is saying..It's uses is less. In all Embedded Products and telecom(semicon) union is used broadly!!!



[debleena_doll2002](#)
Ambitious contributor

8Feb2008, 09:38



#3

Except it (Telecom, semicon etc) , other software for finance, statistical software also should use this pretty nice feature.



[nileshkumar_1982](#)
Go4Expert Member

11Feb2008, 12:52



#4

By using union , we can make optimized code. when we are going to code then we should remember in mind " Union also a feature" . According to that we should write code.



[crazytolearn57](#)
Go4Expert Member

26Feb2008, 18:40



#5

Quote:

Originally Posted by **nileshkumar_1982**

By using union , we can make optimized code. when we are going to code then we should remember in mind " Union also a feature" . According to that we should write code.

i agree



[bashamsc](#)
Go4Expert Member

4 GE AUTHOR

27Feb2008, 19:40



#6

Good information



[aisha.ansari84](#)
Contributor

5Mar2008, 18:15



#7

unions are powerful, but one should know the proper place to use it



[iamanexpert](#)
Newbie Member

11Jun2010, 19:27



#8

i learned a lot from this article



Manojbijnori
Go4Expert Member

12Jun2010, 11:22



This is really good but which one is better union or structure



Tobiasgar
Go4Expert Member

1Sep2011, 15:02



The select queries that you combine in a union query must have the same number of output fields, in the same order, and with the same or compatible data types. When you run a union query, data from each set of corresponding fields is combined into one output field, so that the query output has the same number of fields as each of the select statements. 😊

Page 1 of 2 1 2 >

Article Tools



[Rate this Article](#)

[Printable Version](#)

[Subscribe](#)



Already a member?

User Name: ☐ Save

Password

[Forgot Username/Password?](#)

Not a member yet?

[Create Your Free Account Now.](#)

Search Go4Expert

[Advance Search](#)

Latest Forum Topics

[Need help in Parsing of file in C/C++ !!](#)

Under C

[Need urgent help with C VIM program](#)

Under C

[please help me with printf\(\)](#)

Under C

[draw multiple lines and move then on screen](#)

[Under C](#)

[Problem on Linked list](#)

[Under C](#)

Find Us On Facebook

Latest Articles

[ATOI and ITOA Custom Implementations](#)

[Under C](#)

[Error Handling in C](#)

[Under C](#)

[Understanding float datatype in C](#)

[Under C](#)

[Complete Guide to Understanding static in C](#)

[Under C](#)

[Builtin GCC Functions - __builtin_clz\(\);
__builtin_ctz\(\); __builtin_popcount\(\);](#)

[Under C](#)
