

Re: confused about extern use

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-02/msg03244.html

- *From:* "Nick Keighley" <nick_keighley_nospam@xxxxxxxxxxxx>
 - *Date:* 21 Feb 2007 01:46:48 -0800
-

On 21 Feb, 08:36, "Lalatendu Das" <lalat...@xxxxxxxx> wrote:

On Feb 21, 12:22 pm, Ian Collins <ian-n...@xxxxxxxx> wrote:>

Lalatendu Das wrote:

Here in the above example I am confused about, what extra the coder going to achieve by declaring it as extern in the header file a.h.

it would have helped if you'd left the example in...

So it can be used in another compilation unit.

Actually what u mean by another compilation unit.

abbreviations like "u" don't add to the clarity.

"another compilation" unit is basically another C file (actually it's the C file and its associated includes). A complete program is composed on one or more compilation units.

If my assumption is not wrong do u mean If I will try to compile another C-program no need to include this header file or what ?

you'll need the header file to included in each C file that uses the shared variable.

Re: confused about extern use

I don't think so it will work in that case. And in any case if I have to include this header file to define one variable of structure type "abc" then why to declare it extern there.

Actually I might have some wrong notion, please explain through example if u want to?

thanks for ur replay.

each external object may have many declaration but only one definition. Essentially you **declare** the type wherever it is needed (often in an H file) but **define** the storage used in only one place (usually a C file).

a.h

```
/* declare struct abc type */
```

```
struct abc {  
  unsigned long a;  
  unsigned long b;  
};
```

```
/* declare abc_var to be type struct abc */
```

```
extern struct abc abc_var;
```

file.c

```
/* pull in declarations */
```

```
#include <a.h>
```

```
/* define abc_var in one place */
```

```
struct abc abc_var;
```

```
void f()
```

```
{  
  /* use abc_var */  
  abc_var.a = 1;  
}
```

```
/* another compilation unit */
```

```
file2.c
```

```
/* pull in declarations */
```

```
#include <a.h>
```

```
/* DON'T re-define abc_var */
```

Re: confused about extern use

Re: confused about extern use

```
void g()
{
/* use abc_var again */
abc_var.a = 0;
}
```

the program consists of two compilation units that both access abc_var. How they are joined together ("linked") is implementation dependent (see your compiler documentation).

—
Nick Keighley