

Re: [PATCH] Documentation: Explain a second alternative for multi-line macros.

## Re: [PATCH] Documentation: Explain a second alternative for multi-line macros.

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-12/msg08169.html>

---

- *From:* Muli Ben-Yehuda <[muli@xxxxxxxxxx](mailto:muli@xxxxxxxxxx)>
  - *Date:* Sun, 31 Dec 2006 21:45:01 +0200
- 

On Sun, Dec 31, 2006 at 02:32:25PM -0500, Robert P. J. Day wrote:

Generally, inline functions are preferable to macros resembling functions.

This should be stressed, IMHO. We have too many macros which have no reason to live.

-Macros with multiple statements should be enclosed in a do - while block:  
+There are two techniques for defining macros that contain multiple  
+statements.

```
-#define macrofun(a, b, c) \  
- do { \  
+ (a) Enclose those statements in a do - while block:  
+ \  
+ #define macrofun(a, b, c) \  
+ do { \  
+ if (a == 5) \  
+ do_this(b, c); \  
+ } while (0)  
+ \  
+ (b) Use the gcc extension that a compound statement enclosed in  
+ parentheses represents an expression:  
+ \  
+ #define macrofun(a, b, c) ({ \  
if (a == 5) \  
do_this(b, c); \  
- } while (0)  
+ })
```

When giving two alternatives, the reader will thank you if you explain when each should be used. In this case, the second form should be used when the macro needs to return a value (and you can't use an inline

Re: [PATCH] Documentation: Explain a second alternative for multi-line macros.

function for whatever reason), whereas the first form should be used at all other times.

Cheers,  
Muli

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>