



PREROCESADOR



● Preprocesador

Doc: <http://gcc.gnu.org/onlinedocs/cpp>

```
#include <archivo.h>
```

Inserta `archivo.h` buscándolo en `/usr/include` y los directorios de búsqueda estándar (`/usr/local/include`) o mediante las opciones de línea de comando `-I` y `-I_`.

```
#include "archivo.h"      Busca archivo en .
```

Macros: Sustitución de texto

```
#define identificador texto_reemplazo  
#define identificador(arg1, arg2, ...) texto_reemplazo
```

No va "="

Sin espacio
antes del (

No hay
revisión de
tipos

No va "="

Ejemplos:

```
#define PI 3.141592  
#define min(X, Y) ((X) < (Y) ? (X) : (Y))  
#define vabs(X) \  
    ((X) >= 0 ? (X) : -(X))  
#define porsiempre for(;;)
```

```
#undef identificador      Para eliminar la definición de una macro
```



● Macros: Problemas

```
#define prod(X, Y) (X * Y)
a = prod(1, 2);           ⇨ a = (1 * 2);           // a=2
b = prod(1+2, 3+4);     ⇨ b = (1+2 * 3+4);       // 1+6+4=11
```

```
#define vabs(X) ((X)>=0 ? (X) : -(X))
vabs(a++)
a ⇨ 4
```



- Macros: Entrecorillado

Un texto pasado a una macro se puede tratar como si estuviese entre comillas usando la directiva "#".

```
#define ENTRECORILLAR(x) #x
```

el código `printf("%s\n", ENTRECORILLAR(1+2));`
se expande a `printf("%s\n", "1+2");`

Útil para la concatenación de cadenas automáticas para depurar macros.

```
#define ver(x, format) printf("%s:%u: %s="format, __FILE__, __LINE__, #x, x)
```

```
int una_funcion() {  
    int var=1;  
    /* aquí va código que manipula a var */  
    ver(var, "%d");  
}
```

```
FILE __    LINE __  
DATE __    TIME __  
FUNCTION __
```

imprime el nombre de la expresión y su valor así como el nombre del archivo y la línea donde se ejecuta.



- Macros: Pasting

```
#define B(x,y) xy  
#define C(x,y) x##y
```

```
B(pthreads_,mutex) → pthreads_ mutex //observar el espacio  
C(pthreads_,mutex) → pthreads_mutex
```



- Macros:

```
$ gcc -DVALOR=100 -W -Wall -o programa programa.c
```

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

int main () {
    printf("%f\n", my_function(2));
    return 0;
}
```

```
$ gcc -W -Wall raiz.c -Dmy_function=sqrt
```

```
$ gcc -W -Wall raiz.c -Dmy_function=abs
```



● Macro `assert`

- ◆ El header file `<assert.h>` define esta macro
- ◆ *Assert* = aserto (aserción, afirmación, confirmación)
- ◆ Evalúa una condición 'test' y dependiendo del resultado, puede abortar el programa. Útil para *debugging*

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

```
int main () {
    assert(1>2);
    return 0;
}
```

```
$ gcc -W -Wall assert.c -o assert
$ ./assert
```

```
assert: assert.c:5: main: Assertion `1>2' failed.
Cancelado
```

- ◆ Si se coloca `#define NDEBUG` antes de `#include <assert.h>` o se compila con `-DNDEBUG` (no depurar) todas las sentencias `assert` se ignoran
- ◆ **Ejercicio:** Crear un programa en C que implemente una macro llamada `mi_assert`, con idéntica funcionalidad de `assert` de `assert.h`. No debe emplearse `assert.h`, también debe anularse con `-DNDEBUG` desde `gcc`.



● **Macros** Compilación condicional

- ◆ Existe un conjunto de directivas que puede usarse para realizar compilaciones condicionales

#if, #ifdef, #ifndef, #else, #elif y #endif (también #define #undef)

```
#if SYSTEM == WINDOWS
    #define HEADER "win.h"
#elif SYSTEM == UNIX
    #define HEADER "unistd.h"
#else
    #define HEADER "default.h"
#endif
```

```
#ifndef UNO_SOLO
    #define UNO_SOLO
    #include <maths.h>
    #include <string.h>
    Otros includes...
#endif
Codigo modulo1.c
```

```
#ifndef UNO_SOLO
    #define UNO_SOLO
    #include <maths.h>
    #include <string.h>
    Otros includes...
#endif
Codigo modulo2.c
```

- ◆ #if permite utilizar &&, ||, !, ==, <, >, etc.

Ejemplo:

```
#if VERSION==3.0 && !defined(TEXT)
    ...
    ...
#endif
```



- **Macros vs. Funciones**

- ◆ Eficiencia

Menos sobrecarga (pasar parámetros a la pila, realizar un salto, recibir parámetros ...)

Cálculos en *compilation time* en lugar de *run time*

- ◆ Menor potencia y expresividad

- ◆ Concurrencia limitada

- ◆ Preprocesadas

- ◆ C tiene muchas macros en sus librerías ejemplo: `getchar()`

Indice de macros <http://c.conclase.net/librerias/macros.php>