

Lenguaje de Programación: C++

Directivas al preprocesador

José Luis Alonzo Velázquez

Universidad de Guanajuato

Septiembre 2010

Preprocesador

Un **preprocesador** es un programa separado que es invocado por el compilador antes de que comience la traducción real. Un preprocesador de este tipo puede eliminar los comentarios, incluir otros archivos y ejecutar sustituciones de macros.

Preprocesador

Un **preprocesador** es un programa separado que es invocado por el compilador antes de que comience la traducción real. Un preprocesador de este tipo puede eliminar los comentarios, incluir otros archivos y ejecutar sustituciones de macros.

El preprocesador de C++ es el preprocesador para el lenguaje de programación C. Es el primer programa invocado por el compilador y procesa directivas como `#include`, `#define` y `#if`. Estas directivas no son específicas de C. En realidad pueden ser usadas con cualquier tipo de archivo.

Etapas

El preprocesamiento es el primer paso en la etapa de compilación de un programa esta propiedad es única del compilador de C. El preprocesador tiene más o menos su propio lenguaje el cual puede ser una herramienta muy poderosa para el programador.

Todas las directivas de preprocesador o comandos inician con un #.

Las que nos conciernen en este momento son **#include**, **#define**.

Ejemplo típico

```
#include <stdio.h>

int main (void){
    printf ("; Hola Mundo !\n");
    return 0;
}
```

Ejemplo típico

```
#include <stdio.h>

int main (void){
    printf ("; Hola Mundo !\n");
    return 0;
}
```

En este ejemplo se incluye todo lo que hay en la librería `stdio.h`, así mismo nosotros podemos crear librerías, e incluirlas de esta manera solo que en lugar de los signos `<>` usaremos comillas dobles `"mi_libreria.h"`.

Ejemplo

```
#include <stdio.h>
#include "mi.c"
```

mi.c

```
using namespace std;
int main(){
    for (int n=0; n < 10 ; n++){
        printf("El valor de n es: %d\n",n);
    }
    return 0;
}
```

Cuando se indica `<archivo>` se le dice al compilador que busque donde están los archivos incluidos o “include” del sistema. Usualmente los sistemas con UNIX guardan los archivos en el directorio */usr/include*.

Si se usa la forma “archivo” es buscado en el directorio actual, es decir, donde el programa esta siendo ejecutado.

Los archivos incluidos usualmente contienen los prototipos de las funciones y las declaraciones de los archivos cabecera (header files) y no tienen código de C (algoritmos).

#define

La directiva **#define** se usa para definir constantes o cualquier sustitución de macro.

#define

La directiva **#define** se usa para definir constantes o cualquier sustitución de macro.

Su formato es el siguiente:

```
#define <nombre de macro> <nombre de reemplazo>
```

#define

La directiva **#define** se usa para definir constantes o cualquier sustitución de macro.

Su formato es el siguiente:

```
#define <nombre de macro> <nombre de reemplazo>
```

Ejemplo

```
#define FALSO 0  
#define VERDADERO 1
```

#define

La directiva **#define** se usa para definir constantes o cualquier sustitución de macro.

Su formato es el siguiente:

```
#define <nombre de macro> <nombre de reemplazo>
```

Ejemplo

```
#define FALSO 0  
#define VERDADERO 1
```

Observación

La directiva **#define** tiene otra poderosa característica: el nombre de macro puede tener argumentos. Cada vez que el compilador encuentra el nombre de macro, los argumentos reales encontrados en el programa reemplazan los argumentos asociados con el nombre de la macro.

Hay dos tipos de macros: las que son como objetos y las que son como funciones. Las que se asemejan a funciones toman parámetros mientras que las que se asemejan a objetos no. La forma de definir un identificador como una macro de cada tipo es, respectivamente:

Sintaxis de **define**

```
#define <identificador> <lista de tokens a reemplazar>
```

```
#define <identificador>(<lista de parámetros>) <lista de tokens a reemplazar>
```

Ejemplo de define

```
#include <stdio.h>

#define PI 3.14159

using namespace std;

int main(){
    printf("El valor de PI es: %d\n",PI);
    return 0;
}
```

Ejemplo de define

```
#include <stdio.h>

#define f(x) x*x+2

using namespace std;

int main(){
    printf("El valor de f(3) es: %d\n",f(3));
    return 0;
}
```

Ejemplo típico de define

```
#include <stdio.h>

#define REP(i,n) for(int i=0;i<n;i++)

using namespace std;
int main(){
    REP(i,10){
        printf("El valor de n es: %d\n",n);
    }
    return 0;
}
```


-  Programming Principles and Practice Using C++, Bjarne Stroustrup.
-  <http://www.codeblocks.org>
-  <http://www.wxwidgets.org>
-  (O'Reilly) Practical C Programming (3rd Edition)