

Sesión 01 - 26 febrero, 2018

Introducción

- Las tutorías son en horario de clase. Contactar via Skype con él: l.castillovidal@correo.ugr.es (comprobarlo)
- Usar Linux.
- Veremos cómo programar en la **lista de comandos**.
 - Compilación en `g++`.
 - Gestión compleja con `make`.
 - Gestión de bibliotecas: `ar`.
 - Depurador: `gdb` (no lo veremos en sí).
- Usaremos luego **NetBeans**, mucho más potente que **DevC++**, sólo para Linux (entorno de desarrollo que necesita herramientas de Linux).
 - Escrito en Java.
 - Uno de los IDE más profesionales.
 - Lo veremos la 3ª-4ª semana. Podremos entregar proyectos vía línea de comandos o NetBeans.
- Veremos otros entornos: **Valgrind**, entorno de validación del uso de memoria de los programas (pondremos énfasis en el uso de memoria).
- Usaremos **Doxygen** para la documentación de los proyectos.

Directivas del precompilador

Todas las sentencias que comienzan por `#` son **directivas del precompilador**.

```
#include <iostream>
using namespace std;

#define MSG "Hola mundo"

int main() {
    cout << MSG << endl;
}
```

`define` sustituye cualquier ocurrencia de `MSG` por su contenido. **Constantes simbólicas o macros**. Esto no ocupa memoria en la pila del programa, puesto que se sustituye antes de llegar

al compilador.

```
#include <iostream>
using namespace std;

#define SPANISH

#ifdef SPANISH
    #define MSG "Hola mundo"
#else
    #define MSG "Hello world"
#endif

int main() {
    cout << MSG << endl;
}
```

Código para compilar

```
$ g++ -o hw hw.cpp
```

Pasar al precompilador:

```
$ g++ -E hw.cpp
```

El `#include`

hw2.cpp

```
#include <iostream>
#include "hw2.h"
using namespace std;

int main() {
    cout << MSG << endl;
}
```

hw2.h

```
#define SPANISH
```

```
#ifndef SPANISH
#define MSG "Hola mundo"
#else
#define MSG "Hello world"
#endif
```

La directiva `#include "hw2.h"` borra la línea e incluye ahí el fichero.

Los ***includes*** buscan el fichero y lo sustituyen. Al poner `<>` no busca en la carpeta en la que estamos, sino en la carpeta del sistema.

El precompilador no deja nada de rastro de MSG, simplemente sustituye. Si declarásemos un `const int`, sí aparecería en el precompilador.

Doxygen

Es una forma especial de comentar los ficheros `.cpp` y los ficheros `.h`. Simplemente con hacer `doxygen <archivo .doxygen>` genera en HTML y en LaTeX toda la documentación del programa.

Programación separada

El `using namespace` **nunca** deberíamos colocarlo en el `.h`.

Uso de `#include`

Es un mecanismo de seguridad para cargarlo una sola vez, en el primer `include`. Si se llama otra vez se ignorará, que es lo que hace la estructura que aparece a continuación.

```
#ifndef _PUNTO_H
#define _PUNTO_H
...
#endif
```