# Inclusión condicional de ficheros de cabecera. Directivas #ifndef, #define, #endif

Eva Lucrecia Gibaja Galindo

Dpto. Informática y Análisis Numérico

# Introducción. El preprocesador de C

- El preprocesador es una herramienta que *filtra* el código fuente antes de ser compilado.
- Acepta como entrada código fuente y se encarga de:
  - Eliminar los comentarios.
  - Interpretar y procesar las directivas de preprocesamiento.
    - Estas directivas que resultan una herramienta sumamente útil al programador.
    - Las directivas comienzan *siempre* por el símbolo #.

# Introducción. El preprocesador de C

- Directivas más importantes del preprocesador de C:
  - *#define*: Creación de:
    - Constantes simbólicas.
    - Macros funcionales.
  - *#undef*: Eliminación de constantes simbólicas.
  - *#include*: Inclusión de ficheros.
  - *#if (#else, #endif)*: Inclusión condicional de código.

**Para más información sobre el preprocesador consultar el material adicional disponible en moodle**

# Inclusión de ficheros

- La directiva *#include* hace que se incluya el contenido del fichero especificado en la posición en la que se encuentra la directiva.
  - Permite realizar la inclusión de ficheros de cabecera de otros módulos y/o bibliotecas.
- Especificación del nombre del fichero:
  - #include <fichero>. Indica que *fichero* se encuentra:
    - En alguno de los directorios de ficheros de cabecera del sistema.
    - Entre los especificados con la opción -I del compilador.
  - #include "fichero“. Indica que *fichero* se encuentra en el directorio donde se realiza la compilación.

# Inclusión condicional de ficheros de cabecera

- Objetivo. Prevenir la inclusión repetida de un fichero de cabecera en un mismo fichero fuente.

***fichero.c***

```
#include “cabecera1.h”
#include “cabecera2.h”
```

***cabecera2.h***

```
#include “cabecera1.h”
```

- El resultado es que el contenido de *cabecera1.h* se copia dos veces, dando lugar a errores por definición múltiple.

# Inclusión condicional de ficheros de cabecera

- Solución:
  - En nuestros .h

***sofrito.h***

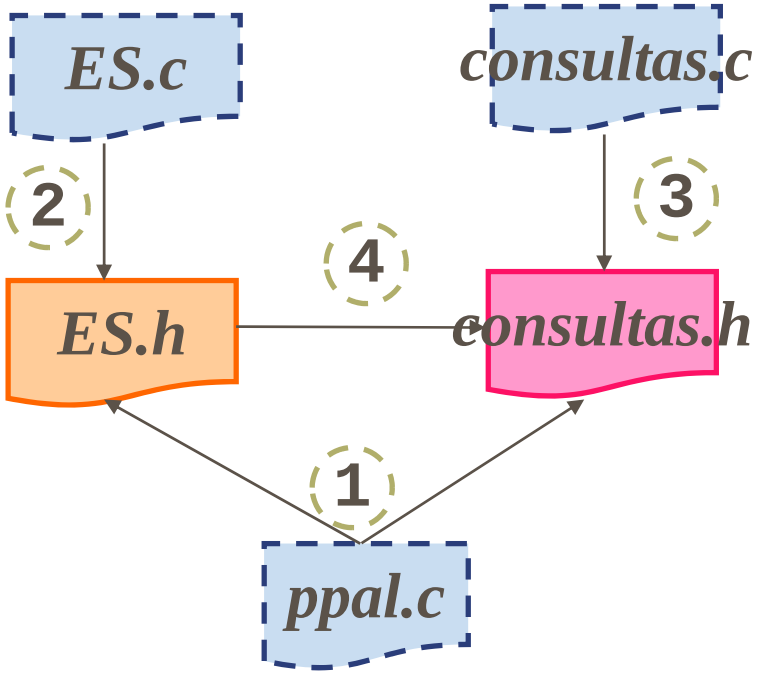
```
#ifndef CTE_SIMBOLICA_SOFRITO
#define CTE_SIMBOLICA_SOFRITO
   //Contenido de sofrito.h

#endif
```

- La constante simbólica evita que el fichero de cabecera se incluya varias veces en el programa que lo usa.
  - Cuando se incluye la primera vez, *CTE_SIMBOLICA* no está definida ⇒
    - *#ifndef (CTE_SIMBOLICA)* es cierto.
    - Se realiza *#define CTE_SIMBOLICA*.
    - Se incluye el resto del fichero de cabecera.
  - Cuando se intenta incluir de nuevo, *CTE_SIMBOLICA* ya está definida ⇒
    - *#ifndef (CTE_SIMBOLICA)* es falso.
    - El preprocesador salta a la línea siguiente al predicado *#endif*.
    - Si no hay nada tras este predicado el resultado es que no incluye nada.

# Ejemplo

1. ppal.c necesita:
   - prototipos: ES.h, consultas.h
   - struct: consultas.h
2. ES.c necesita:
   - prototipos: ES.h,
3. Consultas.c necesita:
   - prototipos : consultas.h
   - struct: consultas.h
4. ES.h neceista:
   - struct: consultas.h



# Ejemplo

**El preprocesador sustituye la linea "*#include consultas.h*" por el contenido del fichero**

***consultas.h***

```
#define MAX_NOMBRE 20
struct alumno
{
  char nombre[MAX_NOMBRE];
  char DNI[10];
  float nota;
};
float calculaMedia(struct alumno Alumnos[], int tope);
void mostrarSuperanNota(struct alumno Alumnos[], int tope, float nota);
```

***consultas.c***

```
#include "consultas.h"
float calculaMedia(struct alumno Alumnos[], int tope){...}
void mostrarSuperanNota(struct alumno Alumnos[], int tope, float nota){...}
```

# Ejemplo

**El preprocesador sustituye la linea #include "consultas.h" por el contenido del fichero**

*ES.h*

```
#include "consultas.h"
void introducirDatos(struct alumno Alumnos[], int* tope);
void mostrarAlumnos(struct alumno Alumnos[], int tope);
```

*ES.c*

```
#include "ES.h"
void introducirDatos(struct alumno Alumnos[], int* tope){...}
void mostrarAlumnos(struct alumno Alumnos[], int tope){...}
```

# Ejemplo

**Al hacer los #include, se incluye dos veces el mismo fichero**

*ppal.c*

```
#include <stdio.h>
#define MAX_ALUMNOS 30

#include "consultas.h“

#include "es.h“

int menu(){..}
void main(){..}
```

```
#define MAX_NOMBRE 20
struct alumno
{
  char nombre[MAX_NOMBRE];
  char DNI[10];
  float nota;
};
float calculaMedia(struct alumno Alumnos[], int tope);
void mostrarSuperanNota(struct alumno Alumnos[], int tope, float nota);
```

```
#include "consultas.h“
    #define MAX_NOMBRE 20
    struct alumno
    {
      char nombre[MAX_NOMBRE];
      char DNI[10];
      float nota;
    };
    float calculaMedia(struct alumno Alumnos[], int tope);
    void mostrarSuperanNota(struct alumno Alumnos[], int tope, float
    nota);
void introducirDatos(struct alumno Alumnos[], int* tope);
void mostrarAlumnos(struct alumno Alumnos[], int tope);
```

# Solución

*consultas.h*

```
#ifndef CONSULTAS
#define CONSULTAS
#define MAX_NOMBRE 20
struct alumno
{
  char nombre[MAX_NOMBRE];
  char DNI[10];
  float nota;
};
float calculaMedia(struct alumno Alumnos[], int tope);
void mostrarSuperanNota(struct alumno Alumnos[], int tope, float nota);
#endif
```

*ES.h*

```
#ifndef ES_H
#define ES_H
#include "consultas.h"
void introducirDatos(struct alumno Alumnos[], int* tope);
void mostrarAlumnos(struct alumno Alumnos[], int tope);
#endif
```

# Solución

**Al estar ya definido CONSULTAS, no se incluye otra vez el fichero**

*ppal.c*

```
#include <stdio.h>
#define MAX_ALUMNOS 30

#include "consultas.h“

#include "es.h“

int menu(){..}
void main(){..}
```

```
#define CONSULTAS
#define MAX_NOMBRE 20
struct alumno
{
  char nombre[MAX_NOMBRE];
  char DNI[10];
  float nota;
};
float calculaMedia(struct alumno Alumnos[], int tope);
void mostrarSuperanNota(struct alumno Alumnos[], int tope, float nota);
```

```
#define ES_H
#include "consultas.h“
void introducirDatos(struct alumno Alumnos[], int* tope);
void mostrarAlumnos(struct alumno Alumnos[], int tope);
```