

## **Tema 8:**

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

# 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 procesamiento
    - Estas directivas resultan una herramienta sumamente útil para el programador.
    - Todas las directivas comienzan por el símbolo #

# 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
  - ◆ `#ifndef` (`#else`, `#endif`): inclusión condicional del código.

# 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 inclusión de ficheros de cabecera de otros módulos y/o bibliotecas.
- Especificación del nombre del fichero
  - ◆ ***#include <fichero.h>***. Indica que el *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.h"***. Indica que el *fichero* se encuentra en el directorio donde se realiza la compilación



# Ejemplo



## 1. ppal.c necesita:

- ♦ prototipos: ES.h, consultas.h
- ♦ tipos: consultas.h

## 2. ES.c necesita:

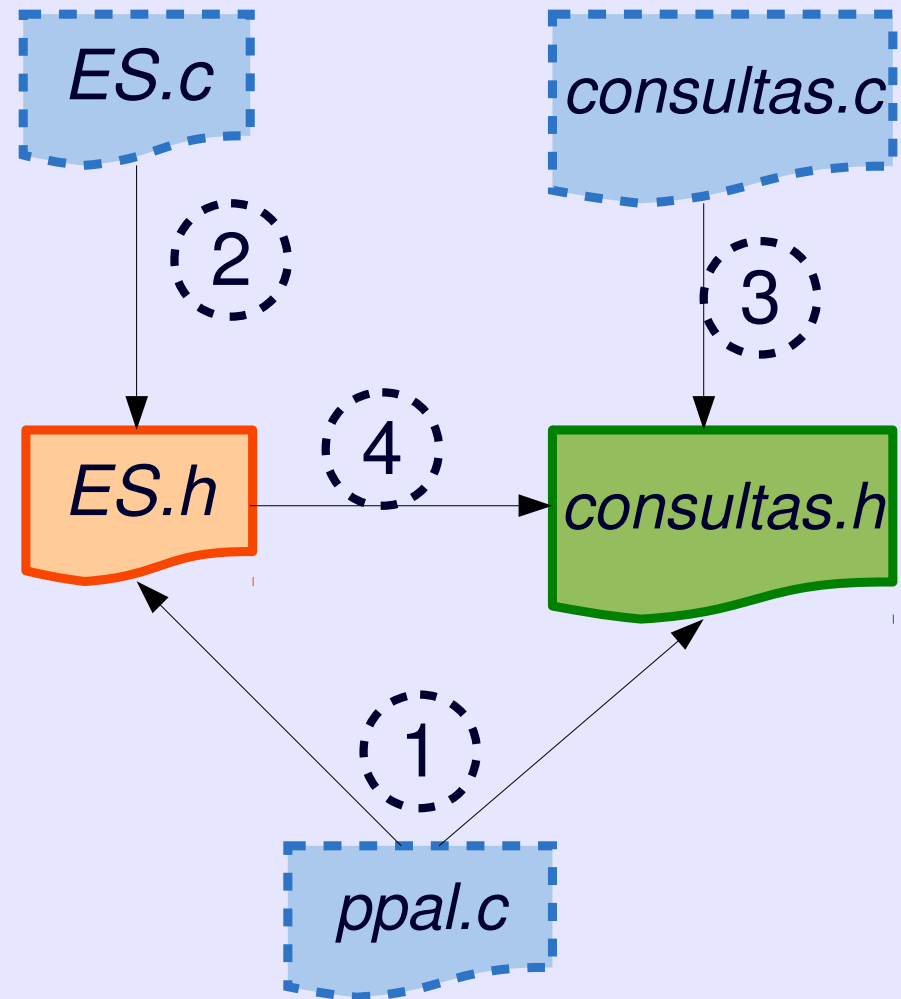
- ♦ prototipos: ES.h

## 3. consulta.c necesita:

- ♦ prototipos: consultas.h
- ♦ tipos: consultas.h

## 4. ES.h necesita:

- ♦ tipos: consultas.h



# Ejemplo



El preprocesador sustituye la línea `#include` “consultas.h” por el contenido del fichero

*consultas.h*

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

*consultas.c*

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

# Ejemplo



El preprocesador sustituye la línea `#include "consultas.h"` por el contenido del fichero

*ES.h*

```
#include "consultas.h"

void introducirDatos(stAlumno Alumnos[], int *
tope);
void mostrarAlumnos(stAlumno Alumnos[], int tope);
```

*ES.c*

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

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

*#include "consultas.h"*

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

```
void introducirDatos (...);
void mostrarAlumnos (...);
```



# 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.

- **Solución**

- ◆ En nuestros .h

```
#ifndef CTE_SIMBOLICA  
#define CTE_SIMBOLICA  
    //contenido .h  
#endif
```

# Inclusión condicional de ficheros de cabecera



- La constante simbólica evita que el fichero de cabecera se incluya varias veces en el programa que lo usa.
  - ◆ Cuando se incluya 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

# Solución



Utilizar directivas para el preprocesador

*consultas.h*

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



# Solución



*ES.h*

```
#ifndef ES_H
#define ES_H
    #include "consultas.h"

    void introducirDatos(stAlumno Alumnos[], int *tope);
    void mostrarAlumnos(stAlumno Alumnos[], int tope);
#endif
```

# Solución



*ppal.c*

Al estar ya definido CONSULTAS\_H, no se incluye otra vez en el fichero

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

#include "consultas.h"
```

```
#include "ES.h"
```

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

```
#define ES_H
#include "consultas.h"

void introducirDatos (...);
void mostrarAlumnos (...);
```

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