

Sistemas Operativos

sesión 4: aspectos del lenguaje C

Grado en Ingeniería Informática
Universidad Carlos III de Madrid

Contenidos



- Tiras de caracteres o *strings*
- Bibliotecas

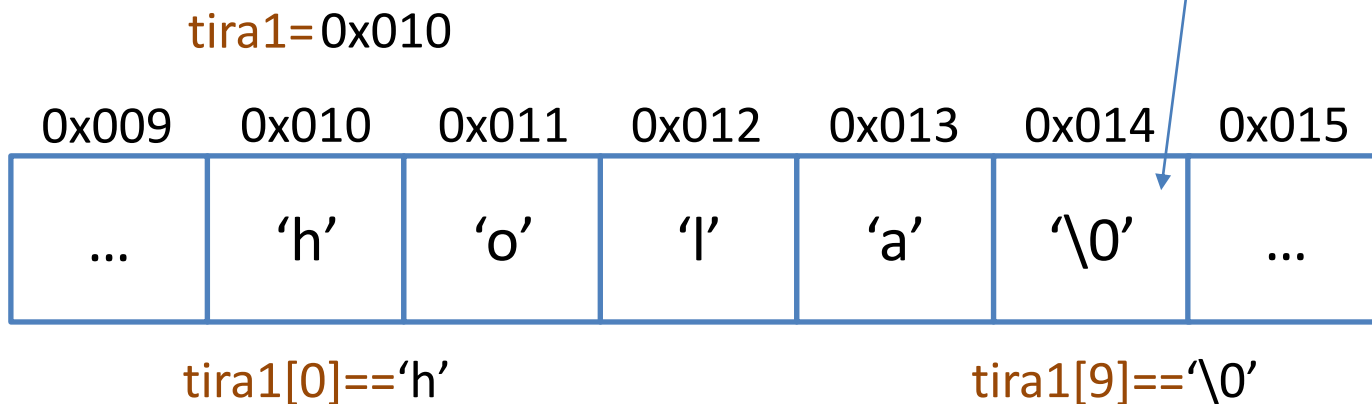
Contenidos



- **Tiras de caracteres o *strings***
- **Bibliotecas**

Tiras de caracteres (*strings*)

- `char tira1[5];`
`strcpy(tira1,"hola");`
 - Tira de caracteres con 5 caracteres (0 a 4), de los cuales uno se usará para el fin de cadena.



Tiras de caracteres (*strings*)

- `char *tira2;`
 - Puntero que podrá apuntar al comienzo de una tira de caracteres:
 - `tira2 = tira1 ;`
 - Existente: espacio de memoria de otra variable
 - `tira2 = malloc(200*sizeof(char)) ;`
`strcpy(tira2,"hola mundo\n") ;`
 - Nueva: reserva de espacio propio de memoria en tiempo de ejecución

string.h

- `strcpy(destino,origen) ;`
 - Copia el contenido de memoria de origen a destino.
- `strcat(destino,origen);`
 - Añade el contenido de origen al final de destino.
- `strlen(cadena);`
 - Indica el número de caracteres de origen (**sin** considerar el carácter reservado para fin de *string*).
- `strcmp(cadena,otraCadena);`
 - Compara dos cadenas alfabéticamente ('A'!='a').

Contenidos



- Tiras de caracteres o *strings*
- **Bibliotecas**

Declaración vs definición

declaración

- **extern** int g1 ;
int funcion1 (int p1, char p2) ;
 - Se indica los tipos usados y el nombre
PERO no se reserva memoria ni se define su valor.

definición

- int g1 = 10 ;
int funcion1 (int p1, char p2) { ... }
 - Se indica los tipo usa y el nombre,
y además se reserva espacio y define valor.

Bibliotecas: versión inicial

declaraciones

```
extern int g1 ;  
  
int funcion1( int p1,  
             char p2 );
```

mi.h

Bibliotecas: versión inicial

declaraciones

```
mi.h  
extern int g1 ;  
  
int funcion1( int p1,  
             char p2 );
```

definiciones

```
mi.c  
int g1 = 10 ;  
  
int funcion1( int p1,  
             char p2 )  
{  
    return p1+(int)p2 ;  
}
```

Bibliotecas: versión inicial

declaraciones

```
mi.h  
extern int g1 ;  
  
int funcion1( int p1,  
             char p2 );
```

definiciones

```
mi.c  
int g1 = 10 ;  
  
int funcion1( int p1,  
             char p2 )  
{  
    return p1+(int)p2 ;  
}
```

```
main.c  
#include "mi.h"  
#include <stdio.h>  
  
int main ( int argc,  
          char *argv[] )  
{  
    int r ;  
  
    r=funcion1(5,'0') ;  
    printf("r=%d\n",r) ;  
    return 0 ;  
}
```

Bibliotecas: versión inicial

declaraciones

```
extern int g1 ;  
  
int funcion1( int p1,  
             char p2 );
```

mi.h

definiciones

```
int g1 = 10 ;  
  
int funcion1( int p1,  
             char p2 )  
{  
    return p1+(int)p2 ;  
}
```

mi.c

```
#include "mi.h"  
#include <stdio.h>  
  
int main ( int argc,  
          char *argv[] )  
{  
    int r ;  
  
    r=funcion1(5,'0') ;  
    printf("r=%d\n",r) ;  
    return 0 ;  
}
```

main.c

Bibliotecas: usuario vs sistema

declaraciones

```
mi.h  
extern int g1 ;  
  
int funcion1( int p1,  
             char p2 );
```

definiciones

```
mi.c  
int g1 = 10 ;  
  
int funcion1( int p1,  
             char p2 )  
{  
    return p1+(int)p2 ;  
}
```

bibliotecas de sistema o de usuario

bibliotecas de sistema

```
#include "mi.h"  
#include <stdio.h>  
  
int main ( int argc,  
          char *argv[] )  
{  
    int r ;  
  
    r=funcion1(5,'0') ;  
    printf("r=%d\n",r) ;  
    return 0 ;  
}
```

Bibliotecas: compilación y enlazado

declaraciones

```
mi.h  
extern int g1 ;  
  
int funcion1( int p1,  
             char p2 );
```

definiciones

```
mi.c  
int g1 = 10 ;  
  
int funcion1( int p1,  
             char p2 )  
{  
    return p1+(int)p2 ;  
}
```

```
main.c  
#include "mi.h"  
#include <stdio.h>  
  
int main ( int argc,  
          char *argv[] )  
{  
    int r ;  
  
    r=funcion1(5,'0') ;  
    printf("r=%d\n",r) ;  
    return 0 ;  
}
```

```
gcc -Wall -g -o mi.o -c mi.c  
gcc -Wall -g -o main.o -c main.c  
gcc -o main main.o mi.o
```

Bibliotecas: compilación y enlazado

declaraciones

```
mi.h  
extern int g1 ;  
int funcion1( int p1,  
             char p2 );
```

definiciones

```
mi.c  
int g1 = 10 ;  
int funcion1( int p1,  
             char p2 )  
{  
    return p1+(int)p2 ;  
}
```

```
extern int g1 ;  
int funcion1( int p1,  
            char p2 );  
... // stdio.h  
  
int main ( int argc,  
          char *argv[] )  
{  
    int r ;  
  
    r=funcion1(5,'0') ;  
    printf("r=%d\n",r) ;  
    return 0 ;  
}
```

#include indica al preprocesador: “cambiarme por el contenido del archivo”

Bibliotecas: versión completa

declaraciones

```
mi.h  
#ifndef _MI_H_  
#define _MI_H_  
extern int g1 ;  
  
int funcion1( int p1,  
              char p2 );  
#endif
```

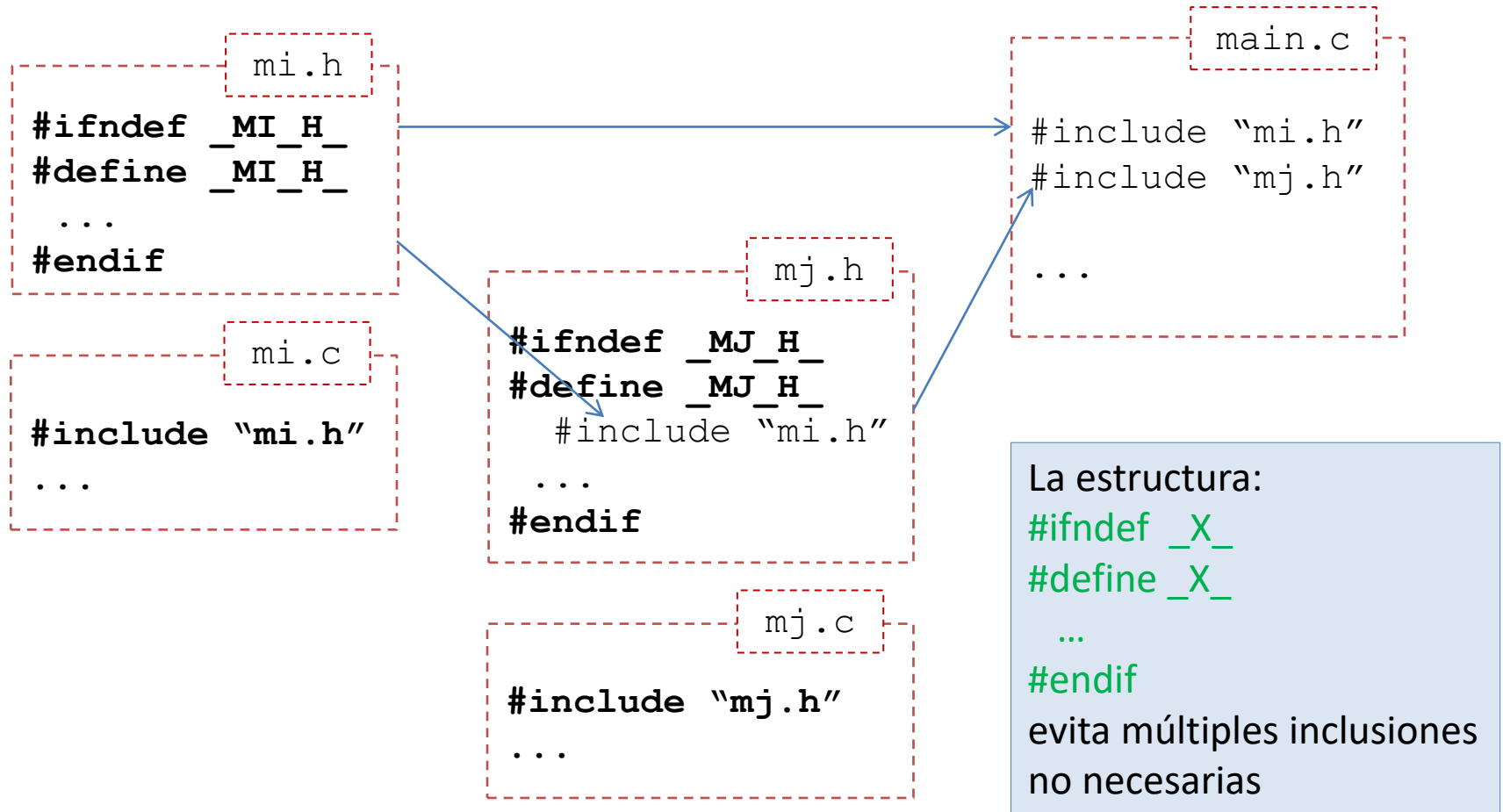
definiciones

```
mi.c  
#include "mi.h"  
  
int g1 = 10 ;  
  
int funcion1( int p1,  
              char p2 )  
{  
    return p1+(int)p2 ;  
}
```

main.c

```
#include "mi.h"  
#include <stdio.h>  
  
int main ( int argc,  
           char *argv[] )  
{  
    int r ;  
  
    r=funcion1(5,'0') ;  
    printf("r=%d\n",r) ;  
    return 0 ;  
}
```


Bibliotecas: versión completa



Sistemas Operativos

sesión 4: aspectos del lenguaje C

Grado en Ingeniería Informática
Universidad Carlos III de Madrid