

Sistemas Operativos

sesión 2: aspectos del lenguaje C

Grado en Ingeniería Informática

Universidad Carlos III de Madrid

Contenidos



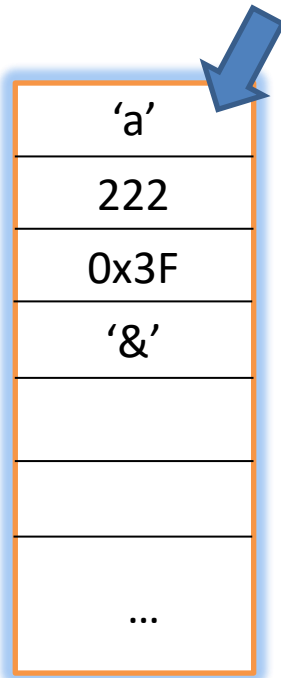
- Punteros
- Paso de parámetros

Contenidos



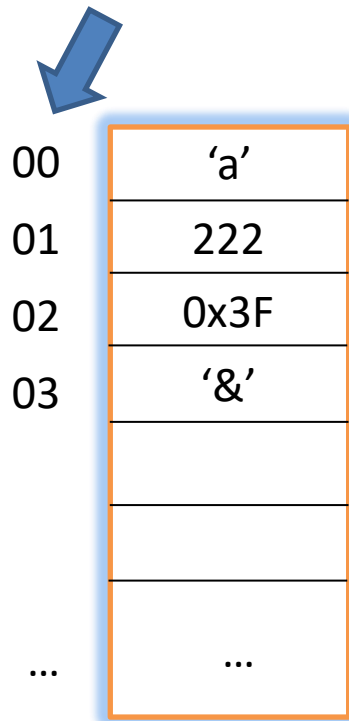
- **Punteros**
- Paso de parámetros

Dirección y contenido



- Contenido
 - Elemento guardado en una posición de memoria

Dirección y contenido



00	'a'
01	222
02	0x3F
03	'&'
...	...

- **Contenido**
 - Elemento guardado en una posición de memoria
- **Dirección**
 - Identificador de la posición de memoria
 - **Puntero** representa la dirección de un elemento de un tipo

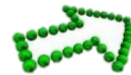
Dirección y contenido: ejemplo

```
#include <stdio.h>

int main ( int argc,
           char *argv[] )
{
    int i = 3;
    int *pi ;

    i = 3;
    pi = &i ;
    printf("%d\n", *pi) ;

    return 0;
}
```



int *

– Puntero a entero



& ...

– Dirección de



* ...

– Valor contenido en

Dirección y contenido: ejemplo

```
#include <stdio.h>

int main ( int argc,
           char *argv[] )
{
    int i = 3;
    int *pi ;

    i = 3;
    pi = &i ;
    printf("%d\n", *pi) ;

    return 0;
}
```



gcc -Wall -g -o e11 e11.c

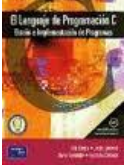
- **Wall:**
mostrar advertencias
- **g:**
añadir información de depuración
- **o:**
establecer el nombre del ejecutable



./e11

- El directorio actual (.) no está en la variable PATH

Contenidos



- Punteros
- Paso de parámetros

Paso de parámetros

```
void prueba2 ( int j, char c, float f, int pj )  
{  
    /* ... */  
}
```


Paso de parámetros

```
i = 10 ;  
float PI = 3.14 ;  
  
prueba2 ( i,      'a',      PI,      &i ) ;  
...
```

```
void prueba2 ( int j, char c, float f, int pj )  
{  
    /* ... */  
}
```

Paso de parámetros

```
i = 10 ;  
float PI = 3.14 ;  
  
prueba2 ( i,      'a',      PI,      &i ) ;  
...
```



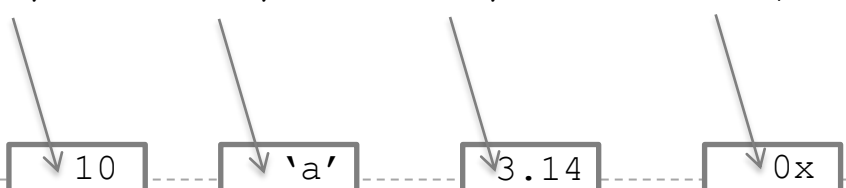
```
void prueba2 ( int j, char c, float f, int pj )  
{  
    /* ... */  
}
```

1) Se crea en pila las variables formales

Paso de parámetros

```
i = 10 ;  
float PI = 3.14 ;
```

```
prueba2 ( i,      'a',      PI,      &i ) ;  
...
```



```
void prueba2 ( int j, char c, float f, int pj )  
{  
    /* ... */  
}
```

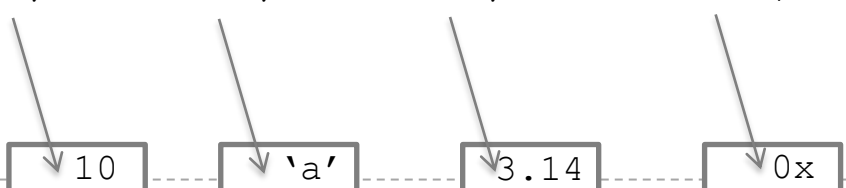
2) Se copia el valor de los parámetros reales

Paso de parámetros

Siempre se realiza una copia de los parámetros

```
i = 10 ;  
float PI = 3.14 ;
```

```
prueba2 ( i,      'a',      PI,      &i ) ;  
...
```



```
void prueba2 ( int j, char c, float f, int pj )  
{  
    /* ... */  
}
```

Paso de parámetros

Paso de parámetro
por **valor**

```
#include <stdio.h>

int main (void)
{
    int i = 10;

    /* ... */
    inc(i) ;
    /* ... */
}
```

```
void inc ( int j )
{
    j = j + 1 ;
}
```

Paso de parámetros

Paso de parámetro
por **valor**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 10;
```

```
    /* ... */
```

```
    inc(i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
i en j

10

```
void inc ( int j )
```

```
{
```

```
    j = j + 1 ;
```

```
}
```

Paso de parámetros

Paso de parámetro
por **valor**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 10;
```

```
    /* ... */
```

```
    inc(i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
i en j

10

```
void inc ( int j )
```

```
{
```

```
    j = j + 1 ;
```

```
}
```

2) se modifica
j (la copia)

Paso de parámetros

Paso de parámetro
por **valor**

Paso de parámetro
por **referencia**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 10;
```

```
    /* ... */
```

```
    inc(i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
i en j

10

```
void inc ( int j )
```

```
{
```

```
    j = j + 1 ;
```

```
}
```

2) se modifica
j (la copia)



Paso de parámetros

Paso de parámetro por **valor**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 10;
```

```
    /* ... */
```

```
    inc(i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
i en j

10

```
void inc ( int j )
```

```
{
```

```
    j = j + 1 ;
```

```
}
```

2) se modifica
j (la copia)

Paso de parámetro por **referencia**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 3;
```

```
    /* ... */
```

```
    inc(&i) ;
```

```
    /* ... */
```

```
}
```

```
void inc ( int *j )
```

```
{
```

```
    *j = *j + 1 ;
```

```
}
```

Paso de parámetros

Paso de parámetro por **valor**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 10;
```

```
    /* ... */
```

```
    inc(i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
i en j

10

```
void inc ( int j )
```

```
{
```

```
    j = j + 1 ;
```

```
}
```

2) se modifica
j (la copia)

Paso de parámetro por **referencia**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 3;
```

```
    /* ... */
```

```
    inc(&i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
&i en j

&i

```
void inc ( int *j )
```

```
{
```

```
    *j = *j + 1 ;
```

```
}
```

Paso de parámetros

Paso de parámetro por **valor**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 10;
```

```
    /* ... */
```

```
    inc(i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
i en j

10

```
void inc ( int j )
```

```
{
```

```
    j = j + 1 ;
```

```
}
```

2) se modifica
j (la copia)

Paso de parámetro por **referencia**

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int i = 3;
```

```
    /* ... */
```

```
    inc(&i) ;
```

```
    /* ... */
```

```
}
```

1) se copia
&i en j

&i

```
void inc ( int *j )
```

```
{
```

```
    *j = *j + 1 ;
```

```
}
```

2) se modifica
i a través de *j

Paso de parámetros: ejemplo

```
#include <stdio.h>

void inc ( int *j )
{
    *j = *j + 1 ;
}

int main (void)
{
    int i = 3;

    inc(&i) ;
    printf("%d\n",i) ;

    return 0;
}
```

- La función *inc* incrementa el valor pasado por referencia en *j*
- La función *main* define una variable *i*, incrementa su valor y lo imprime

Paso de parámetros: ejemplo

```
#include <stdio.h>

void inc ( int *j )
{
    *j = *j + 1 ;
}

int main (void)
{
    int i = 3;

    inc(&i) ;
    printf("%d\n",i) ;

    return 0;
}
```



gcc -Wall -g -o e12 e12.c

- **Wall:**
mostrar advertencias
- **g:**
añadir información de depuración
- **o:**
establecer el nombre del ejecutable



./e12

- El directorio actual (.) no está en la variable PATH

Paso de parámetros: ejemplo

```
#include <stdio.h>

void inc ( int *j )
{
    *j = *j + 1 ;
}

int main (void)
{
    int i = 3;
    inc( i) ;
    printf("%d\n",i) ;

    return 0;
}
```



gcc -Wall -g -o e13 e13.c

- **Wall:**
mostrar advertencias
- **g:**
añadir información de depuración
- **o:**
establecer el nombre del ejecutable



./e13

- El directorio actual (.) no está en la variable PATH

Paso de parámetros: ejemplo



```
acaldero@phoenix:/tmp$ ./e13
Violación de segmento
acaldero@phoenix:/tmp$ gdb e13
GNU gdb (Ubuntu/Linaro 7.2-1ubuntu11) 7.2
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Para las instrucciones de informe de errores, vea:
<http://www.gnu.org/software/gdb/bugs/>...
Leyendo símbolos desde /tmp/e13...hecho.
(gdb) run
Starting program: /tmp/e13

Program received signal SIGSEGV, Segmentation fault.
0x080483ca in inc (j=0x3) at e13.c:5
5          *j = *j + 1 ;
(gdb) 
```


Paso de parámetros: ejemplo

```
#include <stdio.h>

void inc ( int *j )
{
    if (j==NULL)
        printf("j=NULL\n") ;
    else *j = *j + 1 ;
}

int main (void)
{
    int i = 3;

    inc(i) ;
    printf("%d\n",i) ;

    return 0;
}
```



gcc -Wall -g -o e14 e14.c

- **Wall:**
mostrar advertencias
- **g:**
añadir información de depuración
- **o:**
establecer el nombre del ejecutable



./e14

- El directorio actual (.) no está en la variable PATH

Sistemas Operativos

sesión 2: aspectos del lenguaje C

Grado en Ingeniería Informática

Universidad Carlos III de Madrid