Sistemas Operativos

sesión 6: llamadas al sistema

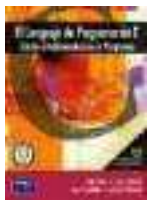
Grado en Ingeniería Informática

Universidad Carlos III de Madrid

ARCOS @ UC3M
Alejandro Calderón Mateos

Agenda

Compilación y bibliotecas



Estructuras y ficheros

Problemas en lenguaje C

Agenda


Compilación y bibliotecas
Estructuras y ficheros
Problemas en lenguaje C

Contenidos

- Proceso de compilación
- Inspección de un ejecutable/proceso
- Bibliotecas estáticas y dinámicas

Contenidos

- **Proceso de compilación**
- Inspección de un ejecutable/proceso
- Bibliotecas estáticas y dinámicas

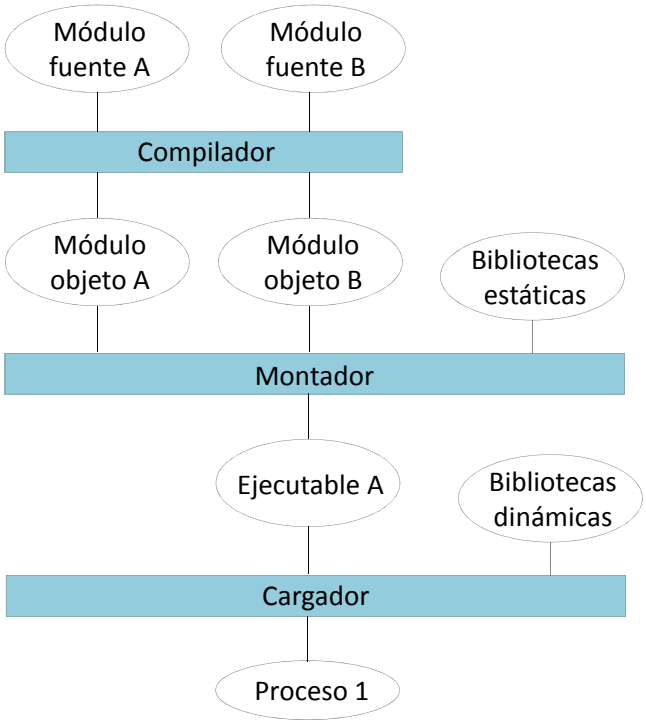
Motivación

- ¿Qué fases hay en la compilación y ejecución de un programa?

```
acaldero@phoenix:~/work$ gcc –o programa fuente.c
acaldero@phoenix:~/work$ ./programa

Cargando…
…
```

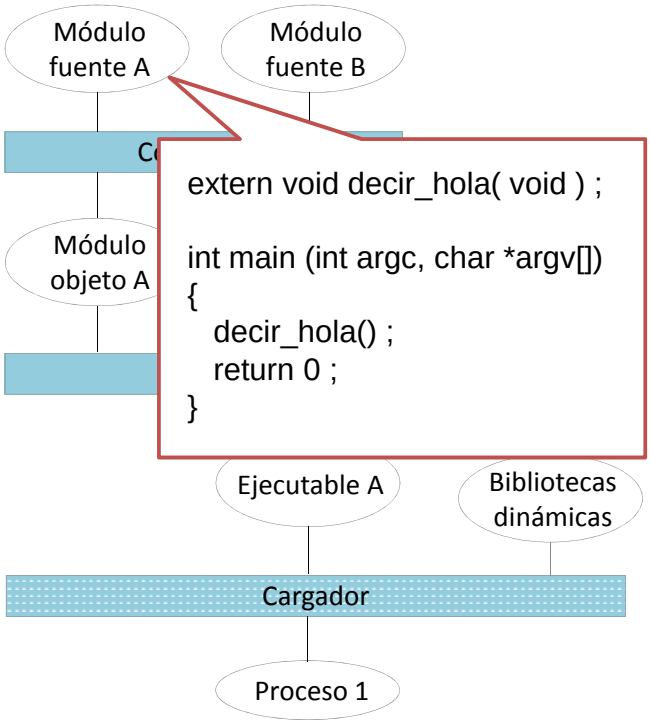
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



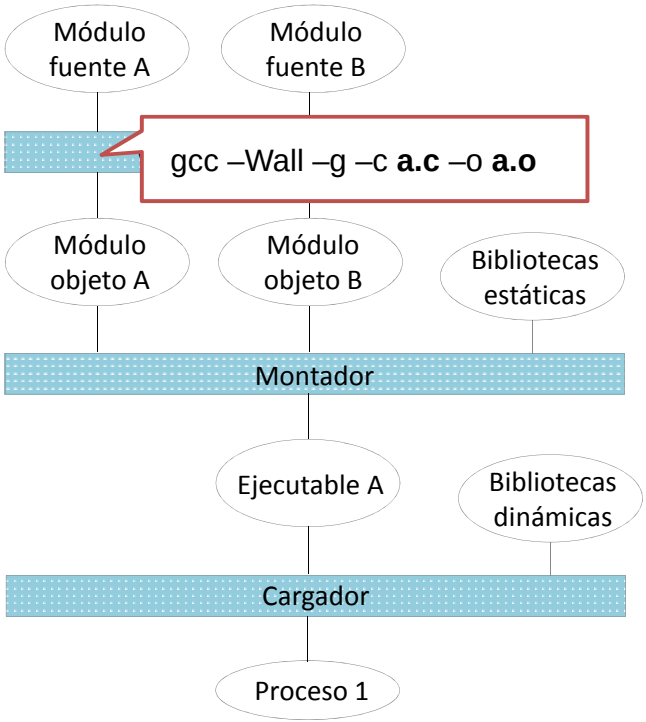
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



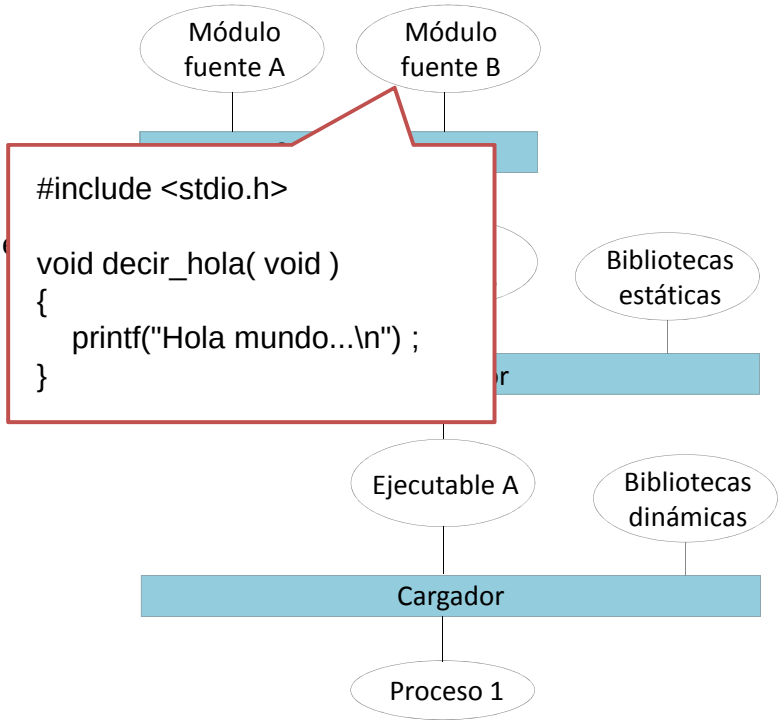
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



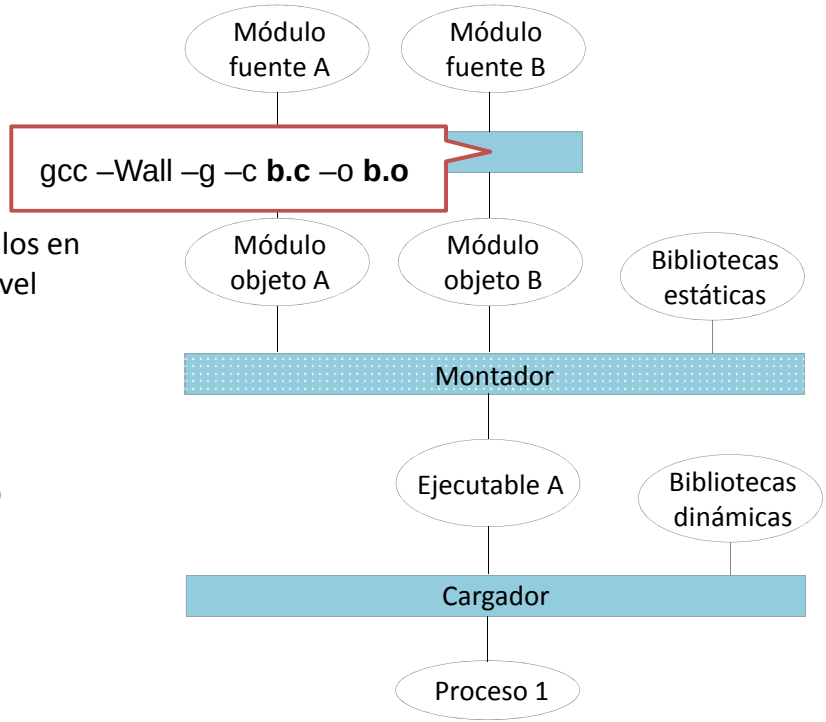
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos
    lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



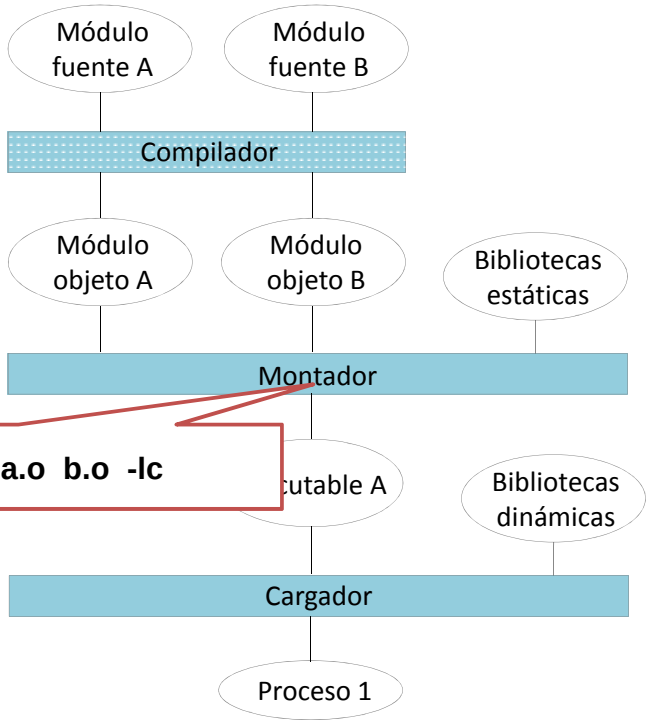
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



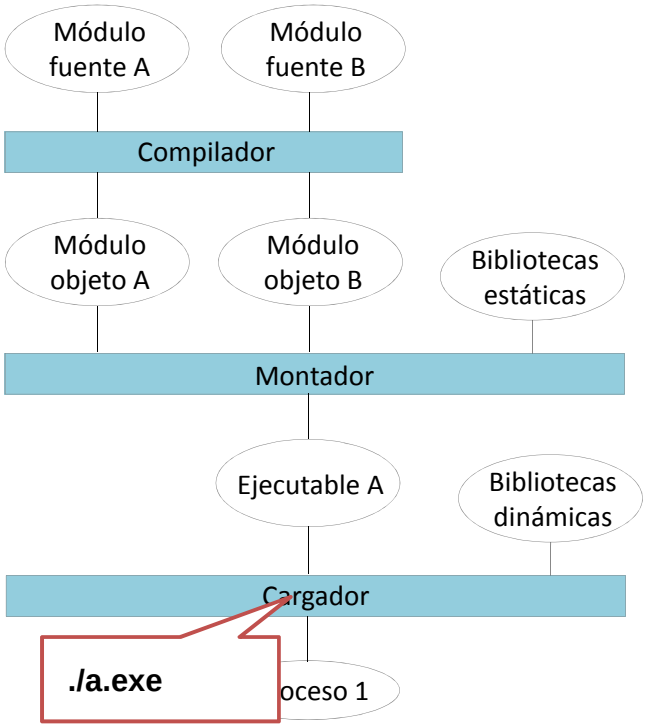
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Mont
  - Enlaz
  - Ejecución



Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



Contenidos

- Proceso de compilación
- **Inspección de un ejecutable/proceso**
- Bibliotecas estáticas y dinámicas

Motivación

- ¿Cómo saber qué llamadas al sistema hace un programa?
- ¿Qué bibliotecas usa?

```
acaldero@phoenix:~/work$ ./programa

Cargando…
…
```

Inspeccionar un ejecutable

- Dependencias de un ejecutable (lib. dinámicas):

```
acaldero@phoenix:~/infodso/$ ldd main.exe
        linux-gate.so.1 =>  (0xb7797000)
        libdinamica.so.1 => not found
        libc.so.6 => /lib/libc.so.6 (0xb761c000)
        /lib/ld-linux.so.2 (0xb7798000)
```

- Símbolos de un ejecutable:

```
acaldero@phoenix:~/infodso/$ nm main.exe
08049f20 d _DYNAMIC
08049ff4 d _GLOBAL_OFFSET_TABLE_
0804856c R _IO_stdin_used
         w _Jv_RegisterClasses
08049f10 d __CTOR_END__
08049f0c d __CTOR_LIST__
...
```

Inspeccionar un proceso

- Detalles de las secciones de un proceso:

```
acaldero@phoenix:~/infodso/$ cat /proc/1/maps
b7688000-b7692000 r-xp 00000000 08:02 1491      /lib/libnss_files-2.12.1.so
b7692000-b7693000 r--p 00009000 08:02 1491      /lib/libnss_files-2.12.1.so
b7693000-b7694000 rw-p 0000a000 08:02 1491      /lib/libnss_files-2.12.1.so
b7694000-b769d000 r-xp 00000000 08:02 3380      /lib/libnss_nis-2.12.1.so
b769d000-b769e000 r--p 00008000 08:02 3380      /lib/libnss_nis-2.12.1.so
b769e000-b769f000 rw-p 00009000 08:02 3380      /lib/libnss_nis-2.12.1.so
b769f000-b76b2000 r-xp 00000000 08:02 1414      /lib/libnsl-2.12.1.so
b76b2000-b76b3000 r--p 00012000 08:02 1414      /lib/libnsl-2.12.1.so
b76b3000-b76b4000 rw-p 00013000 08:02 1414      /lib/libnsl-2.12.1.so
b76b4000-b76b6000 rw-p 00000000 00:00 0
..
b78b7000-b78b8000 r-xp 00000000 00:00 0         [vdso]
b78b8000-b78d4000 r-xp 00000000 08:02 811       /lib/ld-2.12.1.so
b78d4000-b78d5000 r--p 0001b000 08:02 811       /lib/ld-2.12.1.so
b78d5000-b78d6000 rw-p 0001c000 08:02 811       /lib/ld-2.12.1.so
b78d6000-b78ef000 r-xp 00000000 08:02 1699      /sbin/init
b78ef000-b78f0000 r--p 00019000 08:02 1699      /sbin/init
b78f0000-b78f1000 rw-p 0001a000 08:02 1699      /sbin/init
b81e5000-b8247000 rw-p 00000000 00:00 0         [heap]
bf851000-bf872000 rw-p 00000000 00:00 0         [stack]
```

Inspeccionar un proceso

- Llamadas al sistema realizadas por un proceso:

```
acaldero@phoenix:~/infodso/$ strace ls -las
execve("/bin/ls", ["ls", "-las"], [/* 20 vars */]) = 0
brk(0)                                  = 0x8bb7000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb78c3000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=123227, ...}) = 0
mmap2(NULL, 123227, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb78a4000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/i386-linux-gnu/libselinux.so.1", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\300>\0\0004\0\0\0"..., 512) = 512
fstat64(3, {st_mode=S_IFREG|0644, st_size=104116, ...}) = 0
mmap2(NULL, 109440, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7889000
mmap2(0xb78a2000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18) =
0xb78a2000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/i386-linux-gnu/librt.so.1", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\300\30\0\0004\0\0\0"..., 512) = 512
...
```

Contenidos

- Proceso de compilación
- Inspección de un ejecutable/proceso
- **Bibliotecas estáticas y dinámicas**

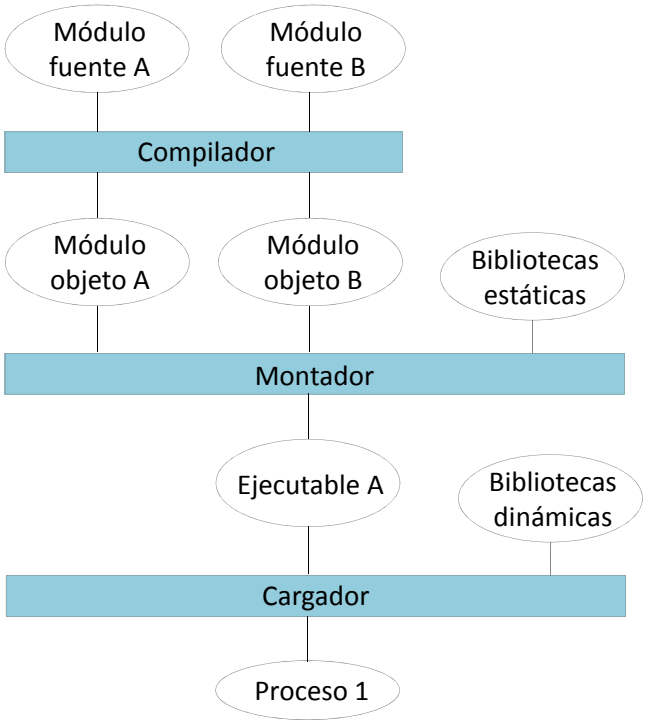
Motivación

- ¿Qué es una biblioteca estática?
- ¿Qué es una biblioteca dinámica?
- ¿Cómo se utilizan?

```
acaldero@phoenix:~/work$ gcc –o programa fuente.c libestatica1.a
acaldero@phoenix:~/work$ ./programa

Cargando…
…
```

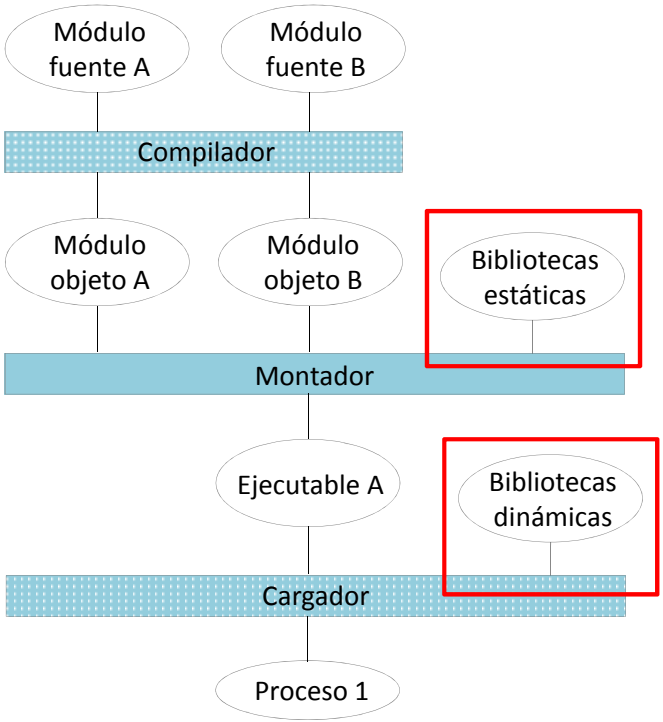
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



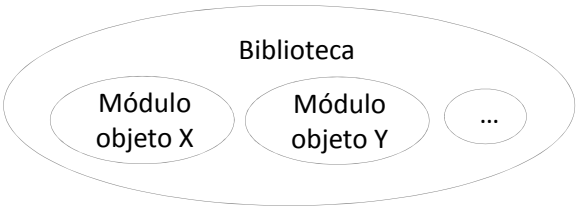
Generación y ejecución de programas

- Aplicación
  - Conjunto de módulos en lenguaje de alto nivel
- Fases:
  - Compilación
  - Montaje
  - Enlazado dinámico
  - Ejecución



Bibliotecas de objetos

- Biblioteca
  - Colección de módulos objetos relacionados



- Biblioteca estática
  - Carga y montaje en tiempo de compilación
- Biblioteca dinámica
  - Carga y montaje en tiempo de ejecución
  - Se indica al montar qué biblioteca usar, carga y montaje posterior

Bibliotecas de objet

B.C

```
#include <stdio.h>

void decir ( char * str )
{
    printf("%s",str) ;
}
```

A.C

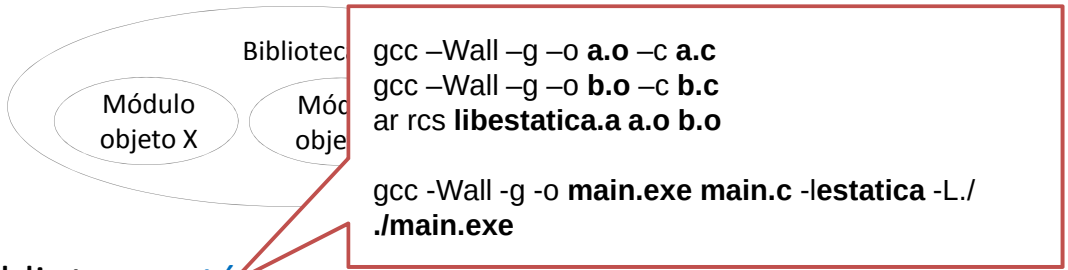
```
extern void decir ( char * str ) ;

void decir_hola( void )
{
    decir("Hola mundo...\n") ;
}
```

MAIN.C

```
extern void decir_hola( void ) ;

int main (int argc, char *argv[])
{
   decir_hola() ;
   return 0 ;
}
```

Bibliotecas de objetos

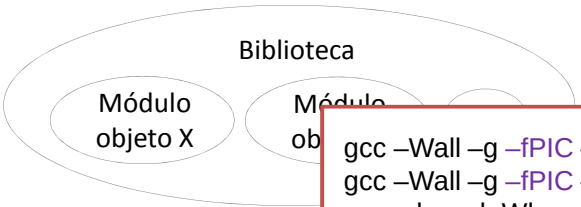
- Biblioteca
  - Colección de módulos objetos relacionados



- Biblioteca estática
  - Carga y montaje en tiempo de compilación
- Biblioteca dinámica
  - Carga y montaje en tiempo de ejecución
  - Se indica al montar qué biblioteca usar, carga y montaje posterior

Bibliotecas de objetos

- Biblioteca
  - Colección de módulos objetos relacionados

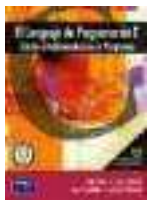


- Biblioteca estática
  - Carga y montaj
- Biblioteca dinámica
  - Carga y montaje en tiempo de ejecución
  - Se indica al montar qué biblioteca usar, carga y montaje posterior

```
gcc –Wall –g –fPIC –o a.o –c a.c
gcc –Wall –g –fPIC –o b.o –c b.c
gcc -shared -Wl,-soname,libdinamica.so \
            -o libdinamica.so.1.0  a.o  b.o
ln –s libdinamica.so.1.0 libdinamica.so

gcc –Wall –g  –o  main.exe main.c  –ldinamica –L./
env LD_LIBRARY_PATH=$LD_LIBRARY_PATH:. ./main.exe
```

Agenda

Compilación y bibliotecas



Estructuras y ficheros

Problemas en lenguaje C

Contenidos

- Estructuras
- Llamadas al sistema
- Ficheros

Contenidos

- **Estructuras**
- Llamadas al sistema
- Ficheros

Definición de estructura

```
/* tipo de datos */
struct
{
   int codigo;
   char nombre[30];
} registro ;



/* variable */
struct registro variableRegistro ;
```

Definición de estructura

```
/* tipo de datos */
struct
{
   int codigo;
   char nombre[30];
} registro ;

/* redefinición de tipo */
typedef struct registro tipoRegistro ;

/* variable */
tipoRegistro variableRegistro ;
```

Acceso a una estructura

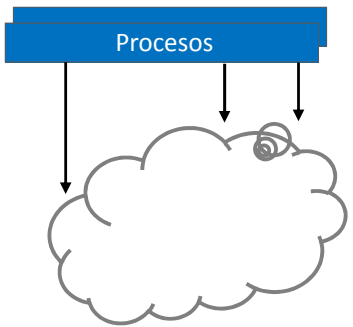
```
/* campos individuales */
variableRegistro.codigo = 3 ;
strcpy(variableRegistro.nombre,”nombre”) ;

/* estructura completa (por referencia) */
funcionEjemplo(&variableRegistro) ;

/* estructura completa (por valor) */
funcionEjemplo(variableRegistro) ;
```

Contenidos

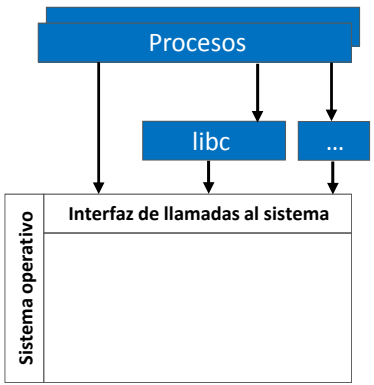
- Estructuras
- **Llamadas al sistema**
- Ficheros

Servicios del sistema

- Gestión de procesos
- Gestión de memoria
- Gestión de ficheros
- Gestión de dispositivos
- Comunicación
- Mantenimiento

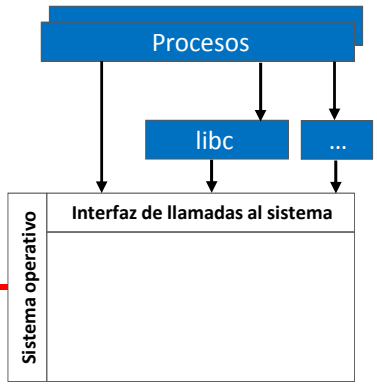


Llamadas al sistema



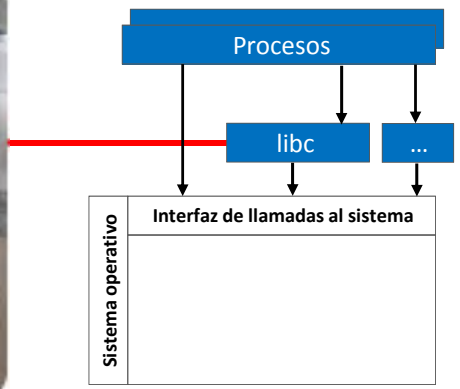
Llamadas al sistema

"servicios muy básicos de la casa"



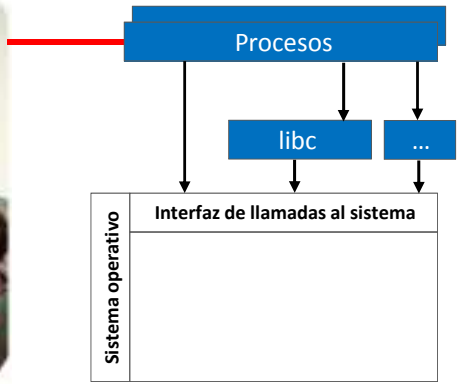
Llamadas al sistema

"servicios básicos de la casa"



Llamadas al sistema

"personas que utilizan los servicios"



Llamadas al sistema

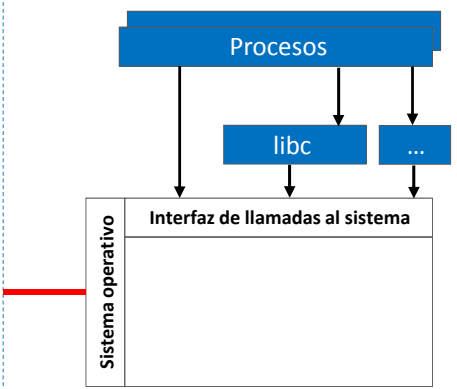
#include <unistd.h>

- int brk (void *);
- void *sbrk (intptr_t);

- int close (int);
- off_t lseek (int, off_t, int);
- ssize_t read (int, void *, size_t);
- ssize_t write (int, const void *, size_t);
- ...

#include <fcntl.h>

- int open (const char *path, int oflag, ...);
- int creat (const char *path, mode_t mode);
- ...



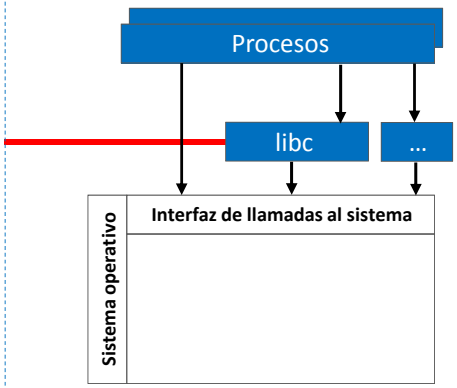
Llamadas al sistema

#include <stdlib.h>

- void *malloc (unsigned long Size);
- void *realloc (void *Ptr, unsigned long NewSize);
- void *calloc (unsigned short NItems, unsigned short SizeOfItems);
- void free (void *Ptr);
- ...

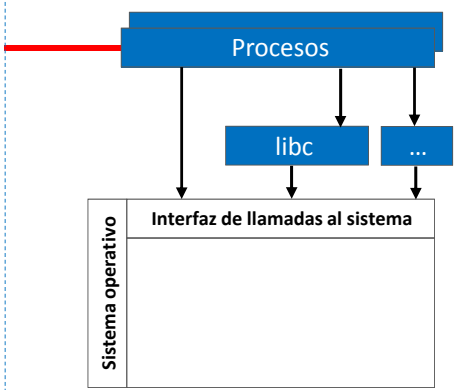
#include <stdio.h>

- FILE * fopen (const char *filename, const char *opentype);
- int fclose (FILE *stream);
- int feof(FILE *fichero);
- int fseek (FILE * stream, long int offset, int origin);
- size_t fread (void * ptr, size_t size, size_t count, FILE * f);
- int fscanf(FILE *f, const char *formato, argumento, ...);
- size_t fwrite(void *ptr, size_t size, size_t neltos, FILE *f);
- int fprintf(FILE *f, const char *fmt, arg1, ...);
- ...



Llamadas al sistema

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

int main ( int argc, char *argv[] )
{
    int *ptr1 ;
    int i ;

    ptr1 = (int *)malloc (100*sizeof(int)) ;
    for (i=0; i<100; i++)
        ptr1[i] = 10 ;
    free(ptr1);
}
```

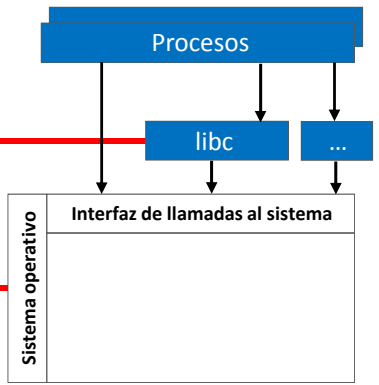


Contenidos

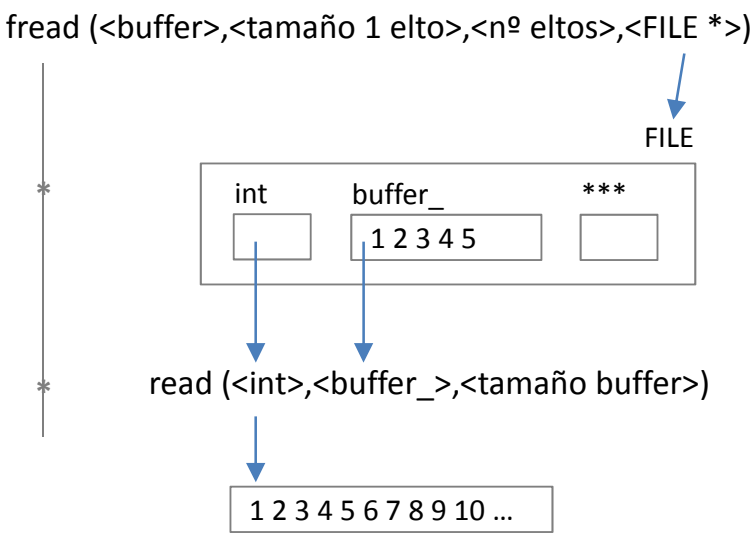
- Estructuras
- Llamadas al sistema
- **Ficheros**

Servicios para ficheros

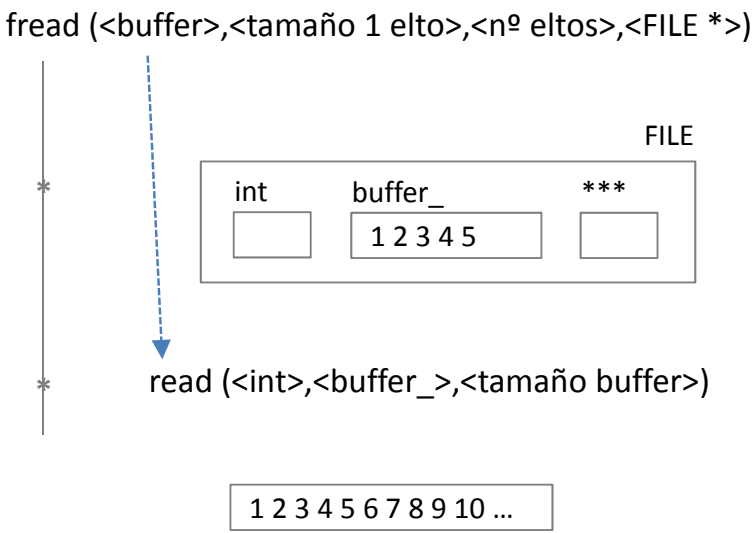
- fopen, fclose, fwrite, fread, fseek y feof
- open, close, write, read y lseek



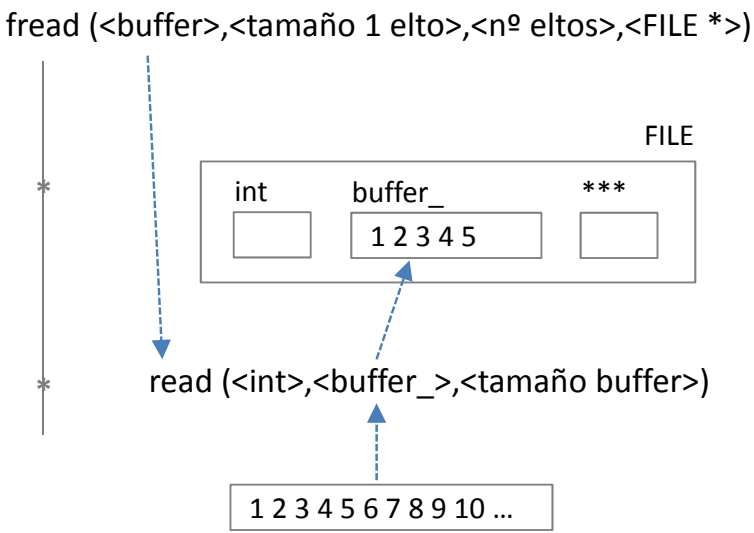
Funcionalidad extendida



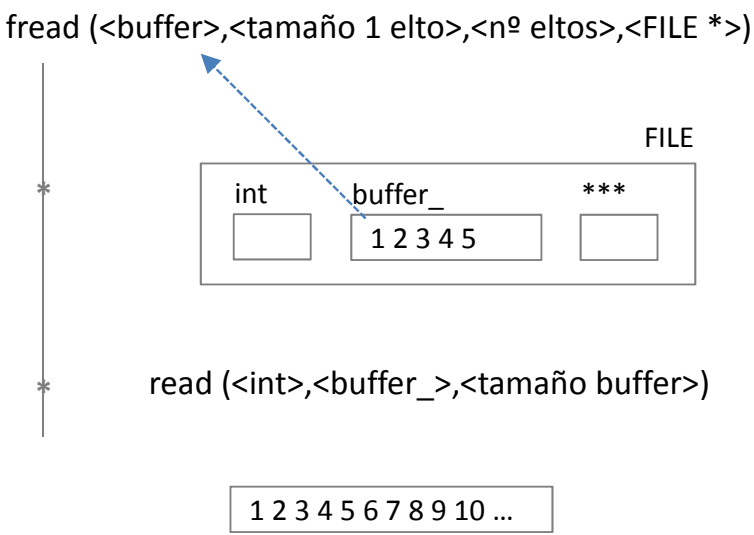
Funcionalidad extendida



Funcionalidad extendida



Funcionalidad extendida



Escritura en fichero

sistema

```c
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>

int main ( int argc, char *argv[] )
{
 int fd1 ;
 char str1[10] ;
 int nb ;

 fd1 = open ("/tmp/txt1",
             O_CREAT|O_RDWR, S_IRWXU);
 if (-1 == fd1) {
    perror("open:");
    exit(-1);
 }

 strcpy(str1,"hola");
 nb = write (fd1,str1,strlen(str1));
 printf("bytes escritos = %d\n",nb);

 close (fd1);
 return (0) ;
}
```

libc

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>


int main ( int argc, char *argv[] )
{
  FILE *fd1 ;
  char str1[10] ;
  int nb ;

  fd1 = fopen ("/tmp/txt2","w+");
  if (NULL == fd1) {
      printf("fopen: error\n");
      exit(-1) ;
  }

  strcpy(str1,"mundo");
  nb = fwrite (str1,strlen(str1),1,fd1);
  printf("items escritos = %d\n",nb);

  fclose (fd1) ;
  return (0) ;
}
```

Lectura desde fichero

sistema

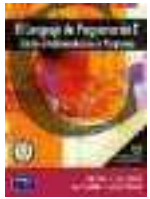
```c
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>

int main ( int argc, char *argv[] )
{
 int fd1 ;
 char str1[10] ;
 int nb, i ;

 fd1 = open ("/tmp/txt1",O_RDONLY);
 if (-1 == fd1) {
     perror("open:");
     exit(-1);
 }

 i=0;
 do {
      nb = read (fd1,&(str1[i]),1);
      i++;
 } while (nb != 0) ;
 str1[i] = '\0';
 printf("%s\n",str1);

 close (fd1);
 return (0);
}
```

libc

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>


int main ( int argc, char *argv[] )
{
 FILE *fd1 ;
 char str1[10] ;
 int nb, i ;

 fd1 = fopen ("/tmp/txt2","r");
 if (NULL == fd1) {
     printf("fopen: error\n");
     exit(-1) ;
 }

 i=0;
 do {
      nb = fread (&(str1[i]),1,1,fd1) ;
      i++ ;
 } while (nb != 0) ; /* feof() */
 str1[i] = '\0' ;
 printf("%s\n",str1);

 fclose (fd1);
 return (0);
}
```

Agenda

Compilación y bibliotecas



Estructuras y ficheros

Problemas en lenguaje C

Contenidos

- Problemas de lenguaje C:
  - Lectura y escritura de registros (estructuras)
  - Lectura y escritura de números

Contenidos

- Problemas de lenguaje C:
  - **Lectura y escritura de registros (estructuras)**
  - Lectura y escritura de números

Lectura/escritura de registros

- Realizar una biblioteca que:
  - Defina el registro tipoRegistro con dos campos:
    - *nombre*: un vector de 30 caracteres ;
    - *codigo*: un número entero de 32 bits ;
- Realizar un programa que:
  - Escriba una lista de registros tipoRegistros:
    - Si no existe el fichero, lo crea y si existe añade al final.
  - Lea una lista de registros tipoRegistros:
    - Lea e imprima todos los registros del fichero.

registro.h

```
struct
{
   int codigo;
   char nombre[30];
} registro ;

typedef struct registro tipoRegistro ;
```

escribir.c (1/3)

```
/*
 * Juan Manuel Pérez Lobato
 * ARCOS @ UC3M
 */

#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <string.h>

#include "registro.h"


int main ( int argc, char *argv[] )
{
```

escribir.c (2/3)

```
int file1 ;
tipoRegistro registro1 ;
long posicion1 ;
char nombreFich[30]="fdescreg.dat" ;


/* Abrir fichero */
  file1 = open (nombreFich,O_APPEND|O_WRONLY) ;
if (-1 != file1) {
    printf ("El fichero ya existe, añado al final\n");
}


if (-1 == file1) {
  printf ("El fichero no existe\n");
  printf ("Se va a crear el fichero\n");
    file1=open (nombreFich, O_CREAT| O_WRONLY, S_IWUSR|S_IRUSR) ;
  if (-1 == file1) {
      printf ("Error en la creación :1\n");
      exit (-1);
  }
}
```

escribir.c (3/3)

```
      /* Escribir registros */
      registro1.codigo=1;
      strcpy (registro1.nombre, "nombre uno");
      write (file1, &registro1, sizeof (registro1) );

      registro1.codigo++;
      strcpy (registro1.nombre, "nombre dos");
      write (file1, &registro1, sizeof (registro1) );

      registro1.codigo++;
      strcpy (registro1.nombre, "nombre tres");
      write (file1, &registro1, sizeof (registro1) );

      /* Imprimir la posición */
        posicion1 = lseek(file1,0,SEEK_CUR) ;
        printf ("Estoy en la posición %d del fichero\n", posicion1 );

      /* Cerrar fichero */
      close(file1);

} /* fin de main */
```

leer.c (1/2)

```
/*
 * Juan Manuel Pérez Lobato
 * ARCOS @ UC3M
 */

#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#include "registro.h"


int main ( int argc, char *argv[] )
{
```

leer.c (2/2)

```
    int file1;
    int bytes_leidos;
    char nombreFich[40]="fdescreg.dat";
    tipoRegistro registro1;


    file1 = open (nombreFich, O_RDONLY) ;
    if (file1 == -1) {
        fprintf (stderr, "No se ha podido abrir el fichero\n");
        exit (-1) ;
  }

    bytes_leidos = read (file1, &registro1, sizeof(registro1));
    while ( bytes_leidos !=0 ){
            printf  ("registro leído -> código:%d: nombre:%s:\n",
                     registro1.codigo,
                     registro1.nombre);
            bytes_leidos = read (file1, &registro1, sizeof(registro1));
    }

    close(file1);

} /* fin de main */
```

Contenidos

- Problemas de lenguaje C:
  - Lectura y escritura de registros (estructuras)
  - **Lectura y escritura de números**

Lectura/escritura de números

1. Realizar un programa que escriba los números pares del 1 al 100 en un fichero.
2. Realizar un programa que lea todos los números del fichero anterior y muestre la suma por pantalla.
3. Realizar un programa que pida una posición por teclado y muestre el numero que se encuentra en esa posición en el fichero.
4. Realizar un programa que pida una posición por teclado y un número, y sustituya el numero leído por el que actualmente existe en el fichero en esa posición.

Sistemas Operativos

sesión 6: llamadas al sistema

Grado en Ingeniería Informática

Universidad Carlos III de Madrid

ARCOS @ UC3M
Alejandro Calderón Mateos