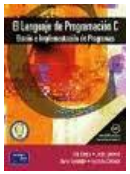
Sistemas Operativos

sesión 6: llamadas al sistema

Grado en Ingeniería Informática

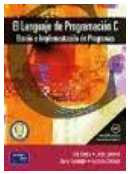
Universidad Carlos III de Madrid

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 ;
```

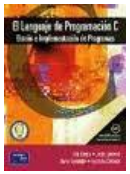
Definición de estructura

```
/* 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) ;
```

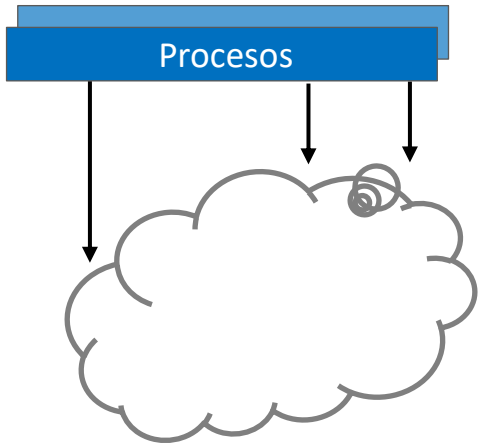
Contenidos



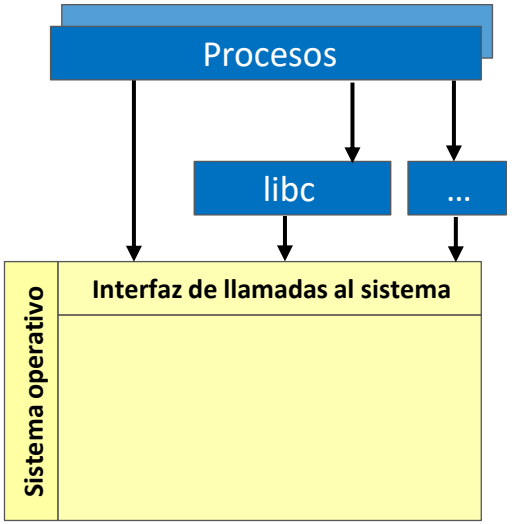
- Estructuras
- **Llamadas al sistema**
- Ficheros

Servicios del sistema

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

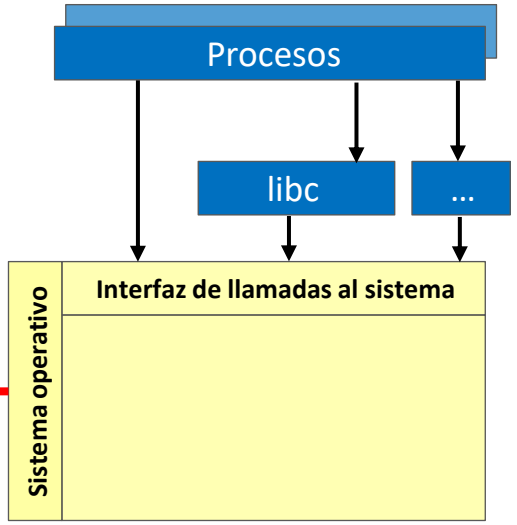


Llamadas al sistema



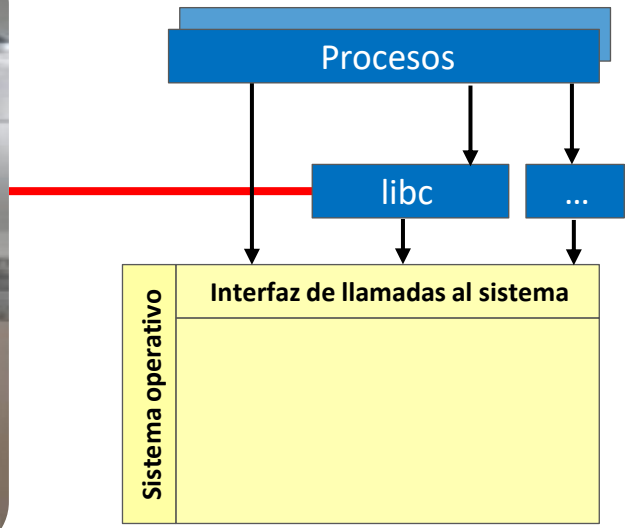
Llamadas al sistema

"servicios muy básicos de la casa"



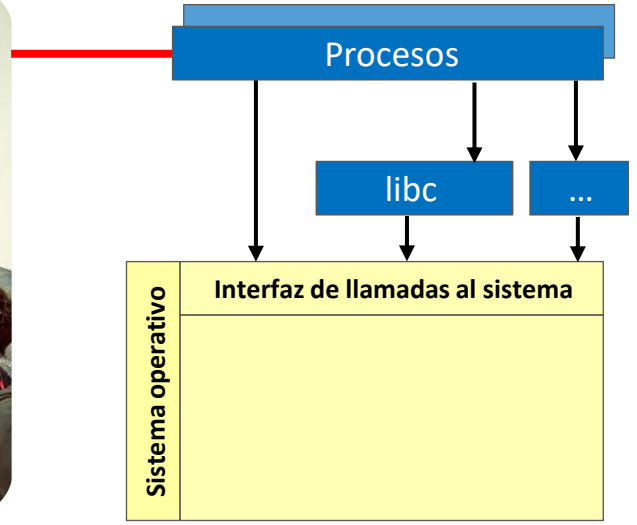
Llamadas al sistema

"servicios básicos de la casa"



Llamadas al sistema

"personas que utilizan los servicios"



Llamadas al sistema

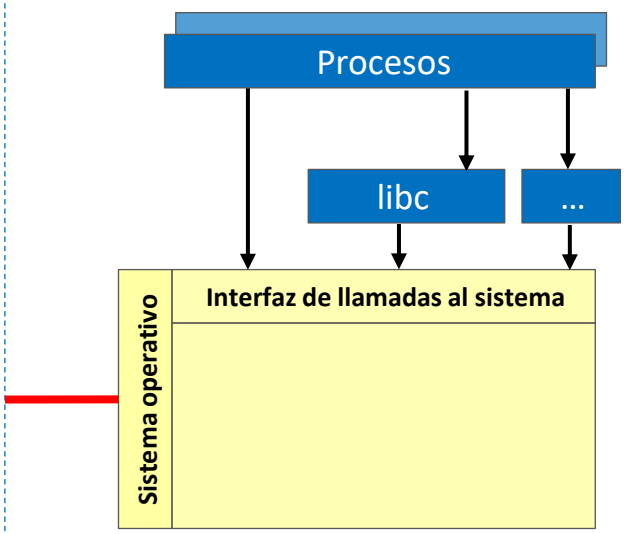
#include <unistd.h>

- 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);
- ...



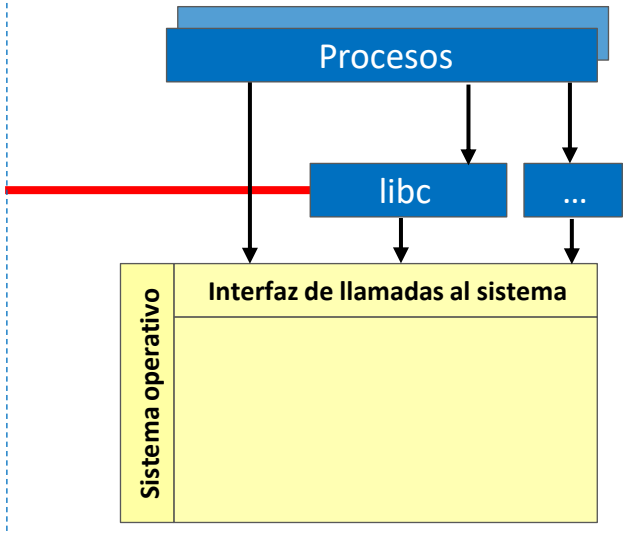
Llamadas al sistema

#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);
- ...

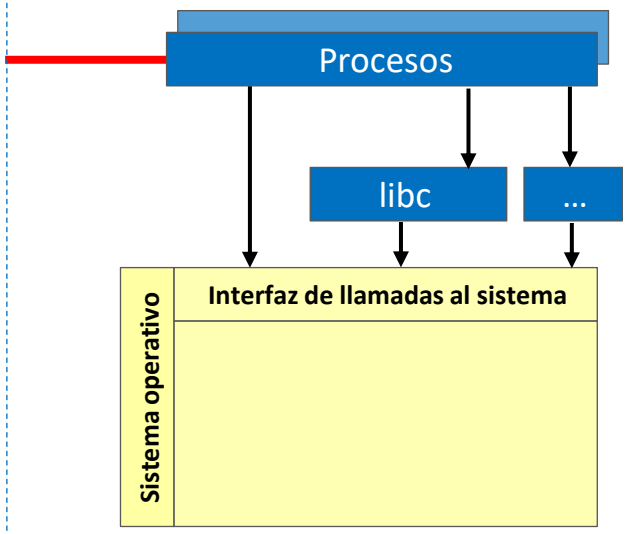
#include <stdio.h>

- 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, ...);
- ...

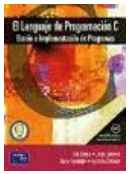


Llamadas al sistema

```
#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);
}
```



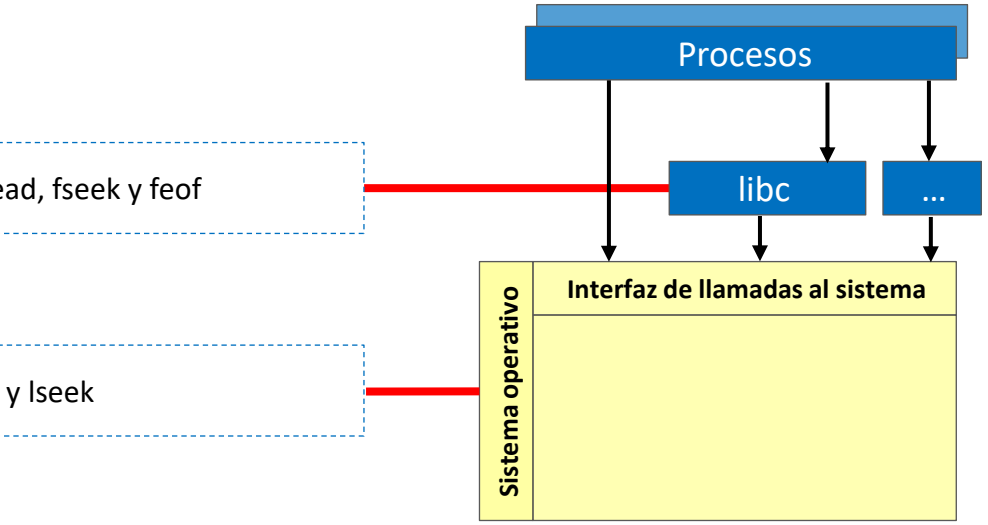
Contenidos



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

Servicios para ficheros

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



sistema vs. libc

```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) ;
}
```

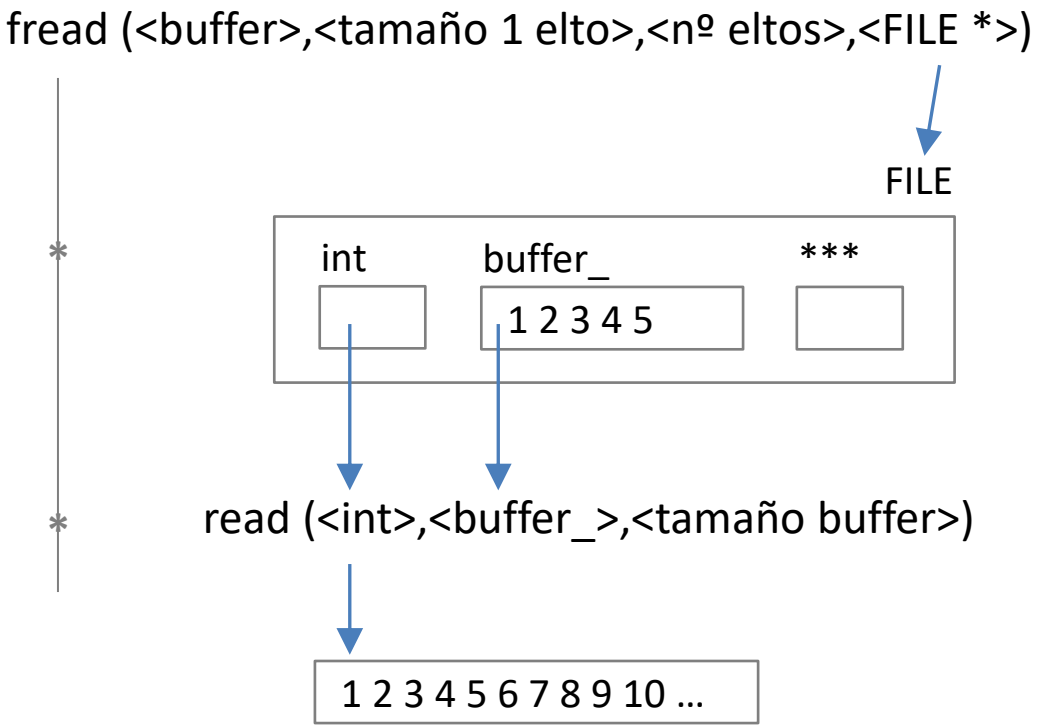
```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>


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) ;
}
```

sistema vs. libc

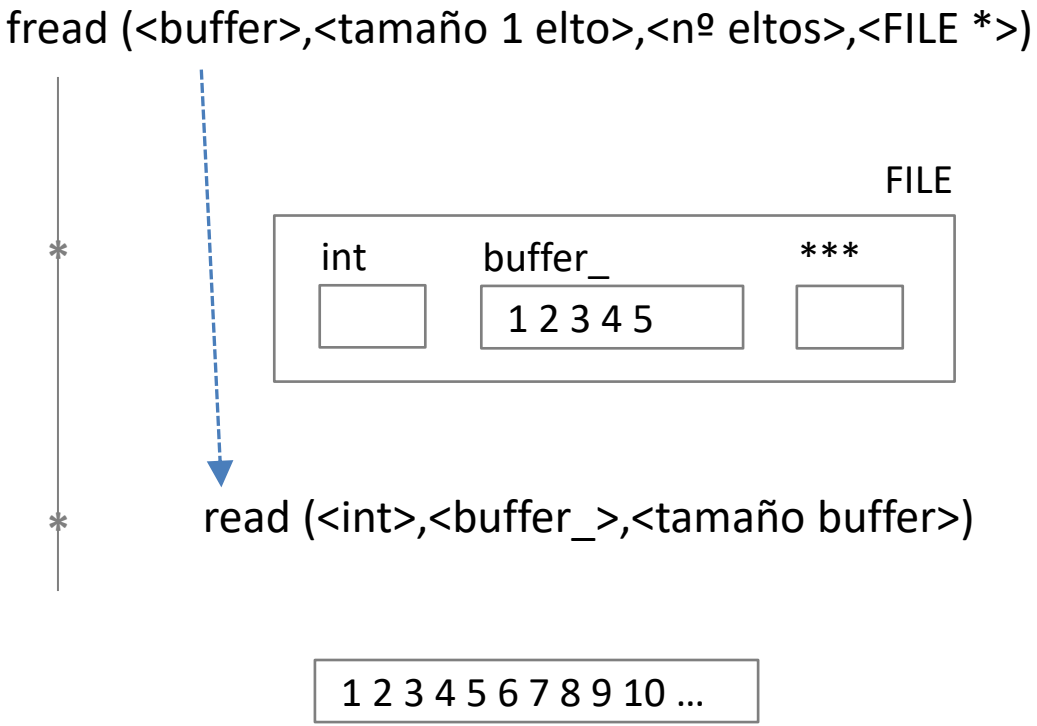
```c
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>

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);
}
```

```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);
}
```

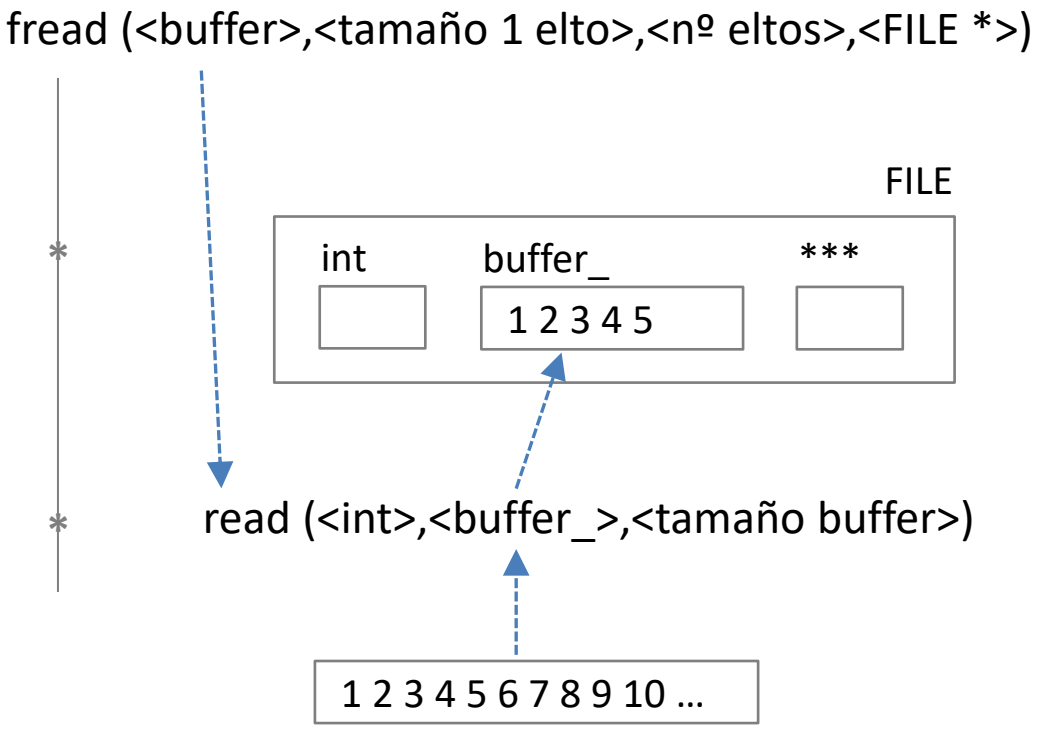
Funcionalidad extendida



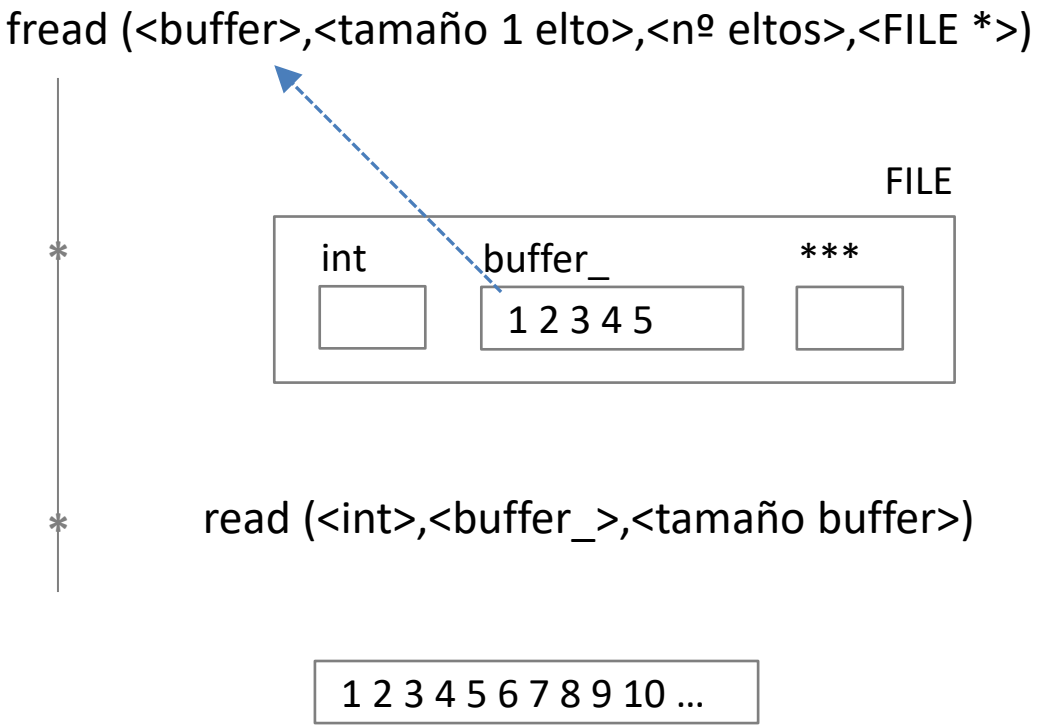
Funcionalidad extendida



Funcionalidad extendida



Funcionalidad extendida



Sistemas Operativos

sesión 6: llamadas al sistema

Grado en Ingeniería Informática

Universidad Carlos III de Madrid