

3.3 Modos de operación sobre dispositivos y ficheros (P05. Acceso Directo a Ficheros en Linux)

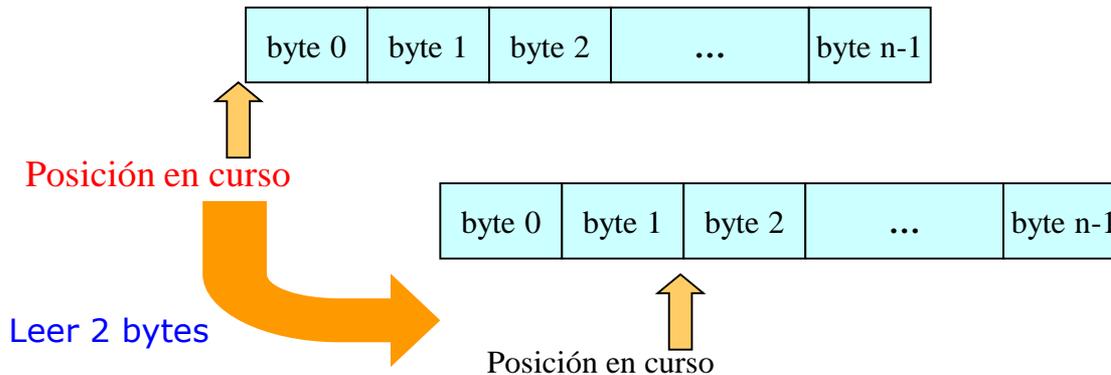
- ❑ *Vamos a estudiar los diferentes modos de acceso a los datos de los ficheros de datos en un SO. Y analizaremos cómo lo realiza Linux.*
- ❑ *Revisaremos las llamadas al sistema de UNIX y las funciones de biblioteca de C que nos permitirán desarrollar aplicaciones para acceder de forma directa o secuencial a los datos de un archivo*
- ❑ *Desarrollemos una utilidad similar al comando tee de Linux que usaremos como base para ampliar las funcionalidades del proyecto de E/S ()*

Modo de acceso a la información de ficheros

(UNIX, Linux)

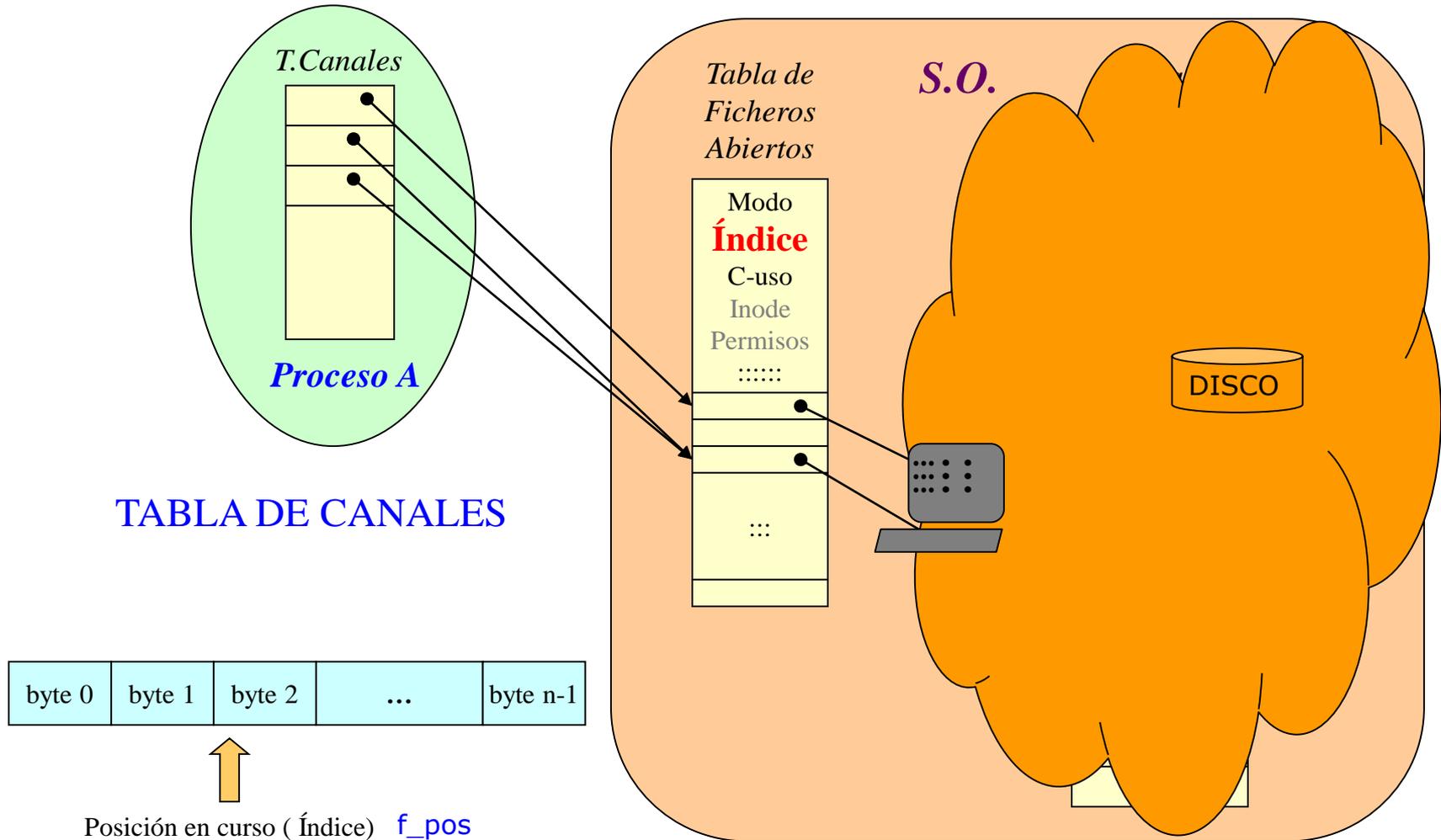
- **Acceso SECUENCIAL**
 - Fichero visto como una secuencia (vector) de bytes.
 - Se acceden a los k bytes situados a partir de la posición en curso. La nueva posición avanza k puestos.
- **Acceso DIRECTO (Aleatorio)**
 - Se permite acceder a cualquier elemento del fichero.
 - En UNIX se implementa con una función que permite posicionarse en cualquier byte del fichero, seguido de un acceso secuencial.

Ejemplos:



- **Otros modos de acceso** (Secuencial Indexado, Indexado, Acceso directo *HASH, Pila,...*)

Independencia del Dispositivo en UNIX



(TFA)Open file Table: `struct file` in `include/linux/fs.h`

Llamadas al Sistema para acceso directo a ficheros (UNIX/Linux)

□ Acceso directo a Ficheros

```
#include <sys/types.h>
#include <unistd.h>

off_t lseek (int fd, off_t offset, int whence);
whence :
    SEEK_SET      // origen:      0
    SEEK_CUR      // pos_actual:  1
    SEEK_END      // final:       2

// off_t es un tipo equivalente a long
```

Funciones de Biblioteca de C

□ Acceso directo a Ficheros

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

```
int fseek (FILE *stream, long offset, int whence);
```

whence :

SEEK_SET	start of the file
SEEK_CUR	the current position indicator
SEEK_END	end-of-file

```
long ftell (FILE *stream);
```

```
long rewind (FILE *stream);
```

Ejemplo: scola.c

```
// cola.c
#include <sys/types.h>
#include <unistd.h>
#include <errno.h>
#include <fcntl.h>

#define error(a) {perror(a); exit(1);};
#define BUFSIZE 80

main (int argc, char *argv[]) /* cola.c */
{
    int fd;
    int n;
    char buf[BUFSIZE];

    if ((fd = open(argv[1], O_RDONLY)) == -1)
        error("open");

    if (lseek(fd, -10L, SEEK_END) == -1)
        error("lseek");

    n = read(fd, buf, 10);

    if (write(1, buf, n) == -1) error("write");
    close(fd);
}
```

Act 05.3: Segunda versión del proyecto de E/S

- ▣ *Vamos a construir la segunda versión del proyecto de E/S a partir de las utilidades sininsertar ya desarrolladas.*

Para su desarrollo, sigue las instrucciones indicadas en la documentación de la actividad Act05.3