

Llamadas al Sistema Unix

Sistemas Operativos

Escuela de Ingeniería Civil Informática

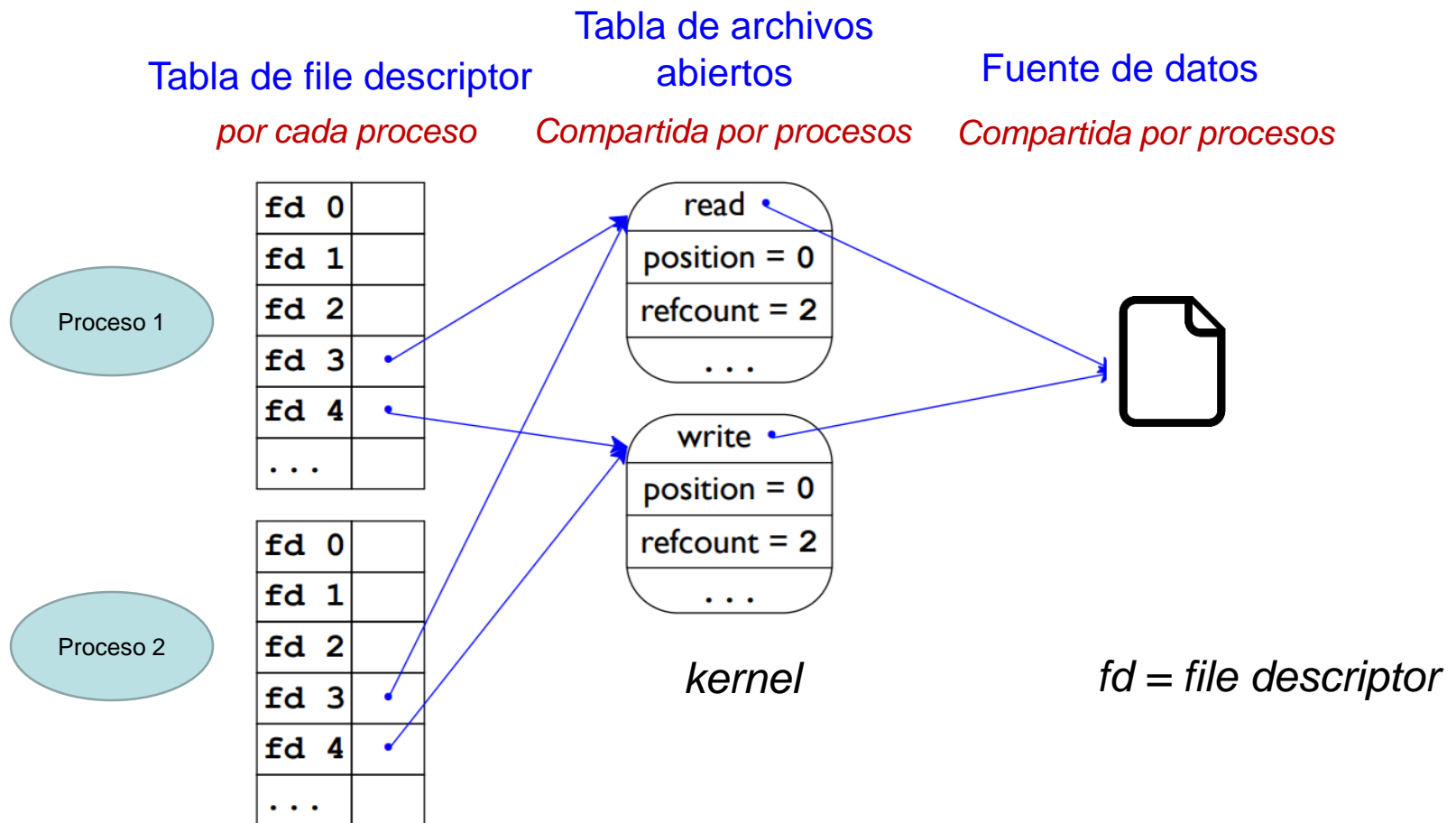
Entrada/Salida
`read()` – `write()`



UNIVERSIDAD DEL BÍO-BÍO

¿Qué es un descriptor de archivo?

- Es un número que identifica a una fuente de datos sobre la cual se puede leer/escribir:



Abrir un Archivo

- Para abrir un archivo se utiliza la llamada al sistema `open`.

Archivo cabecera	<pre>#include <fcntl.h> #include <sys/types.h> #include <sys/stat.h></pre>		
Formato	<pre>int open(const char *path, int oflag [, mode_t mode]);</pre>		
Salida	Exito	Fallo	Valor en errno
	Descriptor > 0	-1	Si

- El valor retornado se llama descriptor y corresponde al número identificador para ese archivo.
- Existe un arreglo de descriptores de archivos abiertos por cada proceso.
- path* corresponde a la ruta hacia el archivo.

Abrir un Archivo

- *oflag* puede tomar los siguientes valores:

O_RDONLY	abrir archivo para lectura
O_WRONLY	abrir archivo para escritura
O_RDWR	abrir archivo para lectura/escritura
O_NONBLOCK	no bloquear el proceso cuando se use el archivo
O_APPEND	agregar al final del archivo
O_CREAT	crear el archivo sino existe
O_TRUNC	crear archivo, si existe vaciarlo

Abrir un Archivo ...

- *mode* puede tomar los siguientes valores:

S_IRWXU	700	/* RWX dueño */
S_IRUSR	400	/* R dueño */
S_IWUSR	200	/* W dueño */
S_IRGRP	100	/* X dueño */
S_IWGRP	070	/* RWX grupo */
S_IXGRP	040	/* R grupo */
S_IXUSR	020	/* W grupo */
S_IRWXG	010	/* X grupo */
S_IRWXO	007	/* RWX otros */
S_IROTH	004	/* R otros */
S_IWOTH	002	/* W otros */
S_IXOTH	001	/* X otros */

Cerrar un Archivo

- Para cerrar un archivo se utiliza la llamada al sistema close.

Archivo cabecera	<code>#include <fcntl.h></code>		
Formato	<code>int close(int fd);</code>		
Salida	Exito	Fallo	Valor en errno
	0	-1	Si

Leer un Archivo

- Para leer un archivo se utiliza la llamada al sistema read.

Archivo cabecera	<code>#include <fcntl.h></code>		
Formato	<code>size_t read (int fd, void* buf, size_t cnt);</code>		
Salida	Exito	Fallo	Valor en errno
	- Número de bytes leídos - 0 si es fin de archivo	-1	Si

Ejemplo

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

int main()
{
    char c;
    int fd1 = Open("foo.txt", O_RDONLY, 0);
    int fd2 = Open("foo.txt", O_RDONLY, 0);
    read(fd1, &c, 1);
    read(fd2, &c, 1);
    printf("c = % c\n", c);
    exit(0);
}
```

foo.txt

foobar

Escribir un Archivo

- Para escribir un archivo se utiliza la llamada al sistema write.

Archivo cabecera	<code>#include <fcntl.h></code>		
Formato	<code>size_t write (int fd, void* buf, size_t cnt);</code>		
Salida	Exito	Fallo	Valor en errno
	- Número de bytes escritos - 0 si es fin de archivo	-1	Si

Ejemplo

```
#include<stdio.h>
#include <fcntl.h>
main()
{
    int sz;

    int fd = open("foo.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644);
    if (fd < 0)
    {
        perror("r1");
        exit(1);
    }

    sz = write(fd, "hola\n", strlen("hola\n"));

    printf("called write(%d, \"hola\\n\", %d).\"
        \" It returned %d\n", fd, strlen("hola\n"), sz);

    close(fd);
}
```