/\*

read con tres parámetros: uno para especifi-

car el archivo, uno para decir en dónde se van a colocar los datos y uno para indicar cuantos bytes

se deben leer”).

cuenta=read(fd, bufer, nbytes);

Los parámetros primero y tercero se pasan por valor, pero

el segundo parámetro se pasa por referencia, lo cual significa que se pasa la dirección del búfer (lo

cual se indica mediante &), no el contenido del mismo.

fd = open(archivo, como, ...) Abre un archivo para lectura, escritura o ambas

n = read(fd, bufer, nbytes) Lee datos de un archivo y los coloca en un búfer

n = write(fd, bufer, nbytes) Escribe datos de un búfer a un archivo

pagina 53 a 57

La llamada a fork devuelve un valor, que es cero en el hijo e igual al

identificador del proceso ( PID ) hijo en el padre. Mediante el uso del PID devuelto, los dos procesos

pueden ver cuál es el proceso padre y cuál es el proceso hijo.

cp archivo1 archivo2

main(argc, argv, envp)

en donde argc es una cuenta del número de elementos en la línea de comandos, incluyendo el nom-

bre del programa.

El segundo parámetro, argv, es un apuntador a un arreglo. El elemento i de ese arreglo es un

apuntador a la i-ésima cadena en la línea de comandos. En nuestro ejemplo, argv[0] apuntaría a la

cadena “cp”, argv[1] apuntaría a la cadena “archivo1” y argv[2] apuntaría a la cadena “archivo2”.

Para leer o escribir en un archivo, éste debe primero abrirse mediante open. Esta llamada es-

pecifica el nombre del archivo que se va a abrir, ya sea como un nombre de ruta absoluto o relati-

vo al directorio de trabajo, y un código de O\_RDONLY, O\_WRONLY o O\_RDWR, que significa

abrir para lectura, escritura o ambos. Para crear un nuevo archivo se utiliza el parámetro O\_CREAT.

Después se puede utilizar el descriptor de archivo devuelto para leer o escribir. Al terminar, el ar-

chivo se puede cerrar mediante close , que hace que el descriptor de archivo esté disponible para

reutilizarlo en una llamada a open posterior.

Las llamadas de uso más frecuente son sin duda read y write . Anteriormente vimos a read .

Write tiene los mismos parámetros.

\*/

/\*

Opendir . Los directorios se pueden leer. Por ejemplo, para listar todos los archivos en un

directorio, un programa de listado abre el directorio para leer los nombres de todos los ar-

chivos que contiene. Antes de poder leer un directorio se debe abrir, en forma análoga al

proceso de abrir y leer un archivo.

Closedir .

Cuando se ha leído un directorio, se debe cerrar para liberar espacio en la tabla

interna.

Readdir . Esta llamada devuelve la siguiente entrada en un directorio abierto. Antes era po-

sible leer directorios utilizando la llamada al sistema read común, pero ese método tiene la

desventaja de forzar al programador a conocer y tratar con la estructura interna de los direc-

torios. En contraste, readdir siempre devuelve una entrada en formato estándar, sin impor-

tar cuál de las posibles estructuras de directorio se utilice.

La función readdir () devuelve un puntero a una estructura dirent que representa la

siguiente entrada de directorio en el flujo de directorio apuntado por dirp.

Devuelve NULL al llegar al final de la secuencia del directorio o si ocurrió un error.

\*/