File System – System Calls (1A)

Young Won Lim
11/28/16

Read System Call

ssize_t **read**(int **fd**, void ***buf**, size_t **count**);

fd: the file descriptor of the file,
buf: the buffer where the read data is to be stored and
count:the number of bytes to be read from the file.

The file is identified by a file descriptor that is normally obtained from a previous call to open. This system call reads in data in bytes, the number of which is specified by the caller, from the file and stores then into a buffer supplied by the calling process.

https://en.wikipedia.org/wiki/Read_(system_call)

Write System Call

ssize_t **write**(int **fd**, const void ***buf**, size_t **nbytes**);

fd: the file code (file descriptor or fd).
buf: the pointer to a buffer where the data is stored (buf).
nbytes: the number of bytes to write from the buffer (nbytes).

The write system call is one of the most basic routines provided by the kernel. It writes data from a buffer declared by the user to a given device, maybe a file. This is primary way to output data from a program by directly using a system call. The destination is identified by a numeric code. The data to be written, for instance a piece of text, is defined by a pointer and a size, given in number of bytes.

https://en.wikipedia.org/wiki/Write_(system_call)

Open System Call

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

int **creat**(const char ***path**, mode_t **mode**);

Path: The name of the file to open. It includes the file path defining where, in which file system, the file is found (or should be created).

Oflag:

This argument formed by OR'ing together optional parameters and (from <fcntl.h>) one of: O_RDONLY, O_RDWR and O_WRONLY

Option parameters include:

O_APPEND, O_CREAT, O_EXCL, O_TRUNC

Mode:

Optional and relevant only when creating a new file, defines the file permissions. These include read, write or execute the file by the owner, group or all users. The mode is masked by the calling process's umask: bits set in the umask are cleared in the mode.

https://en.wikipedia.org/wiki/Open_(system_call)

Close System Call

int **close** (int **filedes**);

For most file systems, a program terminates access to a file in a filesystem using the close system call. This flushes buffers, updates file metadata (which may include and end of file indicator in the data), de-allocates resources associated with the file (including the file descriptor) and updates the system wide table of files in use.

https://en.wikipedia.org/wiki/Open_(system_call)

File Descriptor

a file descriptor (fd, fildes) is an abstract indicator (handle)
used to access a file or other input/output resource,
such as a pipe or network socket.

the POSIX application programming interface
a non-negative integer (int)
(negative for "no value" or an error condition).

three standard POSIX file descriptors,
corresponding to the three standard streams:

Int val	Name	symbolic constant	file stream
0	Standard input	STDIN_FILENO	stdin
1	Standard output	STDOUT_FILENO	stdout
2	Standard error	STDERR_FILENO	stderr
		<unistd.h>	<stdio.h>

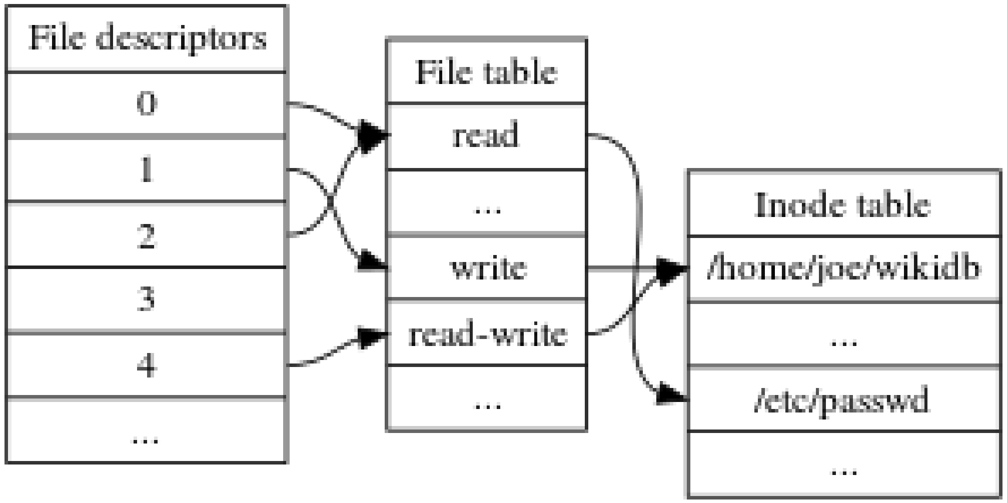
https://en.wikipedia.org/wiki/Open_(system_call)

File Descriptor (1)

file descriptors index into a per-process **file descriptor table**

A file descriptor table indexes into **the file table**

The file table indexes into **the inode table**



https://en.wikipedia.org/wiki/File_descriptor

File Descriptor (2)

A file descriptor table

- maintained by the kernel

The file table

- a system-wide table of files opened by all processes
- records the mode (r, w, a, rw, and etc)
- indexes into a third table called the inode table that describes the actual

The inode table

- describes the actual underlying files

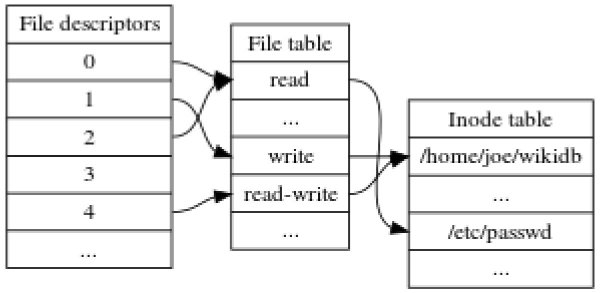
https://en.wikipedia.org/wiki/File_descriptor

File Descriptor (3)

To perform input or output,

the process passes the file descriptor
to the kernel through a system call,
and the kernel will access the file on behalf of the process.

The process does not have direct access
to the file or inode tables.



https://en.wikipedia.org/wiki/File_descriptor

References

[1] http://minix1.woodhull.com/current/2.0.4/
[2]