

OPEN SERVER – VERSION ONE



OPEN SERVER – VERSION ONE

- File descriptor is an abstract indicator used to access a file or other input/output resource, such as a pipe or network socket.
- Passing file descriptors allows one process to do everything required to open a file and simply pass back a descriptor to the calling process that can be used with all I/O functions.
- Using file descriptor passing, an open server is developed: a program that is executed by a process to open one or more files.

OPEN SERVER – VERSION ONE

- Instead of sending the contents of the file back to the calling process, the server sends back an open file descriptor.
- This lets the server work with any type of file (such as a device or a socket) and not simply regular files.
- It also means that a minimum of information is exchanged using IPC: the filename and open mode from the client to the server, and the returned descriptor from the server to the client.
- The contents of the file are not exchanged using IPC.

OPEN SERVER – VERSION ONE

- There are several advantages in designing the server to be a separate executable program.
- The server can easily be contacted by any client, similar to the client calling a library function. We are not hard coding a particular service into the application, but designing a general facility that others can reuse.
- If we need to change the server, only a single program is affected. Conversely, updating a library function can require that all programs that call the function be updated (i.e., relinked with the link editor). Shared libraries can simplify this updating.
- The server can be a set-user-ID program, providing it with additional permissions that the client does not have. A library function (or shared library function) cannot provide this capability.

OPEN SERVER – VERSION ONE

- The client process creates an s-pipe (either a STREAMS-based pipe or a UNIX domain socket pair) and then calls fork and exec to invoke the server.
- The client sends requests across the s-pipe, and the server sends back responses across the s-pipe.
- The following application protocol is defined between the client and the server.
- The client sends a request of the form “open <path_name> <open_mode> \0” across the s-pipe to the server. The <open_mode> is the numeric value, in ASCII decimal, of the second argument to the open function. This request string is terminated by a null byte.
- The server sends back an open descriptor or an error by calling either send_fd or send_err.

OPEN SERVER – VERSION ONE

- The child closes one end of the pipe, and the parent closes the other.
- For the server that it executes, the child also duplicates its end of the pipe onto its standard input and standard output. (Another option would have been to pass the ASCII representation of the descriptor `fd[1]` as an argument to the server.)
- The parent sends to the server the request containing the pathname and open mode.
- Finally, the parent calls `recv_fd` to return either the descriptor or an error.
- If an error is returned by the server, `write` is called to output the message to standard error.

THANK YOU

