Javier Bellón u214002
Raúl Díaz u214409

Report Lab 5

1. Explain why two pipes are enough even if there are several children

All children can write to the same pipe because that pipe will only be read at the end of the execution when the script is synchronous again so there is no problem with the order in which we write the blocks we get. All children can also read from the same pipe because the father is the only process writing in it and the children that reads first will be the one to process that request, when the first child is busy another child will read from the pipe. We only need one pipe to be able to form a “queue” to orderly process the requests as they arrive and as children become available to process the because only one child process will read a request when multiple are waiting.

2. Explain how the ending is handled (when the standard input of the father closes).

When the user is done writing and sends the signal to stop, the parent will close the writing part of the parent-to-children pipe and the children awaiting text from the pipe will stop waiting (the while loop finishes) and finish execution. Then the parent will be the only process left and will read the content from the children-to-parent pipe and finish the rest of the program.

3. An alternative of using file locks would be to use a named semaphore to make any access to the file exclusive. Which disadvantages has this solution?

A named semaphore can cause performance problems because there is no way to block just one part of the file so it would block the whole file. File locks however can lock and unlock certain parts of the file and have other processes access the same file at the same time assuming they don’t try to access the same part.