Tània Pazos, u185115
Júlia López, u214964

Report Lab 5

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

After the fork(), each child has access to both ends of each pipe. Hence, two unidirectional pipes are sufficient to enable communication between the parent and its children: data flows from the parent to the children through pipe A, whereas it flows from the children to the parent through pipe B.

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

When the standard input of the father closes -the user presses Ctrl + D- the father closes the writing end of pipeA, which serves as a signal to the children that no more commands will be sent.

```
while((nBytesRead = read(0, s, 100) ) > 0) {
        (...)
    }
    close(pipeA[1]);
    printf("FINISHED\n");
```

Then, while reading pipe A, the children identify the null terminator '\0' and exit the loop and proceed to close the reading end of pipe B and the writing end of pipe A.

```
    char s[100];
    int nBytesRead;
    /* Read from the standard input*/
    while((nBytesRead = read(0, s, 100) ) > 0) {
        char op[200];
        s[nBytesRead] = '\0';
        int nBlock;
        (...)
}
```

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?

Firstly, using named semaphores to ensure that any access to the file is exclusive would require more operations than using file locks such as initialization, waiting and signaling.

What is more, named semaphores persist in the file system even after the processes that have created them terminate execution. This could lead to unnecessary resource consumption.

Finally, named semaphores could potentially lead to performance overhead due to additional system calls.