whatIf will fork a child process, and then both parent and child process will try to read the file. There are two problems here.

First, the file descriptor is shared by both parent and child. And the file offset the position where the file will be read or write, is also shared by both processes. That means if the parent read 5 chars at first, then the child process can only read the data after the first 5 chars, because the read function will modify the file offset.

Second, the read function will call a system call, which will block other read call until the first read is finished. If the parent process try to read the file, while the system is read the file required from the disk, any other reading call will be blocked, until the parent call is finished.

In the whatIf program, both parent and child use while loop to read the data line by line until reaching the end of the file. We can see that, one process, either parent or child, first start reading lines. After several lines, the other process will start to read the remaining lines. After several lines, randomly, the first process will start again.
We cannot predict how many lines a process will read, this is depended by OS kernel and depended by which process acquire the file lock first. In rare cases, one process read all the lines, and the other read none. The only thing we can make sure is that the data in parent.txt plus the data in child.txt is all the data in the original data.txt.