1.a What happens if the return value of execlp() is not checked and you give an invalid command, say, lssss, as input to the shell?
The child process will not exit automatically. It seems that, the original parent process is replaced by the new child process. The parent process id will be replaced by the child process id, which makes parent process id replaced by child process id.

1.b What happens if the parent process does not perform waitpid() and immediately returns to the beginning of the while-loop?

The child process will hang there and become zombie process. Because the parent process did not wait for the child process to change status. The child process will hang there untill the parent process kill it.

1.c If the concurrent server were a file server that receives client requests from processes on the same host/OS (or over a network), why is performing waitpid() as a blocking call from within the parent process not a valid approach? Describe an asynchronous method for performing waitpid() so child processes are prevented from becoming zombies and their exit status can be checked.

Waitpid() as a blocking call is not good. Because for file server, the package usually need to be sent continuously and blocking the whole system will cause losing some of the package which is delivered during the blocking process.

Solution:
We could make waitpid() a non-blocking call by state WNOHANG in the parameters, which will make it not to block before the child process status change, thus it will be able to receive package all the time.
To realize asynchronous check of process status, we could use SIGCHLD to check the the status check and inform the parent process to kill the zombie process.

1.d Ignoring the functional simplifications of the shell code, point out at least two programming bugs that should be fixed to yield more reliable server code.

a. There are chances that fork() does not successfully create a child process, which will return -1. We should check this status.
b. There is possibility of two or more client requests getting interleaved due to unpredictable process scheduling, we should specify which process id is associated with which cmd.

BONUS:
The bonus problem solution are in fifoserver2.c and fifoclient2.c