Graded task 2 on signals, processes and files

The program accepts a reasonable amount of natural numbers Ni (100>Ni>0) and one file name F as parameters. For each number a dedicated child process is started. Each child chooses a random number C in range [1,Ni]. In loop, child process sends SIGUSR1 to its parent and sleeps C*100ms. The loop iterates exactly C times, after it, child process terminates.

Parent process opens the file, create it if necessary but do not erase its content if it already exists. At each delivery of SIGUSR1 parent writes 100 random bytes from the range [a-z] to the file F, starting from the beginning of the file. It terminates as soon as all the children are gone.

<u>Parent logic can not be implemented in the signal handler!</u>

Graded stages:

1. **(4p)** Children print their Ni number and C numbers, parent process waits until all children terminate
2. **(5p)** Children send the signals as described in the task, parent only reports each signal delivery by printing one “*” on the stdout.
3. **(5p)** Parent fills the file F with the chars as described in the task, at this stage signals can be lost and writing can be interrupted.
4. **(3p)** The only lost signals result from signals merging, file writing can not be interrupted.