

You will create a set of cooperating processes as follows:

The initial (parent) process receives as arguments a text file (X), the number of child processes (K), and the number (N) of requests in which each of the child processes is involved. The number of requests will be the same for all child processes.

The parent process initiates the creation of K child processes, informing them of the number of lines in the file X and the number of requests (recycles) N that each child will perform. The number of lines in the file X is determined by the parent process.

After their creation, the child processes formulate requests to the parent process, specifying a specific line of the given file (random selection). They behave as clients to a server, which returns information according to the submitted requests. Response messages include the requested line. Request example: Child i to parent process: 3 (bring line 3 from the file), parent process to child i: "The rain in Spain stays mainly in the plain!" (this is the 3rd line of the file). Each line of the file can have a maximum length of 100 characters (shared memory scales accordingly).

Design the appropriate structure of shared memory and the required set of semaphores that will make the above architecture functional.

The execution of processes stops when N requests are completed. The parent process receives return codes from the completing children. The child processes record the average time elapsed from submitting a request to receiving the corresponding response. Upon the completion of each child's execution, the average time is reported in the standard output along with information identifying the process."