The program creates three different shared memory segments

- The first segment is for user input. The program asks the user for a number to search in the given array.

- The second shared memory segment is for an array. After user enters the file name the program looks for the file and puts every integer into shared array which is shared among all processes.

- Last shared memory segment is for counter. Each child process concurrently counts the number of occurrences of the number in the array.

Pipes are created to maintain synchronization between main and child processes.

- Since read () function in child's waits for its parents, child processes wait until the main process instruct.

- Same as read () function, with the write () function in child, main process also waits to get result after calculation.

- Main processes initialize an integer as 0 and passes it to the child processes through pipe. The integer is used for determining the index of each portion of the array. Each child passes their last index to the next child. So that, every child processes scan their portion in the array.