The project consists of a parent program and a child program. Each program has its own unique function. The parent program uses a command line argument to generate 4 child processes. The parent program receives two integers (a and b) and a sleep time value (t) from the command line arguments. If the sleep time value is greater than 50 or less than 0, the parent program exits. If the correct number of arguments and correct values are passed, the parent program displays the two integers and then sleeps for a random time modulo t. The parent process creates 4 child processes with the fork function. In the child process, the two integers, the sleep time value and the child number are passed to the child program using the execl function so the child program can execute. The child process will exit if the execl function fails. The parent process displays the child PID number of the forked child. Once 4 child processes have been created the parent process exits.

The child program takes the arguments from the parent program and uses them to perform arithmetic operations on the two values (a and b) and to sleep modulo t. The child program also receives an integer value (0-3) corresponding to a child number. The specific arithmetic operation depends on the child number. If the child number is 0, the child process will add the integers a and b. If the child number is 1, the child process will subtract the integer b from the integer a. If the child number is 2, the child process will multiply the integers a and b. If the child number is 3, the child process will divide b into a, assuming b is not 0. If b is 0, the child process prints a message noting that it cannot divide by zero. After the child process performs its operation the process exits.