

Implement a program that accomplishes the following:

Call your program *Project1*. The program *Project1* requires two parameters called *Param1* and *Param2*.

The first parameter, *Param1*, is a string that represents a UNIX command such as "*ls -l*" or "*cat file1*". The second parameter is a file name that does not exist at program start.

Execute *Project1* in the background as follows:

1. Have the parent create a child process.
2. Using *unnamed pipes*, have the parent send the *Param1* command to the child. In this exchange, use *read* and *write* system calls to effect the exchange of the command from the parent to the child.
3. Have the child use *mkargv* to place the command into a buffer, and then *execvp* to execute the command.
4. Have the child place the result of executing the command into a buffer called *buff*.
5. Then use named pipes with minimum necessary permissions based on the file *my.dat* to have the child send the command output in *buff* back to the parent.
6. Then have the parent use system calls and file descriptors (i.e., *open*, *write*) with minimum necessary permissions to send the *buff* contents to the file name input by *Param2*.

Keep both the parent and the child executing sending a message they are alive exactly every five seconds (i.e. parent alive and child alive).. When the user raises *SIGUSR1* at the command prompt, the parent terminates normally. When the user raises *SIGUSR2* at the command prompt the child terminates normally. When both are terminated, the program *Project1* terminates.