CECS 326: Programming project[1]

Objectives: Creation and inheritance of child processes with *fork* and *wait*

Implement a string replacement software and count the number of times some given string is replaced in a plain text document.

Software components:

- The *UI* (not GUI) program is the interface with the user. Prompt your user to enter a ***target*** string and the ***replacement*** string. All ***target*** strings in some document[2] are to be replaced with the ***replacement*** string. Here, spawn a child process to perform the replacement task. Allow the child process to finish and output the number of string replaced (if any), before prompting the user for the next pair of strings. If the user enter the string "!wq", terminate the program.

  To simplify the programming, a lengthy (300 words minimum) plain text document (e.g. excerpt of a web article) can be hard coded in the program as a C++ string object (or equivalent).
- The child process is spawned by the *UI* process to replace and count the number of times a string is replaced in the document. Output the count and terminate (if count isn't zero.) Note: Each child can work with the original/initial copy of the document.

Injected bug: When a replacement should fail, i.e. count is zero, program your child process to redo and repeat the failed replacement over and over (indefinitely). This bug would require user's intervention at a separate terminal to terminate the child process gracefully via command console (w/o terminating the parent process). To denote its eternal execution, the child process prints a dot (.) prior to each repeated attempt.

Submit a hard copy of the program listing with a cover page (name, date, course #, email address, program description & optional critique). Also, document and submit your test cases which you used to test your program. This can be a small table that lists each test case, the expected result, and the purpose of each test case. Show your instructor the program execution of your test cases & discuss the details of your implementation when necessary.

[1] This project can be completed after the discussion of the sample programs on *fork*.
[2] Each child works on the same original document.