CS 452 Operating Systems Project One
“One Bad Apple”

In this project you will design a circular communication system where k processes will form a ring where each node is only connected to its direct neighbor. For example, node 1 will have a one way write connection to node 2 which has a one way write connection to node 3 etc. Node k will have a write connection to node 0. Node 0 will also have a read connection from node k which will have a read connection from node k-1, etc.

A node can only send/receive a message if it possesses the apple. The apple is passed around the ring in a circular fashion. The apple is used to synchronize the system.

When each node receives the apple, it should determine if the message was intended for them. If it was, it copies the message and sets the header back to ‘empty’. If the message was not intended for them, they should send it on to the next node after a small delay. Verbose diagnostic messages should be displayed so a viewer can follow what the system is doing (e.g., who is sending/receiving data, where the data is heading to, when a process is created/closed, etc.).

When the user presses Control-C, the original process should use signals to gracefully shutdown the simulation. You must use the process management and IPC system calls that were covered in class for the implementation (e.g., fork, exec, signal, pipe, etc.).

The parent process should request from the user the value for k and spawn that many processes. After the processes have completed spawning, the parent process should prompt the user for a message and what node they would like to send it to.

Code should use descriptive variable names using the camelCase convention. Code should be self-documenting where possible with minimal documentation within the code.

Hand-in:

1. A design document clearly describing your project as well as the implementation
2. The source code in c (no zip files please)
3. A screenshot of the execution which clearly shows where the apple is and what nodes are sending/receiving etc. At least two messages must be sent to different nodes from the parent (send one message, after it is received and the apple returns to the parent, send a second message). The screenshots should be from a terminal where the output is easy to read (e.g., not a screenshot of visual studio code executing the code).

Grading:

Points will be deducted based on features that are missing in the project. The more prominent the feature, the larger number of points.

Extra Credit:

1. Turn the project in 2 weeks early (+10%)
2. Assign one of the nodes to be a 'bad apple'. The bad apple should replace some of the data it receives with new random data before sending it on (+5%).

C grade option:

1. All of the above except instead of k, implement a fixed circular ring with 3 nodes.
2. Please note points can still be deducted which will reduce the grade from the max grade of a C.

Appendix A: Visual showing nodes and the direction of communication.