Project One Design Document

“One Bad Apple”

Gabrielle Munson and Gabe Baksa

Program Logic

In the beginning of the main function, we have a signal handler for node creation. When developing our code, we noticed that “Enter a message:” was printing out before all the nodes had been created. This handler works with our while statement on line 87 to ensure all nodes are created before the process continues.

The user is then asked to enter the number of nodes (k). A 2D array of k pipes is then created to allow IPC between the nodes. The parent node then forks off k child process, checks for proper creation, and then outputs the number of each as they are created.

Once the child nodes are created, the parent process continues. On the parent node we place a signal handler for the SIGINT signal; if the user enters Crtl+C at any point in the program, it will gracefully exit. We then create read and write pipes for the parent and clear the buffer with “getchar()”.

Now in a while loop, the parent then prompts the user for a message to send, stores the message in messageToSend.message and removes the extra new line character. The user is then prompted to enter a node to send the message to, stores that node in messageToSend.targetNode and removes the newline character from the buffer. Read and write pipes are then set up for the parent and the messageToSend is sent to the first node.

Once a node receives the apple (messageToSend) it first checks to see if it is the target node. If so, it sets the message header to empty and passes on the message. If not, it just passes on the message to the following node. If the node that received the message is the last child node, it sends the message back to the parent node.

Once the parent receives the apple again, it will print out that it has the apple and whether or not the message reached the target. It then asks the user if they want to send another message, answering y or n. If y then the while loop mentioned in paragraph four continues. If the user enters n, then the program gracefully exits.