Sebastian Torrejon Alonzo
Fabian Kirberg

One Bad Apple

Project Overview

The One Bad Apple is a C program that simulates a circular communication system using multiple processes, where each process acts as a node in the system. The primary goal of this project is to demonstrate interprocess communication (IPC) and the propagation of messages across multiple nodes in a distributed system. The program allows users to set the number of nodes (processes), designate a "bad apple" node that modifies messages, and send a message to a node.

Implementation

Data Structures

The program utilizes the following data structures:

1. `Apple` Struct: Represents the (apple) with two components:
    - `header`: An integer representing the target node for the message.
    - `message`: A character array (string) to store the message content.

Functions

`sigHandler(int sigNum)`

- Description: A signal handler function to gracefully shut down all the processes when the user presses Ctrl+C.
- Parameters:
    - `sigNum`: An integer representing the signal number (Ctrl+C).
- Return Value: None

`main()`

- Description: The main function that orchestrates the distributed messaging system.
- Flow of Execution:
    1. Set up a signal handler for Ctrl+C (SIGINT).
    2. Prompt the user to input the number of nodes (`k`) in the circular system. This value must be a natural number greater than 2.
    3. Prompt the user to choose a "bad apple" node or opt not to have a bad apple. The bad apple node modifies messages.
    4. Create a set of pipes for communication between nodes.
    5. Spawn child processes (nodes) based on the value of `k`. Each node runs a loop for message communication.

6. In the parent process (Node 0), prompt the user to input a message and the target node for the message.
7. Send the apple into Node 0's pipe to initiate the circular process.
8. Continuously receive and send the apple among nodes in a circular manner.
9. If a message reaches its target node, it is displayed, and the target node sends the apple back to node 0.
10. If a node is designated as the "bad apple," it modifies the received message before forwarding it.
11. Messages are propagated with a delay to simulate a real-time system.

- Return Value: None

Interprocess Communication

- The program uses pipes to enable interprocess communication between nodes.
- Each node communicates with its neighboring nodes in a circular manner.

User Interaction

- The program interacts with the user through the command line.
- Users provide input to set the number of nodes, designate a bad apple node, and send messages to a node.

Error Handling

- The program handles various error conditions, such as invalid user inputs and pipe creation failures, by displaying error messages and exiting gracefully.

Build and Execution

To compile and run the program, follow these steps:

1. Save the code in a C file (e.g., ‘oneBadApple.c’).
2. Open a terminal and navigate to the directory containing the C file.
3. Compile the code using a C compiler (e.g., ‘gcc’):

```
gcc -o oneBadApple oneBadApple.c
```

4. Run the compiled program:

```
./oneBadApple
```