Max Khan
Dhruvi Kothari
Inderpreet Kaur
Andrew Li

CSEN 383 Assignment #6, Group 3
5/27/2025

This was an interesting project with a variety of possible ways to implement. As a group we discussed a few ways to tackle the problem, including different code examples of how we could tackle the problem.

Ultimately, this program implements a UNIX I/O system with five child processes communicating via pipes. Four children (2-5) generate timestamped messages every 0-2 seconds, while child 1 reads from the stdin. The parent uses select() to read from all the pipes, prepends its own timestamp, and writes to output.txt. All processes terminate after 30 (human) seconds.

We encountered a variety of issues:

[1] Our initial code from different contributors had multiple mains functions and caused linker errors. After settling on an approach, we consolidated our code into a single main file.

[2] We has incorrect child indexing. Children accessed the wrong pipes (fd[child]), and we fixed this to use fd[i] based on the fork loop index.

[3] Our parent timestamps were missing from our output.txt as they failed to be assigned as a prependage. We added the functions get_time_diff and format_time to fix this.

[4] Our original logic utilizing the parent_flag/noresp_count was flawed, and caused unreliable termination. We implemented the use of a active_children counter and pipe closure on EOF.

[5] Our clock logic for handling our standard input was overly complex, and lead to a lot of confusion. We greatly simplified it and added a timeout based select().

[6] We had timestamp inaccuracies due to utilizing a timeDiff delta. We fixed this by replacing it with a relative time calculation.

[7] In general, we worked to simplify our pipe handling – preventing the leakage of file descriptors.

Sample output:

```
00:00.001: 00:00.000: Child 2 Message 1
00:00.001: 00:00.000: Child 3 Message 1
00:00.001: 00:00.000: Child 4 Message 1
00:00.001: 00:00.000: Child 5 Message 1
00:01.001: 00:01.001: Child 2 Message 2
00:01.001: 00:01.001: Child 3 Message 2
00:01.919: 00:01.918: Child 1 Message 1: lkajsdfl;k jasdl;fkj
00:02.001: 00:02.001: Child 2 Message 3
00:02.001: 00:02.001: Child 4 Message 2
00:02.001: 00:02.001: Child 5 Message 2
00:02.294: 00:02.294: Child 1 Message 2: ;lk
00:02.536: 00:02.536: Child 1 Message 3: ajsdf;lk
00:02.772: 00:02.771: Child 1 Message 4: jasdlf;k
00:03.002: 00:03.002: Child 2 Message 6
00:03.002: 00:03.002: Child 3 Message 3
00:03.044: 00:03.044: Child 1 Message 5: ja;lskdj
00:03.326: 00:03.326: Child 1 Message 6: f;laksjd
```

```
00:03.583: 00:03.583: Child 1 Message 7: f;laksdj
00:03.781: 00:03.781: Child 1 Message 8: f;laskj
00:03.974: 00:03.974: Child 1 Message 9: df;la
```