Pipes and FIFOs

UNX511 Week 9 Class 1

John Sellens

July 8, 2025

Seneca Polytechnic

Outline

Pipes and FIFOs

Pipes in the Shell

- Recall that the shell provides pipes
- A mechanism for connecting the output of one process to the input of another
- Shell pipes are unidirectional
- Shell pipes connect processes on the same machine, with a common ancestor process
- The shell creates pipes using the pipe IPC method

Pipes and FIFOs

- Pipes and FIFOs are inter process communication mechanisms
- FIFOs – short for “first in, first out”
  - Also known as “named pipes” since they appear in the filesystem
- For communication between processes on a single machine
- Pipes and FIFOs are unidirectional
- Pipes can only be used by related processes (common ancestor)
- FIFOs can be used by unrelated processes, as long as they have permission to open the pipe in the filesystem
- Pipes and FIFOs predate IP protocols (e.g. UNIX domain sockets) on UNIX machines

Creating and Using Pipes

- A pipe is created by calling the `pipe(2)` system call
- Pass a two element integer array to pipe()
- pipe() returns two file descriptors in the array
  - The first for read, the second for write
- These file descriptors can be `dup()`'d or inherited by child processes
- One process `write()`s then `close()`s
- The other `read()`s until eof-of-file, then `close()`s
- The kernel provides intermediate buffer space "within"' the pipe
- See `pipe(7)` for an overview of pipes and FIFOs
- See also `popen(3)` for an easy way to run a command in a pipe

Creating and Using FIFOs

- Call `mkfifo(3)` (or `mkfifo(1)`) to create a named pipe in the file system
- Then processes can open the pipe for read or write – subject to normal filesystem permission
- The kernel provides intermediate buffer space “within”’ the FIFO
- Nothing actually gets written to disk – it’s just a device
- Very similar to the bidirectional UNIX domain socket

Readers and Writers

- You're allowed to have multiple readers and writers on pipes and FIFOs
- But I imagine it might not be predictable or easy to use
- Multiple writers should be careful to write a single block in an atomic `write()` operation
- I don't know if multiple readers each get a copy of all input, or if it's first come, first served

Pipe and FIFO Code Examples

- Let's have a look in unx511_samples
  - `https://github.com/jsellens/unx511_samples`
- `week9_1/1_pipes` – various pipe examples
- `week9_1/2_fifos` – FIFO examples

Pipe and FIFO References

- `mkfifo(1)` `mkfifo(2)` `pipe(2)` `popen(3)` `dup(2)`
- `https://github-pages.senecapolytechnic.ca/unx511/Week9/Week9.html`

Summary

- Pipes and FIFOs – more IPC mechanisms
- There are restrictions
- These pre-date IP networking in UNIX
- Pipes are necessary in the shell
- Otherwise, some might argue IP networking is often a better choice