Homework
Part 2: Concurrency –
9) IPC, Event-Based Concurrency

P. Mainini / E. Benoist / C. Fuhrer / L. Ith

Autumn 2025-26 `master@352a46f (20250901-155000)`

1 IPC and Locking

1.1 Consumer/Producer Using Processes

Implement a version of the producer/consumer problem from *08-concurrency-3 (p. 12 ff.)* using multi-programming and separate processes for producer(s) and consumer(s):

- Make the size of the buffer configurable (e.g. using `#define MAX` or as command-line option)
- Use shared memory for data exchange and synchronization
- Provide two versions: one using condition variables and one using semaphores
- You may use a third process to start producers and consumers (using `fork()` and `exec()`) or start them individually by hand/script

1.2 ★ Benchmarks

Try to compare the multi-programming consumer/producer problem with a version using threads. Can you measure any differences? If so, how (varying number of producers / consumers, different buffer sizes, …)? Do you notice any difference on the OS side?

2 Asynchronous I/O (AIO)

Read [arv] to learn about the real-world complexities when using asynchronous I/O.

3 ★ Event-Based Concurrency

To learn more about multi-programmed-, multi-threaded- and event-based architectures, read [PDZ99] (also provided as part of the course material).

References

[arv] arvid, *asynchronous disk I/O*, https://blog.libtorrent.org/2012/10/asynchronous-disk-io/.

[PDZ99] V. Pai, P. Druschel, and W. Zwaenepoel, *Flash: An Efficient and Portable Web Server*.