

CSE 451: Operating Systems

Hard Lessons Learned

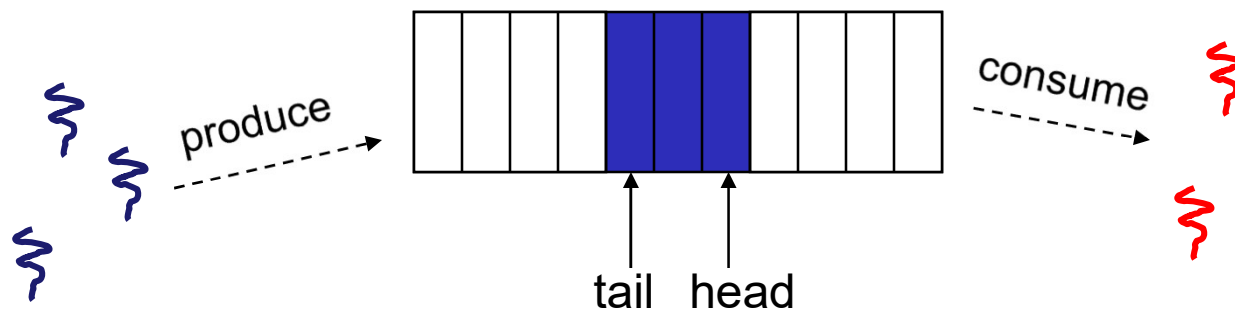
Windows

Pipes

Gary Kimura

Not this pretty picture

- AKA “producer/consumer” problem
 - there is a circular buffer in memory with N entries (slots)
 - producer threads insert entries into it (one at a time)
 - consumer threads remove entries from it (one at a time)
- Threads are concurrent
 - so, we must use synchronization constructs to control access to shared variables describing buffer state



Windows NPFS

- 32 source files

```
aliassup.c  cleanup.c  close.c    create.c   createnp.c  datasup.c
deviosup.c  dir.c      dumpsup.c  eventsup.c fileinfo.c  filobsup.c
flushbuf.c  fsctrl.c  nodetype.h npdata.c   npdata.h    npinit.c
npprocs.h   npstruc.h prefixsup.c read.c      readsup.c   resrcsup.c
secursup.c  seinfo.c  statesup.c strucsup.c volinfo.c   waitsup.c
write.c     writesup.c
```

- 20,000 lines of code (including comments)
- Integrated in the file system model, with added NtCreateNamedPipe() API
- Pipes can be Inbound, Outbound, or Full Duplex

Design considerations

- Named and unnamed pipes
- Message and byte streamed pipes
- How much data to buffer in the kernel
- Use of quotas to keep user from overusing kernel memory
- What if the user tries to write or read more data than is allowed in the pipe at any moment?

Naming Pipes

- Named and unnamed pipes

Message or byte streamed

- Message and byte streamed pipes
- Messages are easy to handle
- Byte streams have some behavior issues. How much to read on a byte stream pipe.

To buffer or not to buffer...

- How much data to buffer in the kernel
- Design Tradeoff

Quotas

- Use of quotas to keep user from overusing kernel memory
- How to handle situations when the user tries to write or read more data than is allowed in the pipe at any moment?

Other Pipe Considerations

- Needed to support the major file/directory operations, Get Info, Set Info, etc.
- Pipe States: Listening, Connected, Disconnected, Closing
- Who comes first, the Reader? Or the Writer? Is the pipe full of data or full of read requests?
- Delayed create until both sides do an open/create

More Pipe Considerations

- What happens when the writer closes the pipe?
What about outstanding writes that are buffered in memory or still in the writer's memory?
- What happens when the reader closes the pipe?
- What happens if the readers asks for less than what's in the pipe? Or more?
- Peeking into a pipe?
- And a whole lot more...