Info Objects Basics

P. J. Denning

For CS471 / CS571

Information Objects

- INFO OBJECT: a container, source, sink, or conveyor of a sequence of bytes
- OS supports three kinds of Info Object:
  - **file**: container
  - **pipe**: synchronized conveyor
  - **device**: source or sink
- All object managers support CREATE and DELETE operations

Info Objects

- Because info objects deal with byte sequences, they have similar operations:
  - OPEN
  - CLOSE
  - READ
  - WRITE
- Can we create a common interface?

OPEN and CLOSE

- oh = OPEN_X(h,rw) --- object of type X
  - Creates a fast channel to the object
  - Makes copy of object in RAM (caching)
  - Open handle ≠ object handle!
  - Example: opening file creates open-file control block containing copy of file index and at least some of the file blocks
- CLOSE_X(oh)

READ and WRITE

- a = READ_X(oh, b, c)
  - Transfers c bytes from position b in the open object to the caller's address space at address a
  - If c = "all", transfers all bytes in the open object up through and including EOF mark
- WRITE_X(oh, a, b, c)
  - Same, with transfer from address space to open object

CREATE and DELETE

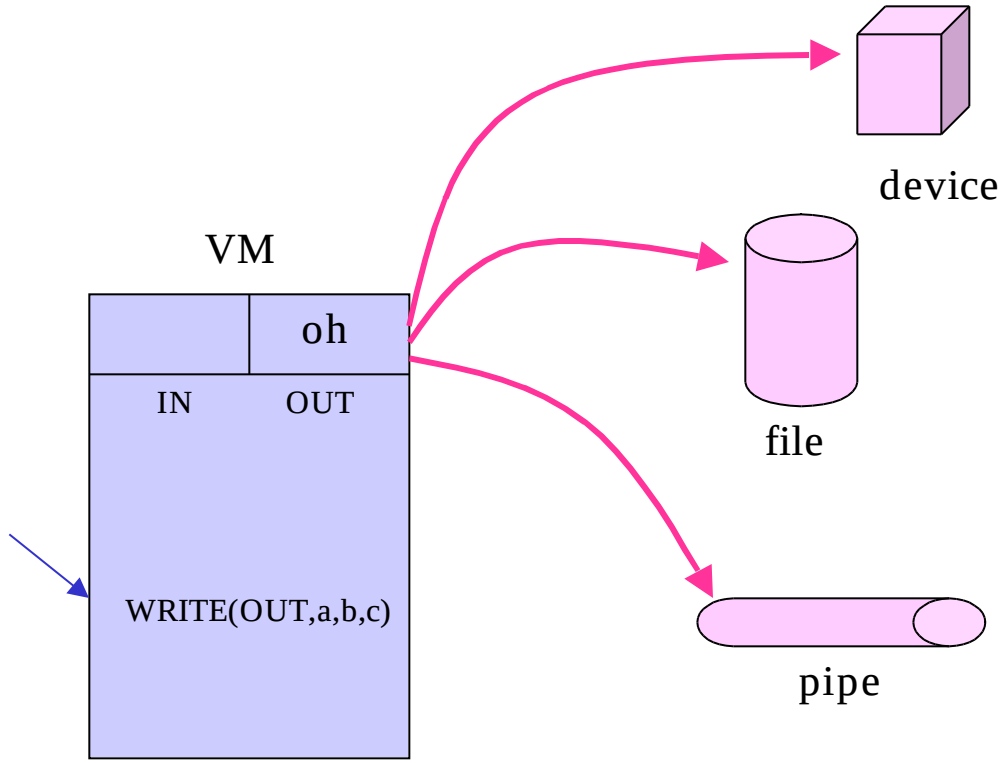
- h = CREATE_X(init state)
  - Creates a new type X object, returns handle
  - For files: allocates index and content blocks on disk
  - For pipes: allocates a pipe control block in pipes manager
  - For devices: detects hardware and installs a driver linked to the hardware
- DELETE_X(h)

Common Interface

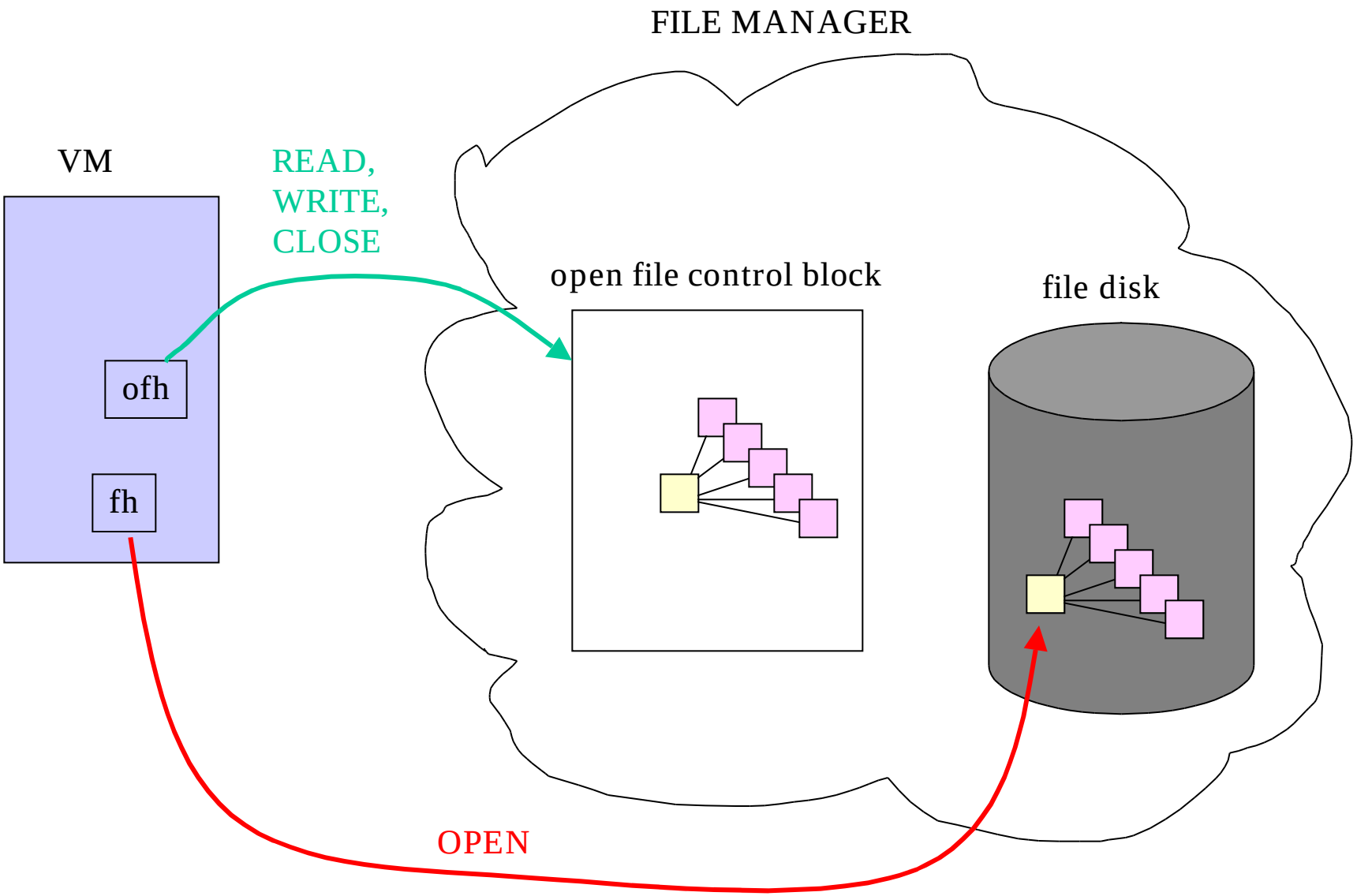
- Instead of OPEN_FILE, OPEN_PIPE, and OPEN_DEV, have just one operation OPEN
- Generic OPEN reads type field of handle
- If type = X, OPEN then behaves like OPEN_X
- Similarly for CLOSE, READ, and WRITE

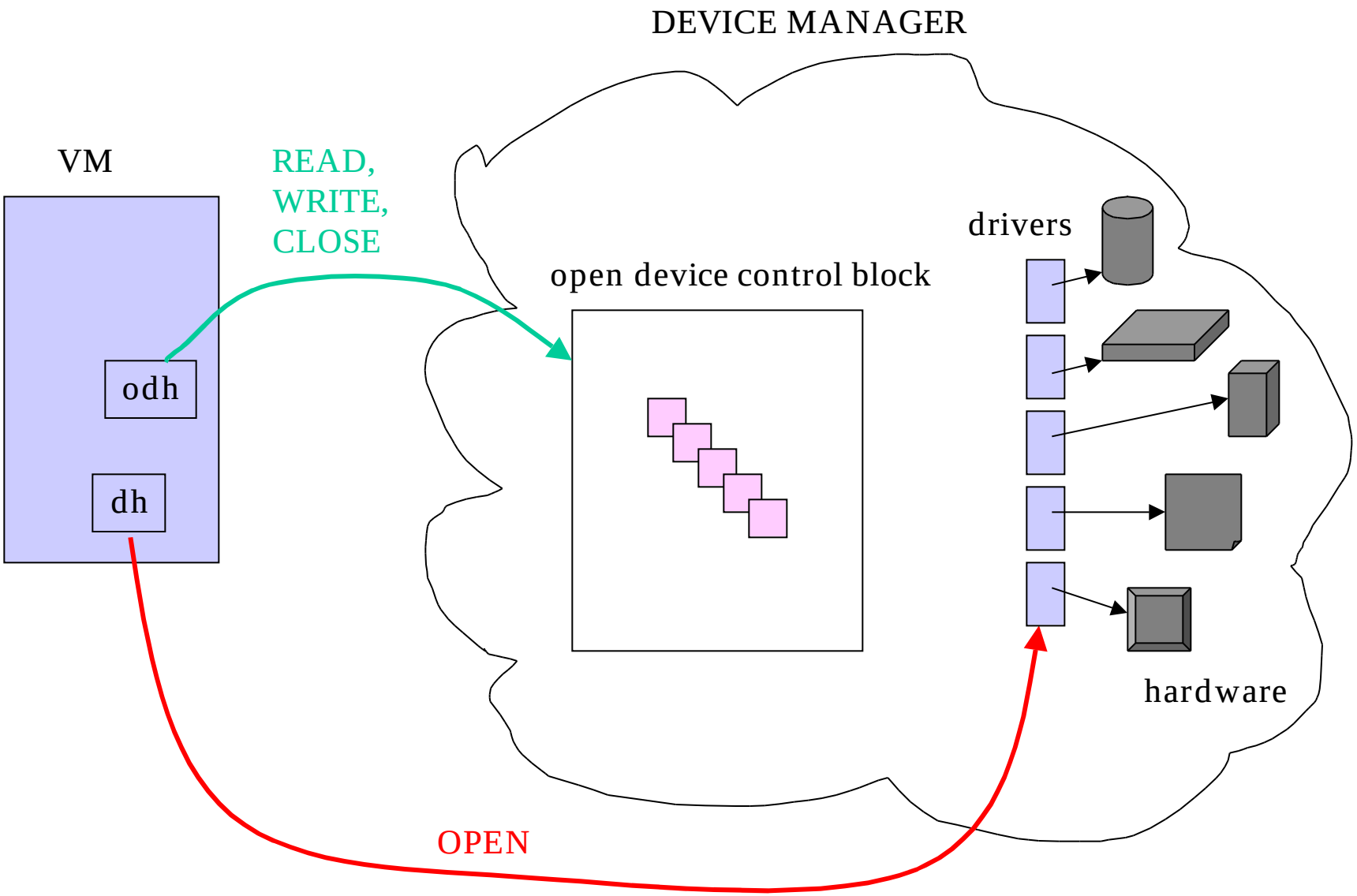
- Common interface enables universal connectivity between virtual machines and info objects
- There are differences because files, pipes, and devices are not exactly alike
- “stream” is the abstraction supported by the uniform interface

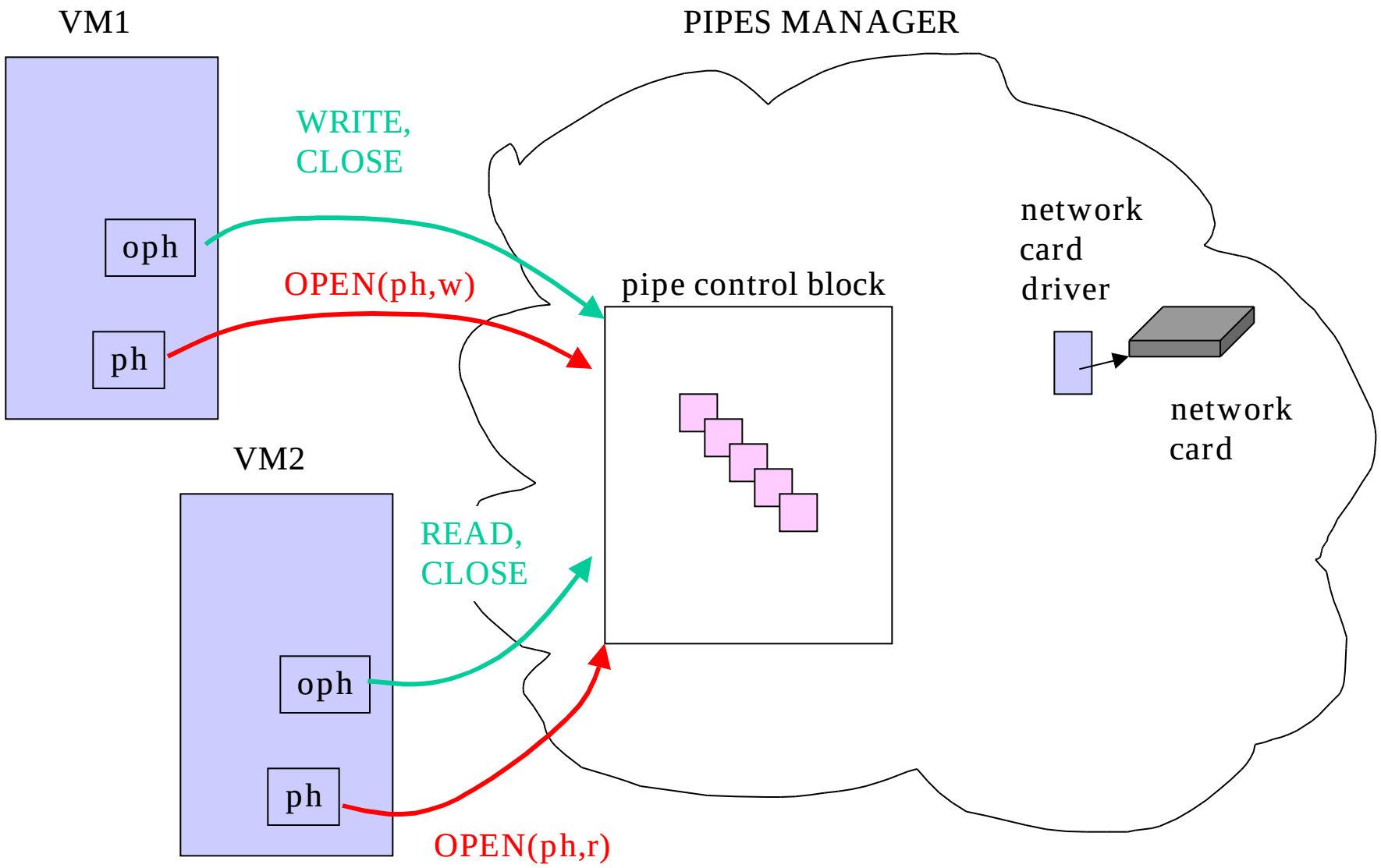
SAME write operation within the VM can read any Info Object without reprogramming or recompiling

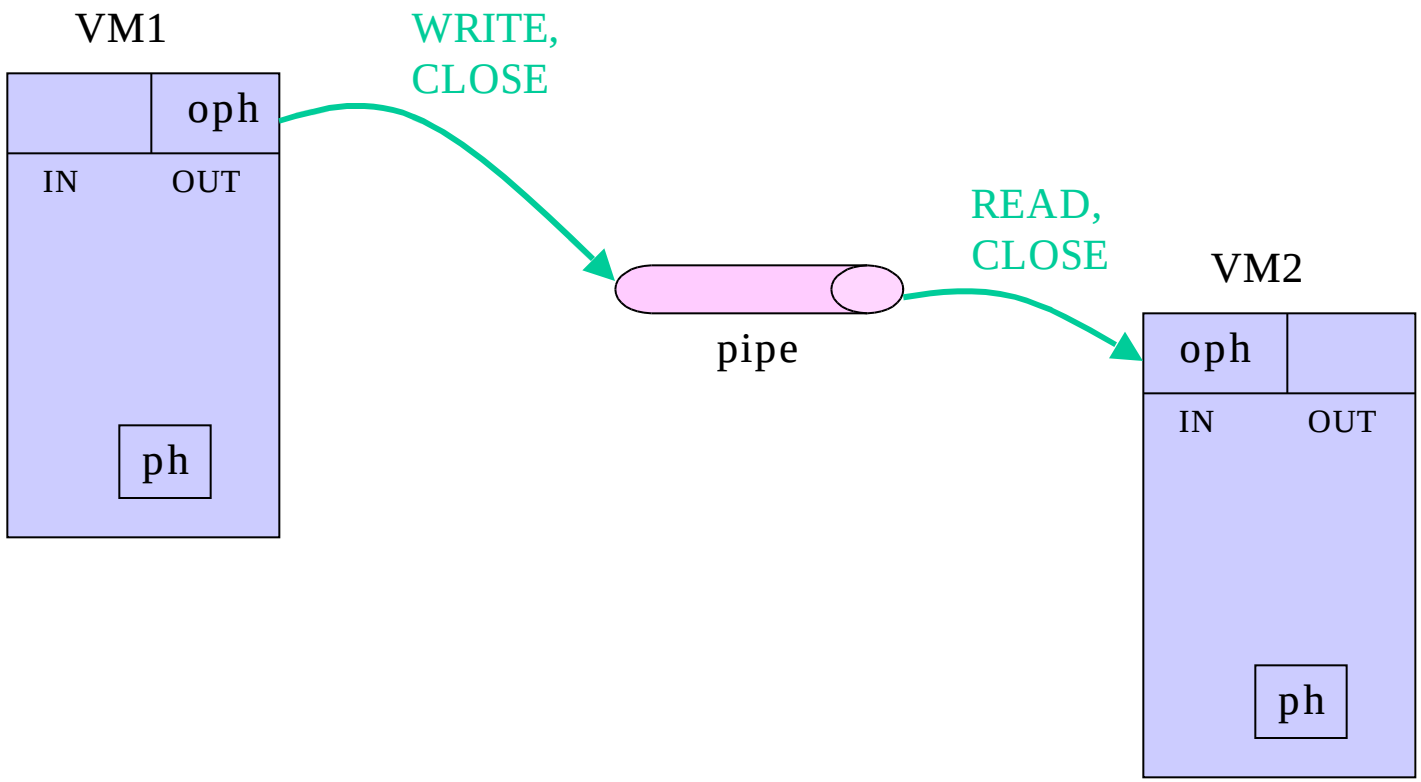


The following pictures illustrate similarities and differences of the O, C, R, and W operations on Info Objects









READ delays

- READ can delay the caller if not all the bytes called for are present.
  - READ(oph,b,7) will delay until the pipe contains 7 or more bytes
  - READ(odh,-,all) will delay until everything up through and including EOF can be returned
- Pipes synchronize concurrent VMs

WRITE wakeups

- WRITE can awaken a delayed reader if enough new bytes are written
  - WRITE(oph,a,b,10) will add 10 bytes and awaken a pipe reader looking for 7 bytes