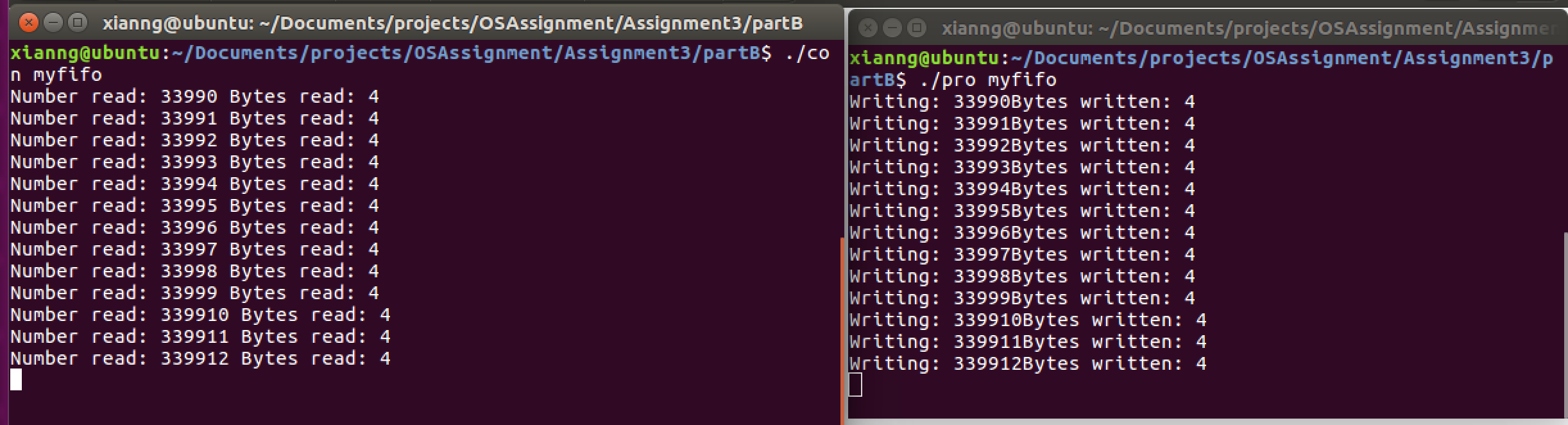
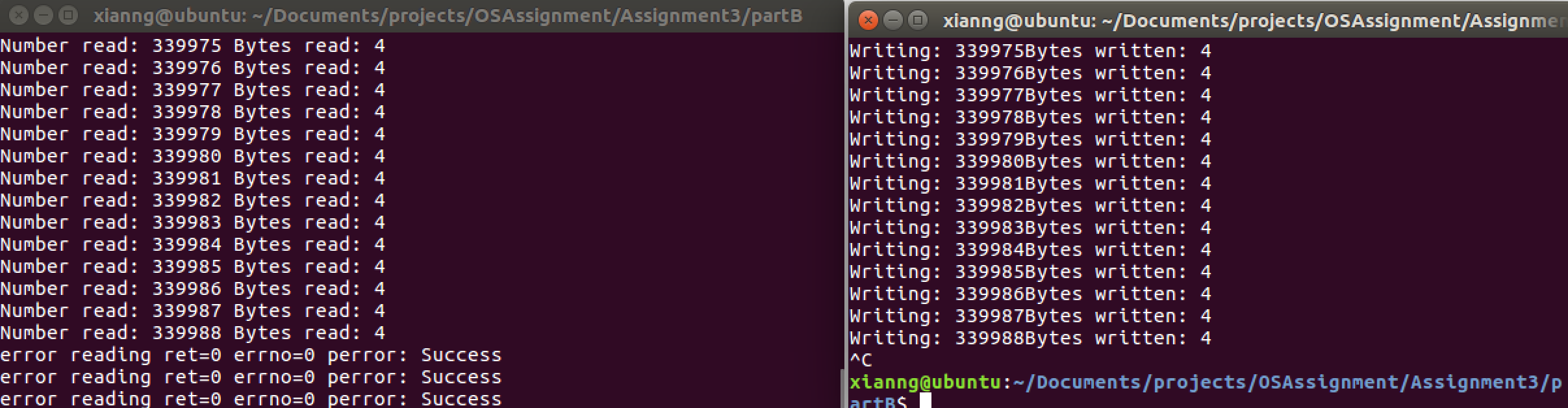
left terminals are Consumers , right ones are Producers

* Run one consumer and one producer concurrently.

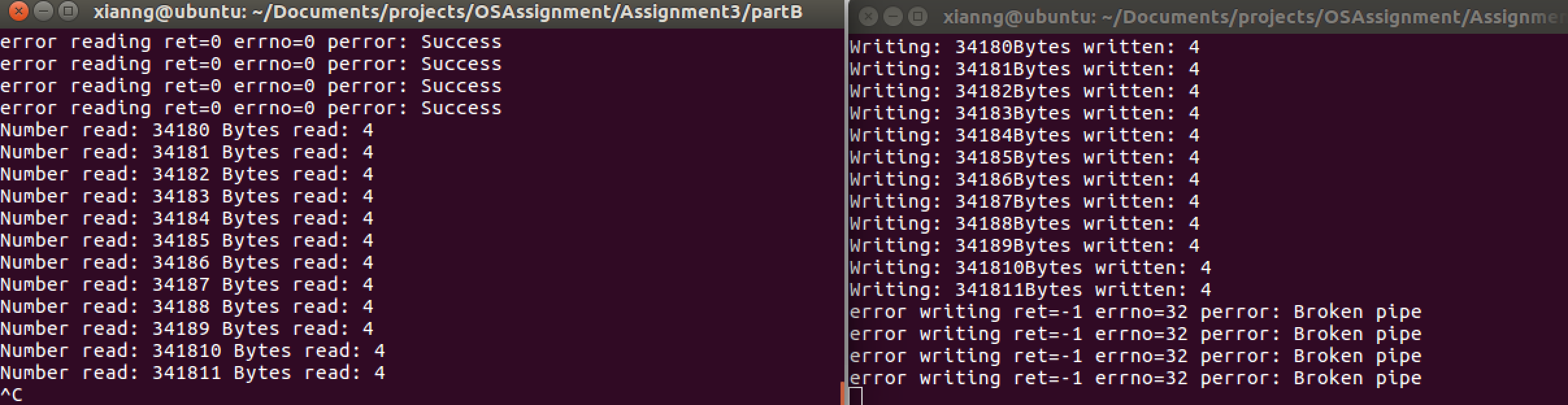


* + Kill the producer with Ctrl-C. Leave consumer running. What happens and why?



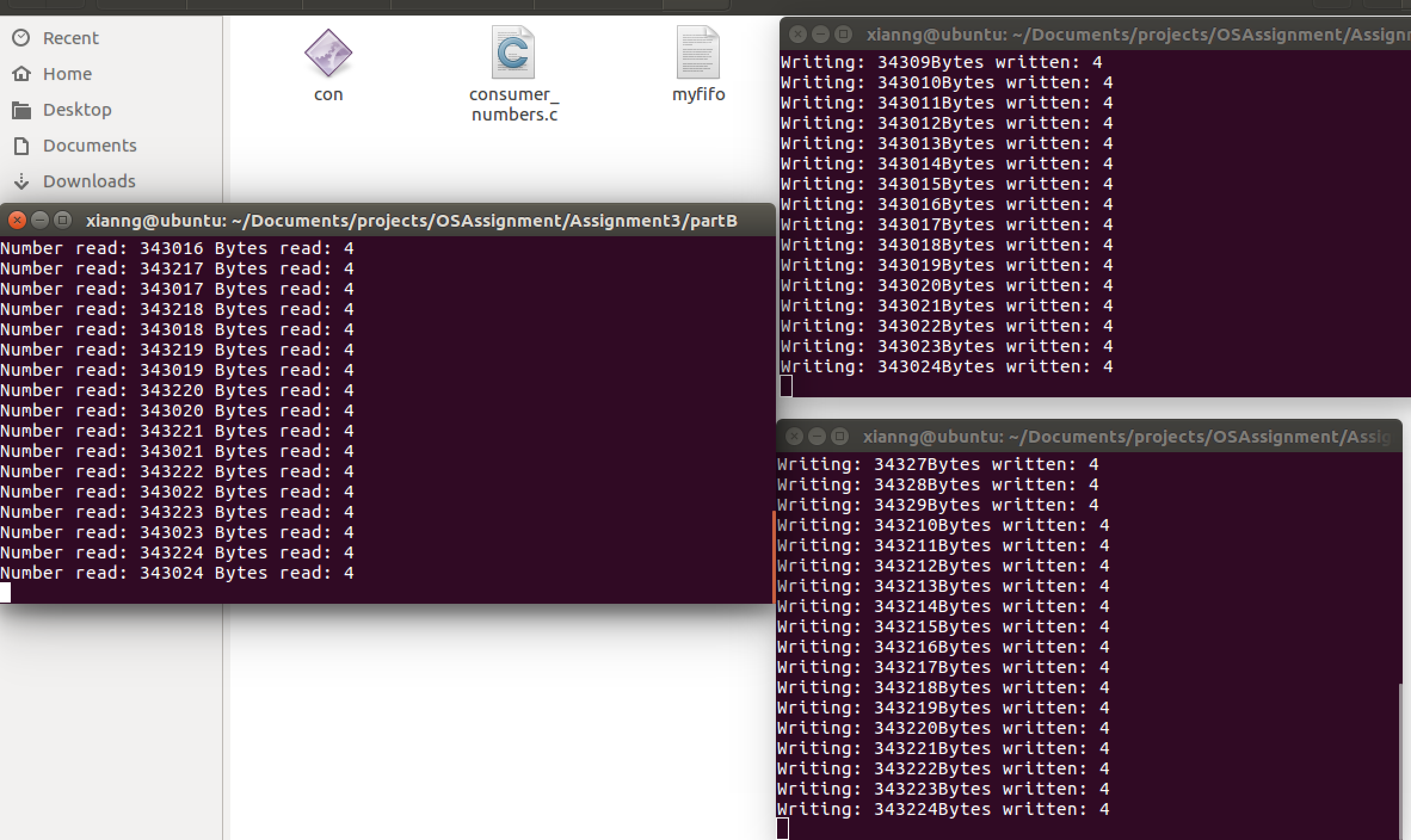
Closing pipe’s writing descriptor will get 0 by calling reading function, which means it read the end of file

* + Kill the consumer with Ctrl-C. Leave producer running. What happens and why?



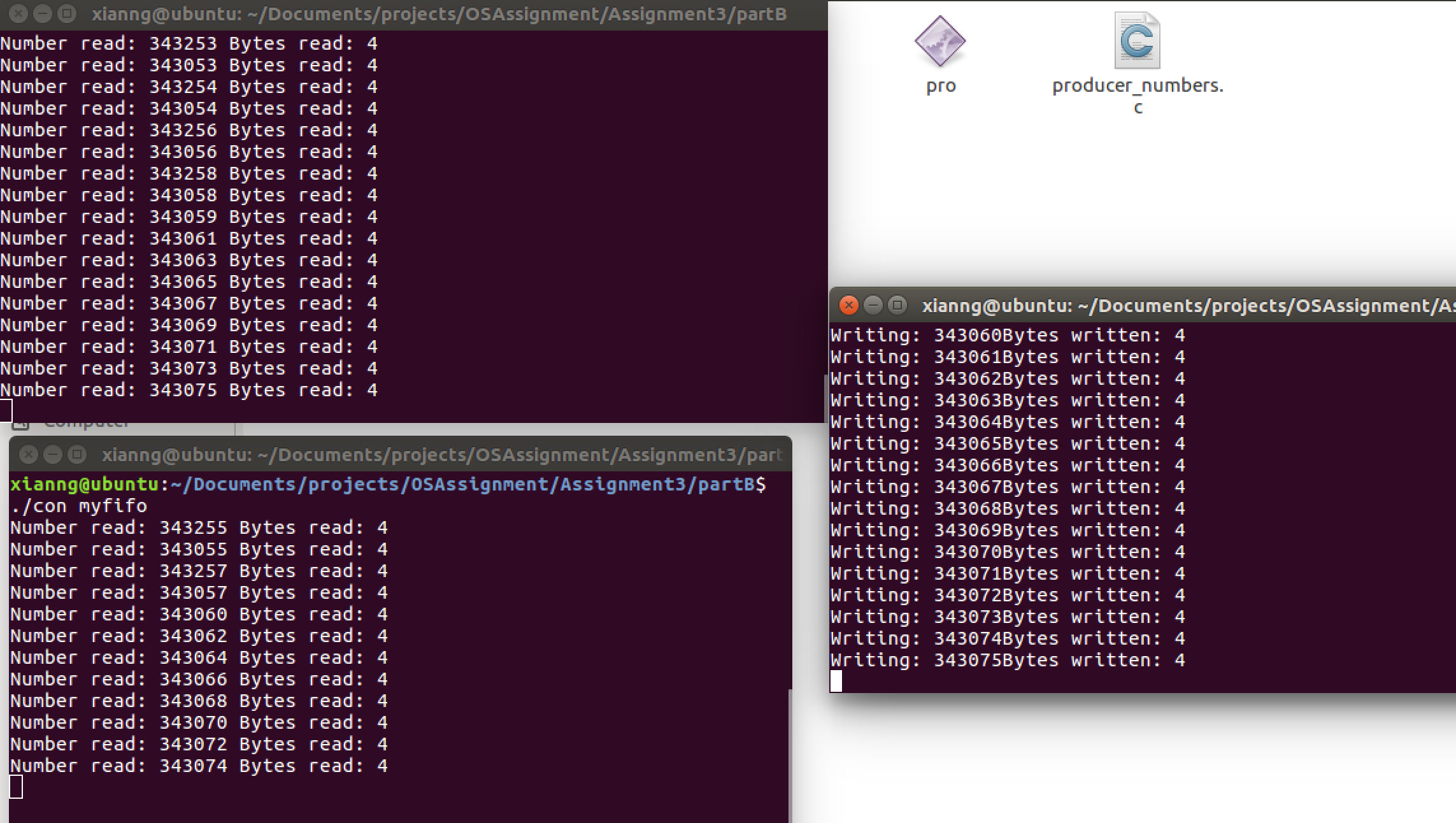
Writing to reading closed pipe will cause SIGPIPE signal to terminate the process. But producer using signal(SIGPIPE, SIG\_IGN) to ignore the pipe SIGPIPE signal, so it continues running and getting -1 return value and setting errno 32(EPIPE) when it invoke write method again.

* Run one consumer and multiple producers concurrently.



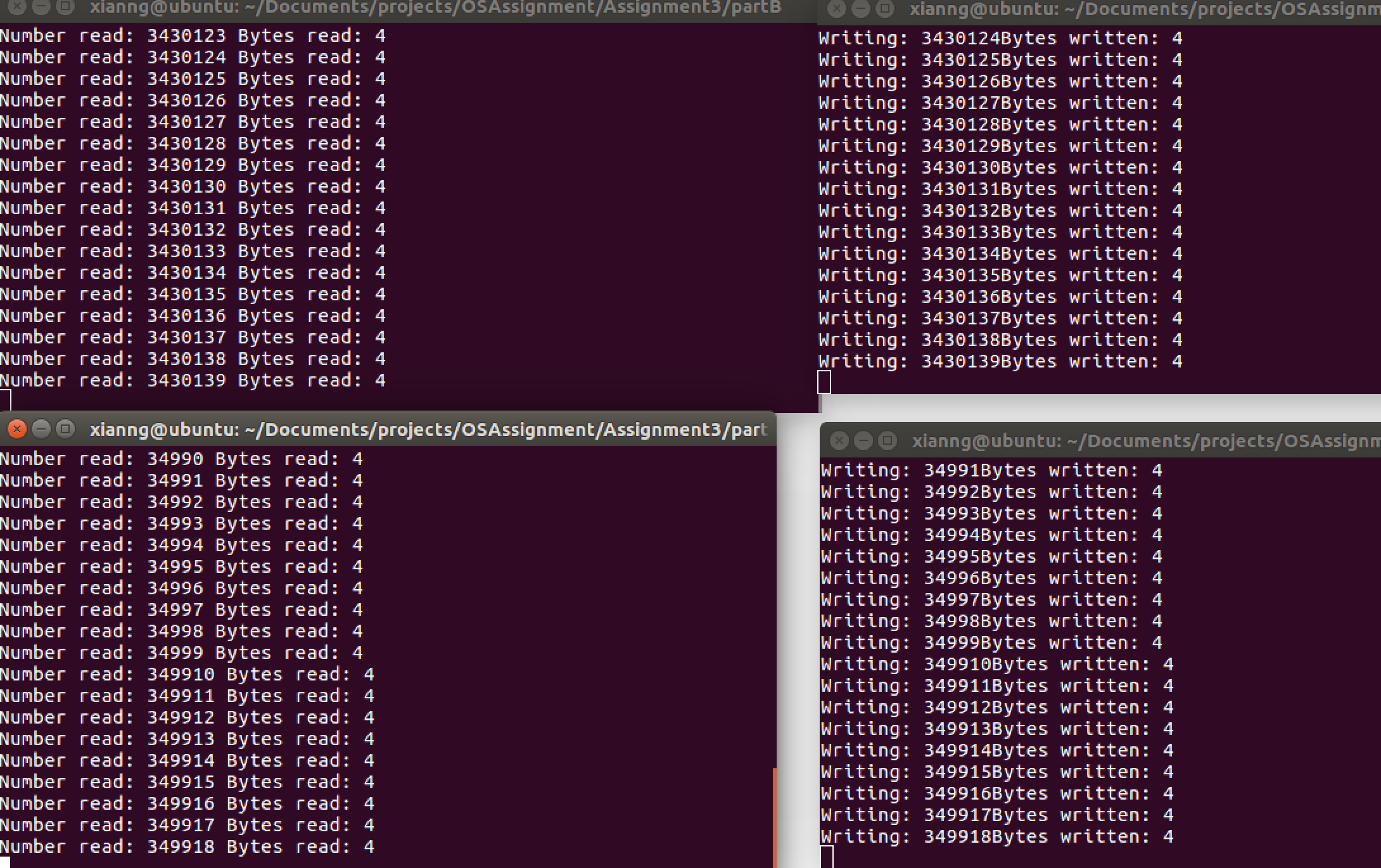
consumer consume two(multiple) data one second! Due to the writing speed is higher than reading speed! Reading data become more.

* Run multiple consumers and one producer concurrently.



each consumer reading speed is lower, because the writing data is not enough for every consumer to read every second.

* Run multiple consumers and multiple producers concurrently.



Basically, reading speed decided by (Num.producer / Num.concumer).

PartC plan:

Build a new device to solve this consumer and producer problem! P and C access to the shared resources and by using SEMAPHORE and MUTEX to implement the mechanism of locking the resource to prevent concurrent modification as well as inform the other side data is available.