

## Fall 2024 CIS 620 Homework 4

(Due Oct. 31)

### 4-3 EXERCISE

A classic operating system problem is that of coordinating a producer and consumer process. The producer *produces* a value and stores the value (such as in a common buffer or file) that can *hold only one* of the items produced. The consumer obtains (in a nondestructive manner) the value from the storage location and *consumes* it. The producer and consumer work at different rates. To guarantee integrity, each value *produced* must be *consumed* (not lost via overwriting by a speedy producer with a slow consumer), and no value should be consumed twice (such as when the consumer is faster than the producer). Write a producer/consumer process pair that uses a lock file communication technique to coordinate their activities. To ensure that no data is lost or duplicated, the producer process should produce its values by reading them one-by-one from an input file and in turn storing them in the common location. The consumer should append the values it consumes (reads from the common location) to an output file. After processing, say, 10, unique values, both the input file for the producer and the output file for the consumer should be identical. Use the **sleep** library call with small random values to simulate the producer and consumer working at different rates.

One way to solve the problem is to use two lock files. When using two lock files, one file would indicate whether or not the number has been produced, and the second file would indicate if the number has been consumed. The activities of the two processes to be coordinated can be summarized as follows:

```
Producer
do
    sleep random amount
    read a number from input file
    if # has been consumed
        write number to common buffer
        indicate new # produced
until 10 numbers produced

Consumer
do
    sleep random amount
    if a new # produced
        read number from common buffer
        indicate # was consumed
        append number read to output file
until 10 numbers produced
```

*Hint:* When using lock files, we test whether or not we can create a lock file. Thus, we could use the successful creation of the lock file as an indication of access and the inability to create the lock file as a prohibition of access. Using this approach initially, the lock file indicating a number that has been consumed would be absent, and the lock file indicating a new number that has been produced would be present.

## Turning it in

Zip all your source files into a file called ex4-3.xxxxxxx.zip where xxxxxxxx is your CSU ID. Turnin the zipped file on the **grail** machine using the following command:

```
turnin -c cis620x -p hwch4b ex4-3.xxxxxxx.zip
```

Start on time and good luck. If you have any questions, send e-mail to [a.shetty13@vikes.csuohio.edu](mailto:a.shetty13@vikes.csuohio.edu) or [w.xiong15@csuohio.edu](mailto:w.xiong15@csuohio.edu).