#1

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

The below error message starts displaying on the consumer terminal in countinuous

loop:

error reading ret=0 errno=0 perror: Success

Reason:

Since producer was killed, it did not generate any number. Nothing was getting written in the pipe. Therefore there was nothing left to be consumed. But since consumer was not terminated. It kept looking for items to be consumed. And since read operation failed in every iteration after this. It kept displaying error message.

#2

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

The below error message starts displaying on the producer terminal in countinuous

loop:

error writing ret=-1 errno=32 perror: Broken pipe

Reason:

Since consumer was killed, the consumer process stopped consuming numbers. Nothing was getting read out of the pipe. Since producer was still executing it attempted to produce items in every iteration. But failed to write the item to the pipe. Since write operation failed in every iteration, the program kept displaying error message.

#3

Run one consumer and multiple producers concurrently.

Producers are concurrently producing items.

Example 3 concurrently executing producing producers and 1 consumer was taken.

The producers 1,2,3 produced number in following sequences

Producer 1 generated 67220,67221,67222,67223,67224,67225 etc

Producer 2 generated 67230,67231,67232,67233,67234,67235 etc

Producer 3 generated 67240,67241,67242,67243,67244,67244 etc

Consumer 1 consumed numbers in following sequence:

67220,67230,67240,67221,67231,672241,67222,67232,67242,67223,67233,67243 and so on

#4

Run multiple consumers and one producer concurrently.

Result observed:

One producer and 4 consumers where executed concurrently.

When the 1st consumer was started it read numbers in original sequence of producer.

When the 2nd consumer was introduced the work was shared between consumer 1&2.

One of the consumer was observed reading only odd numbers.

One of the consumer was observed reading only even numbers.

It must mean that alternate numbers were being read by each consumer.

For example when producer writes 491510,491511,491512,491513,491514 etc

The number 491510 is read by 1st consumer. 2nd consumer reads 491511. 1st one reads 491512 and so on.

When a 3rd consumer is started concurrently with consumer 1 & 2, the 3rd consumer initially consumes a few numbers.

It was observed that the 3rd consumer stopped reading roughly after reading 3 numbers, It seems to be waiting(spining) for something to be produced.

Similarly when 4th consumer was started it showed the same behaviour as the 3rd consumer.

It was observed that the 4th consumer stopped reading roughly after 2 numbers, It seemed to be waiting for something to be produced. It means that there are not enough numbers produced by producer to be read by continuously by consumers 3 and 4.

This exercise showed that the consumers became greedy.

Consumers tried to read items even before they were written by the producer.

When multiple consumers are reading data from a file without using any locks then race condition can occur. And race condition can sometimes lead to a deadlock situation.

#5

Run multiple consumers and multiple producers concurrently.

Result observed:

Numbers were being generated and read in any sequence.