Derek Lee, Shine Li

Professor Hakner

Problem 2 Sample Runs

Preface:

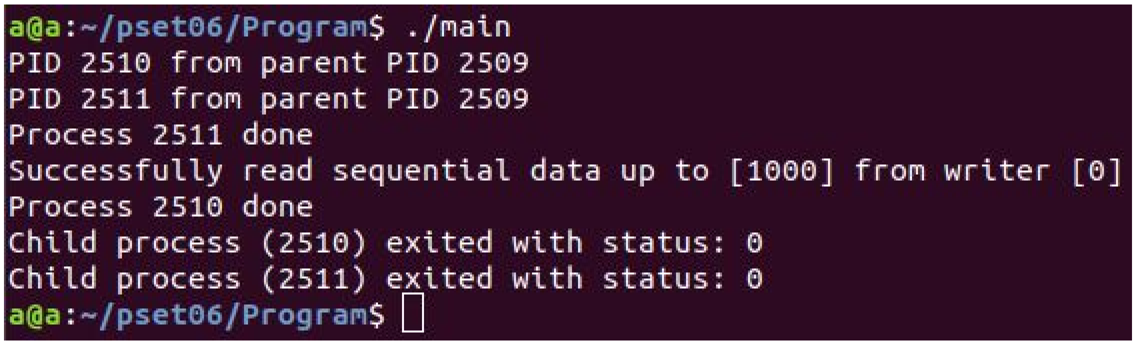
"main" was written to test the functionality of the FIFO. It was assumed that there would be exactly one reader. The number of writers, as well as the number of write operations each writer performs, can be specified within main.c (via the #defined variables N_WRITER and N_WRCOUNT, respectively). Each of the [N_WRITER] writers performs [N_WRCOUNT] writes to the FIFO, while the reader process is to perform [N_WRITER * N_WRCOUNT] reads from the FIFO.

The writer processes send datum, in which the following information is encoded:

- their "writer number" - an integer ranging from [0] to [N_WRITER - 1], used to represent which of the [N_WRITER] writers are sending the datum
- data in the form of an integer. Each writer process starts by sending 0, 1, 2… up to [N_WRCOUNT - 1], such that on the Nth fifo_wr() operation a given writer process performs, the data [N - 1] is sent.

The singular reader process reads in datum, decoding it into a writer number, and the data. The reader expects to receive data in sequential order from each writer process - any missing or out-of-order data received by the reader will be reported in an error message. Before the reader process returns, it reports how much sequential data was successfully read from each of the writer processes. Note that other debugging and diagnostic write-outs were left in the program.

I. Single reader, Single writer (1k[1] writes per writer)

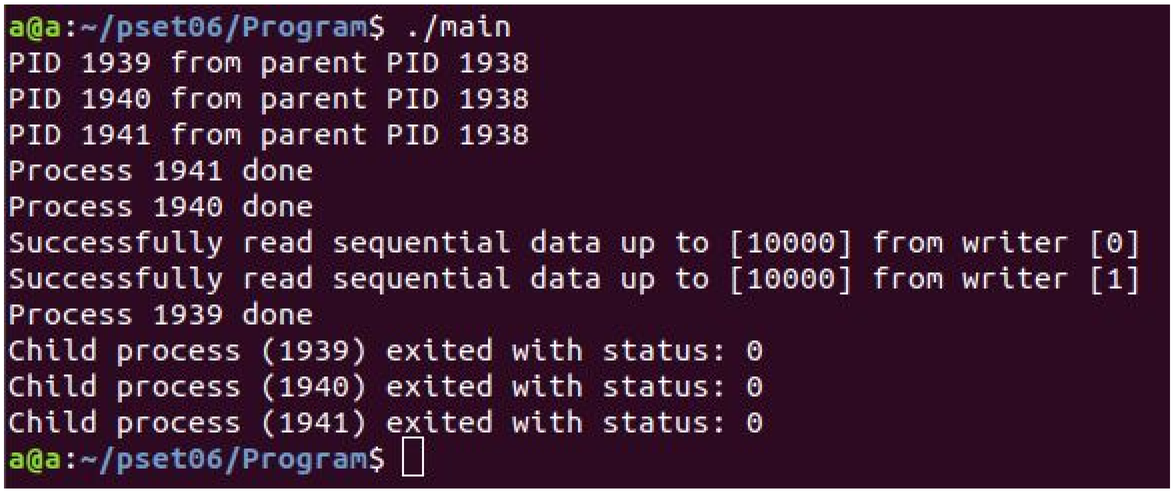


```
a@a:~/pset06/Program$ ./main
PID 2510 from parent PID 2509
PID 2511 from parent PID 2509
Process 2511 done
Successfully read sequential data up to [1000] from writer [0]
Process 2510 done
Child process (2510) exited with status: 0
Child process (2511) exited with status: 0
a@a:~/pset06/Program$ 
```

Figure 1: Test I output

Test run 1 is the simplest case of one writer and one reader. This is intended to test the basic functionality of the FIFO.

II. Single reader, Two writers (10k writes per writer)



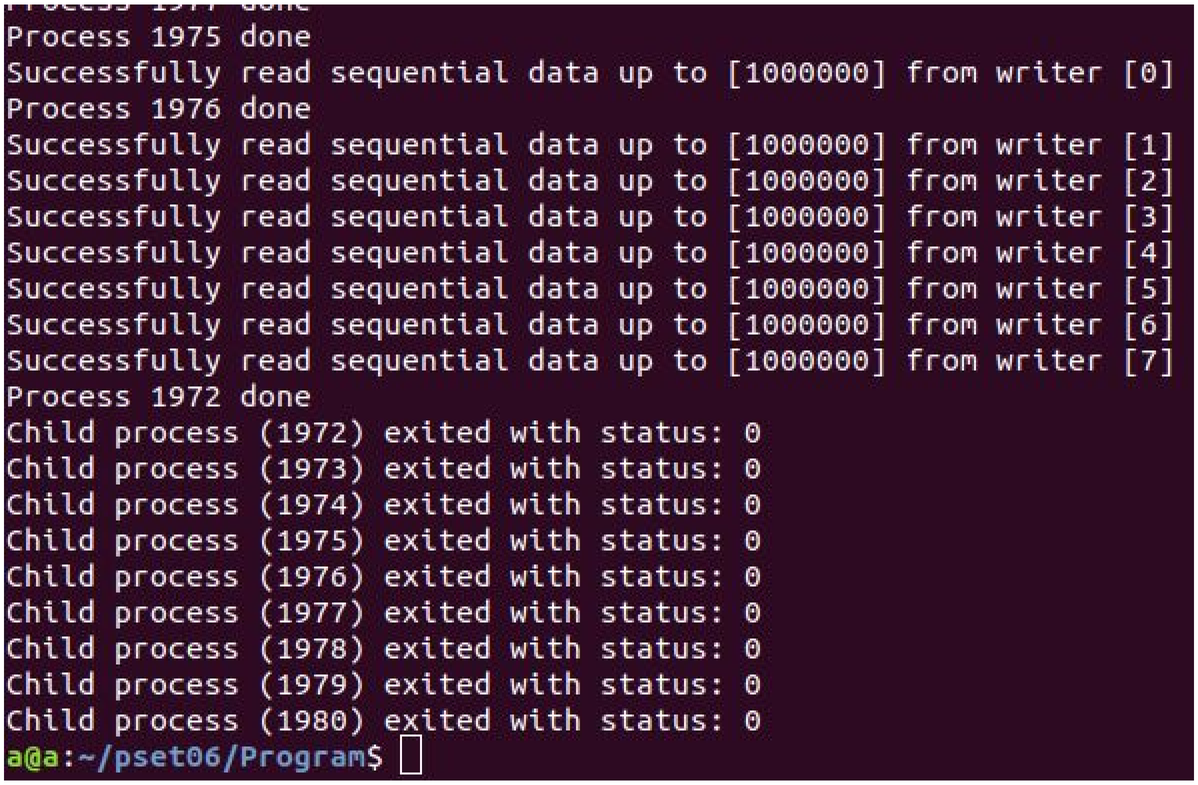
```
a@a:~/pset06/Program$ ./main
PID 1939 from parent PID 1938
PID 1940 from parent PID 1938
PID 1941 from parent PID 1938
Process 1941 done
Process 1940 done
Successfully read sequential data up to [10000] from writer [0]
Successfully read sequential data up to [10000] from writer [1]
Process 1939 done
Child process (1939) exited with status: 0
Child process (1940) exited with status: 0
Child process (1941) exited with status: 0
a@a:~/pset06/Program$ 
```

Figure 2: Test II output

Test run 2 is a slightly more complicated case of one reader, and two writers. This is intended to test the FIFO's ability to coordinate between multiple writers, and to demonstrate that the FIFO data structures' integrity was maintained. The number of write operations was also increased such that the total number of writes exceeded the MYFIFO_BUFSIZ of 4096; this tested the functionality of the FIFO's circular buffer.

[1] Usage of the prefixes 'k' and 'M' in this document will refer to decimal 1,000 and 1,000,000 respectively.

III. Single reader, Eight writers (1M writes per writer)

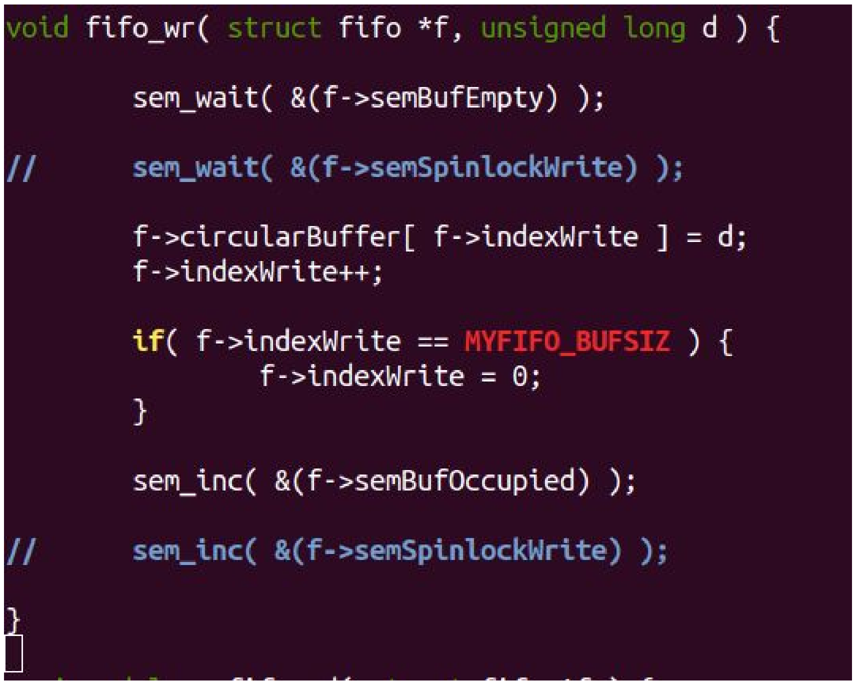


```
Process 1977 done
Process 1975 done
Successfully read sequential data up to [1000000] from writer [0]
Process 1976 done
Successfully read sequential data up to [1000000] from writer [1]
Successfully read sequential data up to [1000000] from writer [2]
Successfully read sequential data up to [1000000] from writer [3]
Successfully read sequential data up to [1000000] from writer [4]
Successfully read sequential data up to [1000000] from writer [5]
Successfully read sequential data up to [1000000] from writer [6]
Successfully read sequential data up to [1000000] from writer [7]
Process 1972 done
Child process (1972) exited with status: 0
Child process (1973) exited with status: 0
Child process (1974) exited with status: 0
Child process (1975) exited with status: 0
Child process (1976) exited with status: 0
Child process (1977) exited with status: 0
Child process (1978) exited with status: 0
Child process (1979) exited with status: 0
Child process (1980) exited with status: 0
a@a:~/pset06/Program$
```

Figure 3: Test III output

Test run 3 is the so-called "acid test". After test 2 we can be reasonably confident that the FIFO can coordinate between multiple writers. However, in order to catch more insidious errors such as the infamous "lost-wakeup" bug, it is necessary to run millions of write-read operations. This was done successfully in test 3, which involved eight writers, each writing one million times to the FIFO. Test 3 was repeated over ten times (not documented here), and successfully ran to completion without hanging in each try.

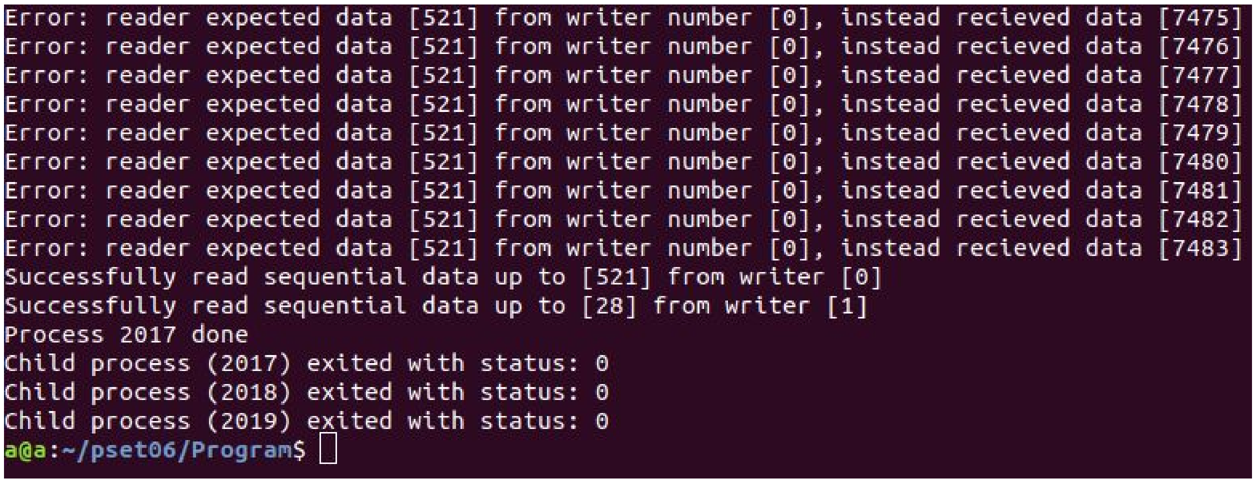
IV. Single reader, Two writers (10k writes per writer), w/ improper locking



```
void fifo_wr( struct fifo *f, unsigned long d ) {

        sem_wait( &(f->semBufEmpty) );

//      sem_wait( &(f->semSpinlockWrite) );

        f->circularBuffer[ f->indexWrite ] = d;
        f->indexWrite++;

        if( f->indexWrite == MYFIFO_BUFSIZ ) {
                f->indexWrite = 0;
        }

        sem_inc( &(f->semBufOccupied) );

//      sem_inc( &(f->semSpinlockWrite) );

}
```

Figure 4: Altered section of [fifo.c].

In order to demonstrate failure, the section of code responsible for mutex locking the FIFO data structure was disabled in fifo_wr(), as seen above.



```
Error: reader expected data [521] from writer number [0], instead recieved data [7475]
Error: reader expected data [521] from writer number [0], instead recieved data [7476]
Error: reader expected data [521] from writer number [0], instead recieved data [7477]
Error: reader expected data [521] from writer number [0], instead recieved data [7478]
Error: reader expected data [521] from writer number [0], instead recieved data [7479]
Error: reader expected data [521] from writer number [0], instead recieved data [7480]
Error: reader expected data [521] from writer number [0], instead recieved data [7481]
Error: reader expected data [521] from writer number [0], instead recieved data [7482]
Error: reader expected data [521] from writer number [0], instead recieved data [7483]
Successfully read sequential data up to [521] from writer [0]
Successfully read sequential data up to [28] from writer [1]
Process 2017 done
Child process (2017) exited with status: 0
Child process (2018) exited with status: 0
Child process (2019) exited with status: 0
a@a:~/pset06/Program$ 
```

Figure 5: Test IV output. Note the missing datum from writers [0] and [1].

Test 2 conditions were emulated (2 writers, 10,000 writes each). As a result of the aforementioned modifications to fifo_wr(), the integrity of the FIFO circular buffer was not protected, and the two writer processes wrote over each other's data. In the example shown above, writer 0 sent the data "521" to the FIFO, but this must have been overwritten by writer 1. The reader, expecting "521" from writer 1, reports an error when further data is received.