CS474 Operating Systems Project #2

Michael Smith

Computer Science Dept, NMSU

p173939@nmsu.edu

***Abstract*— This document describes the process needed to ensure mutually excluded access to a shared buffer and safe access to such buffer.**

I. Introduction

A program written in the C Language “buffer.c” was written to accomplish the goal of demonstrating how a shared buffer, when protected adequately with mutual exclusion using semaphores, can be access safely with different threads.

II. Program

“buffer.c” first initializes three semaphores, “full”, “empty” and “mutex”. Then it creates 2 threads, the producer and the consumer. These threads are then joined so that the parent process that create them will only continue when both of the created threads have finished. A shared memory buffer “buffer” with 15 slots is also created and shared between the two threads. The consumer thread immediately goes into waiting but the producer thread starts reading from file “mytest.dat” character by character and inserts them into the shared buffer until the buffer is full the it signals the “full” and “mutex” semaphores and goes into waiting. The consumer then takes over and prints each character in the buffer until it is empty then signals “empty” and “mutex” and goes into waiting. The process continues until the producer reads EOF and writes it to the buffer then signals “full” and “mutex” and the thread ends. The consumer then prints out the last remaining characters and then ends itself. Control is then granted back to the parent process and it releases the semaphores and the shared buffer.

III. results

The program will print any string of characters, 15 at a time, one for each line until the end of file is reached. Each character printed will only print in increments of 1 second due to a sleep(1) present between each print in the consumer thread. An example output for a file containing the ABCs is listed below:

```
writing A to buffer
writing B to buffer
writing C to buffer
writing D to buffer
writing E to buffer
writing F to buffer
writing G to buffer
writing H to buffer
writing I to buffer
writing J to buffer
writing K to buffer
writing L to buffer
writing M to buffer
writing N to buffer
writing O to buffer
Buffer full transferring to consumer
Reading A from buffer
Reading B from buffer
Reading C from buffer
Reading D from buffer
Reading E from buffer
Reading F from buffer
Reading G from buffer
Reading H from buffer
Reading I from buffer
Reading J from buffer
Reading K from buffer
Reading L from buffer
Reading M from buffer
Reading N from buffer
Reading O from buffer
Buffer read, transferring control to producer
writing P to buffer
writing Q to buffer
writing R to buffer
writing S to buffer
writing T to buffer
writing U to buffer
writing V to buffer
writing W to buffer
writing X to buffer
writing Y to buffer
writing Z to buffer
File completely read, wrote EOF and transferred control to consumer
Reading P from buffer
Reading Q from buffer
Reading R from buffer
Reading S from buffer
Reading T from buffer
Reading U from buffer
Reading V from buffer
Reading W from buffer
Reading X from buffer
Reading Y from buffer
Reading Z from buffer
read EOF, exiting
```

It is important to note that the producer does not 1 second between each of its writes to the buffer and to the user it will be instantaneous. There is a limit of 150 characters according to the assignment an example with 150 characters is given below:

```
writing R to buffer
writing o to buffer
writing s to buffer
writing e to buffer
writing s to buffer
writing   to buffer
writing a to buffer
writing r to buffer
writing e to buffer
writing   to buffer
writing r to buffer
writing e to buffer
writing d to buffer
writing , to buffer
writing
 to buffer
Buffer full transferring to consumer
Reading R from buffer, value = 82
Reading o from buffer, value = 111
Reading s from buffer, value = 115
Reading e from buffer, value = 101
Reading s from buffer, value = 115
Reading   from buffer, value = 32
Reading a from buffer, value = 97
Reading r from buffer, value = 114
Reading e from buffer, value = 101
Reading   from buffer, value = 32
Reading r from buffer, value = 114
Reading e from buffer, value = 101
Reading d from buffer, value = 100
Reading , from buffer, value = 44
Reading
 from buffer, value = 10
Buffer read, transferring control to producer
writing V to buffer
writing i to buffer
writing o to buffer
writing l to buffer
writing e to buffer
writing t to buffer
writing s to buffer
writing   to buffer
writing a to buffer
writing r to buffer
writing e to buffer
writing   to buffer
writing B to buffer
writing l to buffer
writing u to buffer
Buffer full transferring to consumer
Reading V from buffer, value = 86
Reading i from buffer, value = 105
Reading o from buffer, value = 111
Reading l from buffer, value = 108
Reading e from buffer, value = 101
Reading t from buffer, value = 116
Reading s from buffer, value = 115
Reading   from buffer, value = 32
Reading a from buffer, value = 97
Reading r from buffer, value = 114
Reading e from buffer, value = 101
Reading   from buffer, value = 32
Reading B from buffer, value = 66
Reading l from buffer, value = 108
Reading u from buffer, value = 117
Buffer read, transferring control to producer
writing e to buffer
writing , to buffer
writing
 to buffer
writing C to buffer
writing o to buffer
writing m to buffer
writing p to buffer
writing u to buffer
writing t to buffer
writing e to buffer
writing r to buffer
writing   to buffer
writing S to buffer
writing c to buffer
writing i to buffer
Buffer full transferring to consumer
Reading e from buffer, value = 101
Reading , from buffer, value = 44
Reading
 from buffer, value = 10
Reading C from buffer, value = 67
Reading o from buffer, value = 111
Reading m from buffer, value = 109
Reading p from buffer, value = 112
Reading u from buffer, value = 117
Reading t from buffer, value = 116
Reading e from buffer, value = 101
Reading r from buffer, value = 114
Reading   from buffer, value = 32
Reading S from buffer, value = 83
Reading c from buffer, value = 99
Reading i from buffer, value = 105
Buffer read, transferring control to producer
writing e to buffer
writing n to buffer
writing c to buffer
writing e to buffer
writing   to buffer
writing i to buffer
writing s to buffer
writing   to buffer
writing f to buffer
writing u to buffer
writing n to buffer
writing , to buffer
writing
 to buffer
writing A to buffer
writing n to buffer
Buffer full transferring to consumer
Reading e from buffer, value = 101
Reading n from buffer, value = 110
Reading c from buffer, value = 99
Reading e from buffer, value = 101
Reading   from buffer, value = 32
Reading i from buffer, value = 105
Reading s from buffer, value = 115
Reading   from buffer, value = 32
Reading f from buffer, value = 102
Reading u from buffer, value = 117
Reading n from buffer, value = 110
Reading , from buffer, value = 44
Reading
 from buffer, value = 10
Reading A from buffer, value = 65
Reading n from buffer, value = 110
Buffer read, transferring control to producer
writing d to buffer
writing   to buffer
writing s to buffer
writing o to buffer
writing   to buffer
writing a to buffer
writing r to buffer
writing e to buffer
writing   to buffer
writing y to buffer
writing o to buffer
writing u to buffer
writing . to buffer
writing
 to buffer
writing
 to buffer
Buffer full transferring to consumer
Reading d from buffer, value = 100
Reading   from buffer, value = 32
Reading s from buffer, value = 115
Reading o from buffer, value = 111
Reading   from buffer, value = 32
Reading a from buffer, value = 97
Reading r from buffer, value = 114
Reading e from buffer, value = 101
Reading   from buffer, value = 32
Reading y from buffer, value = 121
Reading o from buffer, value = 111
Reading u from buffer, value = 117
Reading . from buffer, value = 46
Reading
 from buffer, value = 10
Reading
 from buffer, value = 10
Buffer read, transferring control to producer
writing T to buffer
writing h to buffer
writing i to buffer
writing s to buffer
writing   to buffer
writing i to buffer
writing s to buffer
writing   to buffer
writing m to buffer
writing o to buffer
writing r to buffer
writing e to buffer
writing   to buffer
writing t to buffer
writing e to buffer
Buffer full transferring to consumer
Reading T from buffer, value = 84
Reading h from buffer, value = 104
Reading i from buffer, value = 105
Reading s from buffer, value = 115
Reading   from buffer, value = 32
Reading i from buffer, value = 105
Reading s from buffer, value = 115
Reading   from buffer, value = 32
Reading m from buffer, value = 109
Reading o from buffer, value = 111
Reading r from buffer, value = 114
Reading e from buffer, value = 101
Reading   from buffer, value = 32
Reading t from buffer, value = 116
Reading e from buffer, value = 101
Buffer read, transferring control to producer
writing x to buffer
writing t to buffer
writing   to buffer
writing t to buffer
writing o to buffer
writing   to buffer
writing f to buffer
writing i to buffer
writing l to buffer
writing l to buffer
writing   to buffer
writing t to buffer
writing h to buffer
writing e to buffer
writing   to buffer
Buffer full transferring to consumer
Reading x from buffer, value = 120
Reading t from buffer, value = 116
Reading   from buffer, value = 32
Reading t from buffer, value = 116
Reading o from buffer, value = 111
Reading   from buffer, value = 32
Reading f from buffer, value = 102
Reading i from buffer, value = 105
Reading l from buffer, value = 108
Reading l from buffer, value = 108
Reading   from buffer, value = 32
Reading t from buffer, value = 116
Reading h from buffer, value = 104
Reading e from buffer, value = 101
Reading   from buffer, value = 32
Buffer read, transferring control to producer
writing 1 to buffer
writing 5 to buffer
writing 0 to buffer
writing   to buffer
writing c to buffer
writing h to buffer
writing a to buffer
writing r to buffer
writing a to buffer
writing c to buffer
writing t to buffer
writing e to buffer
writing r to buffer
writing   to buffer
writing l to buffer
Buffer full transferring to consumer
Reading 1 from buffer, value = 49
Reading 5 from buffer, value = 53
Reading 0 from buffer, value = 48
Reading   from buffer, value = 32
Reading c from buffer, value = 99
Reading h from buffer, value = 104
Reading a from buffer, value = 97
Reading r from buffer, value = 114
Reading a from buffer, value = 97
Reading c from buffer, value = 99
Reading t from buffer, value = 116
Reading e from buffer, value = 101
Reading r from buffer, value = 114
Reading   from buffer, value = 32
Reading l from buffer, value = 108
Buffer read, transferring control to producer
writing i to buffer
writing m to buffer
writing i to buffer
writing t to buffer
writing   to buffer
writing s to buffer
writing o to buffer
writing   to buffer
writing t to buffer
writing h to buffer
writing a to buffer
writing t to buffer
writing   to buffer
writing I to buffer
writing   to buffer
Buffer full transferring to consumer
Reading i from buffer, value = 105
Reading m from buffer, value = 109
Reading i from buffer, value = 105
Reading t from buffer, value = 116
Reading   from buffer, value = 32
Reading s from buffer, value = 115
Reading o from buffer, value = 111
Reading   from buffer, value = 32
Reading t from buffer, value = 116
Reading h from buffer, value = 104
Reading a from buffer, value = 97
Reading t from buffer, value = 116
Reading   from buffer, value = 32
Reading I from buffer, value = 73
Reading   from buffer, value = 32
Buffer read, transferring control to producer
writing c to buffer
writing a to buffer
writing n to buffer
writing   to buffer
writing t to buffer
writing e to buffer
writing s to buffer
writing t to buffer
writing   to buffer
writing b to buffer
writing e to buffer
writing t to buffer
writing t to buffer
writing e to buffer
writing r to buffer
Buffer full transferring to consumer
Reading c from buffer, value = 99
Reading a from buffer, value = 97
Reading n from buffer, value = 110
Reading   from buffer, value = 32
Reading t from buffer, value = 116
Reading e from buffer, value = 101
Reading s from buffer, value = 115
Reading t from buffer, value = 116
Reading   from buffer, value = 32
Reading b from buffer, value = 98
Reading e from buffer, value = 101
Reading t from buffer, value = 116
Reading t from buffer, value = 116
Reading e from buffer, value = 101
Reading r from buffer, value = 114
Buffer read, transferring control to producer
File completely read, wrote EOF and transferred control to consumer
read EOF, exiting
```

That was a long output, I know, but it shows two important thigs, one any character is printed explicitly meaning that even special characters like the newline character are printed as a literal newline. This means that we are properly catching them and that if there were any junk that made it into our buffer we would notice it very easily.

IV. Conclusion

In order to effectively prevent access to a critical section there are several factors that need to be met. The bounded waiting problem must be solved, and the mutual exclusion problem must be solved. To solve the bounded waiting problem, we limited the buffer to 15 characters and did an effective handoff between the producer and the consumer. To handle mutual exclusion, we used semaphores that indicated when the buffer was full and empty as well as a special mutex semaphore to ensure mutual exclusion between the producer and consumer accessing the critical section, in this case, the 15-character buffer. Another important thing that we had to do was to ensure that the threads were joined, meaning that the parent process that created them could not continue its execution until both of the threads were completed. This simple demonstration in my opinion show the most significance in how multiprocessing works when used in an inefficient manner. A piece of shared memory allows for only one thread to access it at a time, meaning that each thread does not have it in their stack, meaning it must be in the heap. This can slow down multiprocessing for things were the data can be copied to the stack of each thread. However, for data that cannot be copied, like in this assignment, special care must be taken in order to not access the data when no allowed so that the data is not corrupted or the wrong values are used.