Part One write up:

Posix:

For the version with an unsynchronized global variable I just moved the sum variable to be global and instead of incrementing each threads local count variable each time they found a word, I incremented the global sum variable. There wasn't much difference in execution time between this and the original, or between the ones with different amounts of threads. Except that the one thread of the global variable version was slightly slow but I think this was probably just slight randomness in execution time. I think my cpu is processing the words so fast that the increased number of threads does not make up for the cost of creating the threads. I almost doubled the amount of words in the test file to try and get a difference but it didn't really help. But as the number of threads increased the accuracy of the word count dropped off as each thread would often override the attempt of another to increment the count.

For the synchronized version I just added a mutex to control access to the global sum variable which fixed the accuracy problem of the actual word count. There was a tiny slow down in execution time as each thread needed to wait for other threads to finish using the sum var before they could. But this time the increased number of threads did increase executions speed. The increase was small though so it might just be a bit of randomness and coincidence.

Windows:

I did essentially the same thing with the windows versions of the synched and unsynched word counters. The only real difference between the windows versions and the posix versions is that the increased number of threads actually seem to make the program execute faster.

Part Two write up:

Posix:

I started by making 5 global variables, a full and empty semaphore, a lock mutex and both a cIndex and pIndex initialized to 0. Then in main I take the command line args to print a message to stdout and use their values to determine the size of the shared buffer, and the number of consumer and producer threads and their structs. This info is also used to initialize the threads structs. The producer struct contains the threadNumber a counter, the amount it will produce, a pointer to the buffer and the buffer size. The consumer thread contains a threadNumber, the amount it will consume, a pointer to the buffer and the buffer size. The amount each consumer consumes will be the number of producers times how much they produce divide by the number of consumers which may leave a remainder r so I then make a for loop that will takes the first r consumers and increments the number they will consume by one. The threads are then created and either sent off to the consumerFunc or producerFunc. In the producerFunc they go into while loop that lasts until their counter var is not less then the amount they are to produce. They then wait on the empty semaphore followed by the lock mutex. When down waiting they produce in the buffer at pIndex, print what they produced, increment their counter and pIndex mod buffer size, unlock the mutex and increment the full semaphore. The consumerFunc enters a while loop that lasts until the amount to consume is 0. It first wait on the full semaphore then the lock mutex.

It then prints that it consumed the buffer at cIndex, decrements the amount to consume and increments cIndex by 1 mod buffer size. Then it unlocks the mutex and increments the empty semaphore.

The execution times increased with the increased number of producers and consumers which correspond to increased number of threads. This makes sense because even though there are more threads doing work there is a similar increase in the amount of work to be done so it should take longer to account for the time it takes to make the extra threads.

Windows:
The windows version has the same logic with just some different functions used. Also for some reason I needed to use malloc in the windows version to initialize the size of my arrays but I didn't in the posix version.

The execution times were similar for windows in terms of relative time between them, though there seemed to be little difference between 1 consumer/producer pair and 2 consumer/producer pairs.

Part 1 output:
Windows original code
1 thread:

```
C:\Users\Alan\Desktop\415\hw5>winWordCount test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winWordCount test }
Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 0
Milliseconds      : 62
Ticks             : 621363
TotalDays         : 7.19170138888889E-07
TotalHours        : 1.72600833333333E-05
TotalMinutes      : 0.001035605
TotalSeconds      : 0.0621363
TotalMilliseconds : 62.1363
```

2 threads:

```
C:\Users\Alan\Desktop\415\hw5>winWordCount test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winWordCount test }
Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 0
Milliseconds      : 25
Ticks             : 254975
```

TotalDays : 2.95109953703704E-07
TotalHours : 7.08263888888889E-06
TotalMinutes : 0.000424958333333333
TotalSeconds : 0.0254975
TotalMilliseconds : 25.4975

4 threads:
C:\Users\Alan\Desktop\415\hw5>winWordCount test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winWordCount test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 29
Ticks : 299396
TotalDays : 3.46523148148148E-07
TotalHours : 8.31655555555556E-06
TotalMinutes : 0.000498993333333333
TotalSeconds : 0.0299396
TotalMilliseconds : 29.9396

8 threads:
C:\Users\Alan\Desktop\415\hw5>winWordCount test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winWordCount test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 26
Ticks : 269282
TotalDays : 3.11668981481481E-07
TotalHours : 7.48005555555555E-06
TotalMinutes : 0.000448803333333333
TotalSeconds : 0.0269282
TotalMilliseconds : 26.9282

Posix original code
1 thread:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.052s
user    0m0.046s
sys 0m0.005s
```

2 threads:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.072s
user    0m0.063s
sys 0m0.007s
```

4 threads:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.075s
user    0m0.074s
sys 0m0.008s
```

8 threads:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.060s
user    0m0.065s
sys 0m0.006s
```

Global not synchronized:
windows
1 thread:

```
C:\Users\Alan\Desktop\415\hw5>winGlobalUnSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalUnSync test }
Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 0
Milliseconds      : 19
Ticks             : 197935
TotalDays         : 2.29091435185185E-07
TotalHours        : 5.49819444444444E-06
TotalMinutes      : 0.000329891666666667
TotalSeconds      : 0.0197935
TotalMilliseconds : 19.7935
```

2 threads:

```
C:\Users\Alan\Desktop\415\hw5>winGlobalUnSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalUnSync test }
Days              : 0
```

Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 31
Ticks : 311823
TotalDays : 3.6090625E-07
TotalHours : 8.66175E-06
TotalMinutes : 0.000519705
TotalSeconds : 0.0311823
TotalMilliseconds : 31.1823

4 threads:
C:\Users\Alan\Desktop\415\hw5>winGlobalUnSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalUnSync test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 24
Ticks : 240368
TotalDays : 2.78203703703704E-07
TotalHours : 6.67688888888889E-06
TotalMinutes : 0.000400613333333333
TotalSeconds : 0.0240368
TotalMilliseconds : 24.0368

8 threads:
C:\Users\Alan\Desktop\415\hw5>winGlobalUnSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalUnSync test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 21
Ticks : 216750
TotalDays : 2.50868055555556E-07
TotalHours : 6.02083333333333E-06
TotalMinutes : 0.00036125
TotalSeconds : 0.021675
TotalMilliseconds : 21.675

Posix
1 thread:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
```

```
393674 words

real    0m0.074s
user    0m0.047s
sys 0m0.008s
```
2 threads:
```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
378385 words

real    0m0.065s
user    0m0.053s
sys 0m0.006s
```
4 threads:
```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
213628 words

real    0m0.068s
user    0m0.074s
sys 0m0.008s
```
8 threads:
```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
191140 words

real    0m0.064s
user    0m0.077s
sys 0m0.007s
```
Synchornized
Windows
Synchronized:
1 thread:
C:\Users\Alan\Desktop\415\hw5>winGlobalSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalSync test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 21
Ticks : 211530
TotalDays : 2.44826388888889E-07
TotalHours : 5.87583333333333E-06
TotalMinutes : 0.00035255
TotalSeconds : 0.021153
TotalMilliseconds : 21.153

2 threads:
C:\Users\Alan\Desktop\415\hw5>winGlobalSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalSync test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 21
Ticks : 213748
TotalDays : 2.47393518518519E-07
TotalHours : 5.93744444444444E-06
TotalMinutes : 0.000356246666666667
TotalSeconds : 0.0213748
TotalMilliseconds : 21.3748

4 threads:
C:\Users\Alan\Desktop\415\hw5>winGlobalSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalSync test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 22
Ticks : 223235
TotalDays : 2.58373842592593E-07
TotalHours : 6.20097222222222E-06
TotalMinutes : 0.000372058333333333
TotalSeconds : 0.0223235
TotalMilliseconds : 22.3235

8 threads:
C:\Users\Alan\Desktop\415\hw5>winGlobalSync test
192049 words
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winGlobalSync test }
Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 24
Ticks : 249772
TotalDays : 2.89087962962963E-07
TotalHours : 6.93811111111111E-06
TotalMinutes : 0.000416286666666667
TotalSeconds : 0.0249772

TotalMilliseconds : 24.9772

Posix
1 thread:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.084s
user    0m0.053s
sys 0m0.008s
```

2 threads:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.082s
user    0m0.054s
sys 0m0.007s
```

4 threads:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.076s
user    0m0.048s
sys 0m0.007s
```

8 threads:

```
ALEXANDERs-MacBook-Air:hw5 alexanderryner$ time ./a.out ~/
Desktop/greatExpectations.rtf
393674 words

real    0m0.072s
user    0m0.056s
sys 0m0.007s
```

Part 2 output:
Posix

```
akee@akee-1005HA:~/school/spring2014/415/hw5$ time ./a.out 4 1 1
1000
Buffer size = 4, number of producer threads = 1, number of
consumer threads = 1, and each producer produces 1000
Producer 1 produced: 1000000
Producer 1 produced: 1000001
Producer 1 produced: 1000002
Producer 1 produced: 1000003
```

```
Consumer 1 consumed: 1000000
Consumer 1 consumed: 1000001
.
.
.
Consumer 1 consumed: 1000997
Consumer 1 consumed: 1000998
Producer 1 produced: 1000999
Consumer 1 consumed: 1000999
****All jobs finished***

real    0m0.074s
user    0m0.008s
sys     0m0.060s
akee@akee-1005HA:~/school/spring2014/415/hw5$ time ./a.out 4 2 2
1000
Buffer size = 4, number of producer threads = 2, number of
consumer threads = 2, and each producer produces 1000
Producer 1 produced: 1000000
Producer 1 produced: 1000001
Producer 1 produced: 1000002
Consumer 1 consumed: 1000000
.
.
.
Producer 2 produced: 2000998
Producer 2 produced: 2000999
Consumer 1 consumed: 2000998
Consumer 1 consumed: 2000999
****All jobs finished***

real    0m0.175s
user    0m0.036s
sys     0m0.140s
akee@akee-1005HA:~/school/spring2014/415/hw5$ time ./a.out 4 4 4
1000
Buffer size = 4, number of producer threads = 4, number of
consumer threads = 4, and each producer produces 1000
Producer 1 produced: 1000000
Producer 3 produced: 3000000
Consumer 1 consumed: 1000000
Producer 3 produced: 3000001
Consumer 1 consumed: 3000000
.
.
.
Producer 4 produced: 4000998
```

```
Producer 4 produced: 4000999
Consumer 1 consumed: 4000998
Consumer 1 consumed: 4000999
****All jobs finished***

real    0m0.451s
user    0m0.112s
sys     0m0.460s
```

Windows:
Buffer size = 4, number of producer threads = 1, number of consumer threads = 1, and each producer produces 1000
Producer 1 produced: 1000000
Consumer 1 consumed: 1000000
Producer 1 produced: 1000001
.
.
.
Producer 1 produced: 1000998
Consumer 1 consumed: 1000998
Producer 1 produced: 1000999
Consumer 1 consumed: 1000999
All jobs finished
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winProducerConsumer 4 1 1 1000 }

Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 83
Ticks : 830577
TotalDays : 9.61315972222222E-07
TotalHours : 2.30715833333333E-05
TotalMinutes : 0.001384295
TotalSeconds : 0.0830577
TotalMilliseconds : 83.0577

Buffer size = 4, number of producer threads = 2, number of consumer threads = 2, and each producer produces 1000
Producer 1 produced: 1000000
Producer 2 produced: 2000000
Producer 2 produced: 2000001

Producer 2 produced: 2000002
.
.
.
Consumer 2 consumed: 2000999
Consumer 2 consumed: 1000997
Consumer 2 consumed: 1000998
Consumer 2 consumed: 1000999
All jobs finished
PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winProducerConsumer 4 2 2 1000 }

Days : 0
Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 84
Ticks : 840280
TotalDays : 9.72546296296296E-07
TotalHours : 2.33411111111111E-05
TotalMinutes : 0.00140046666666667
TotalSeconds : 0.084028
TotalMilliseconds : 84.028

Buffer size = 4, number of producer threads = 4, number of consumer threads = 4, and each producer produces 1000
Producer 1 produced: 1000000
Producer 2 produced: 2000000
Producer 2 produced: 2000001
Producer 2 produced: 2000002
.
.
.
Producer 4 produced: 4000999
Consumer 4 consumed: 3000998
Consumer 4 consumed: 4000998
Consumer 4 consumed: 3000999
Consumer 4 consumed: 4000999
All jobs finished

PS C:\Users\Alan\Desktop\415\hw5> Measure-Command { .\winProducerConsumer 4 4 4 1000 }

Days : 0

Hours : 0
Minutes : 0
Seconds : 0
Milliseconds : 160
Ticks : 1604458
TotalDays : 1.85701157407407E-06
TotalHours : 4.45682777777778E-05
TotalMinutes : 0.00267409666666667
TotalSeconds : 0.1604458
TotalMilliseconds : 160.4458