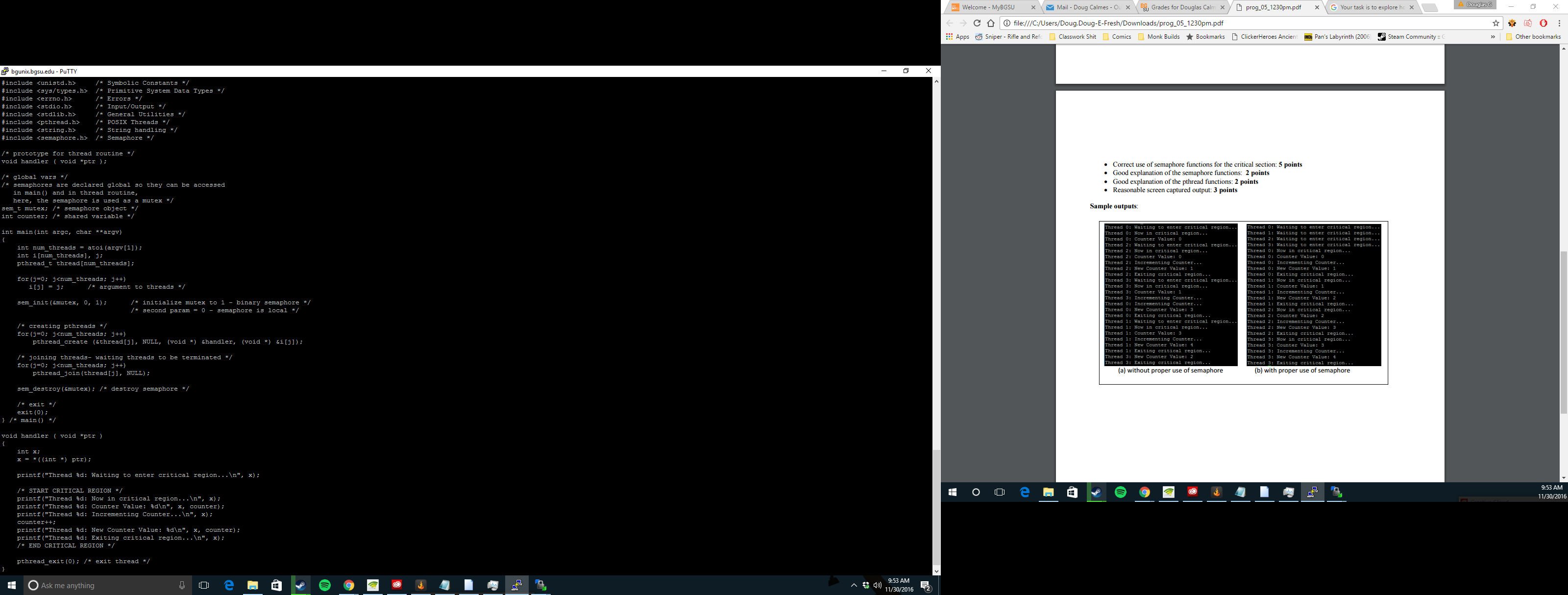
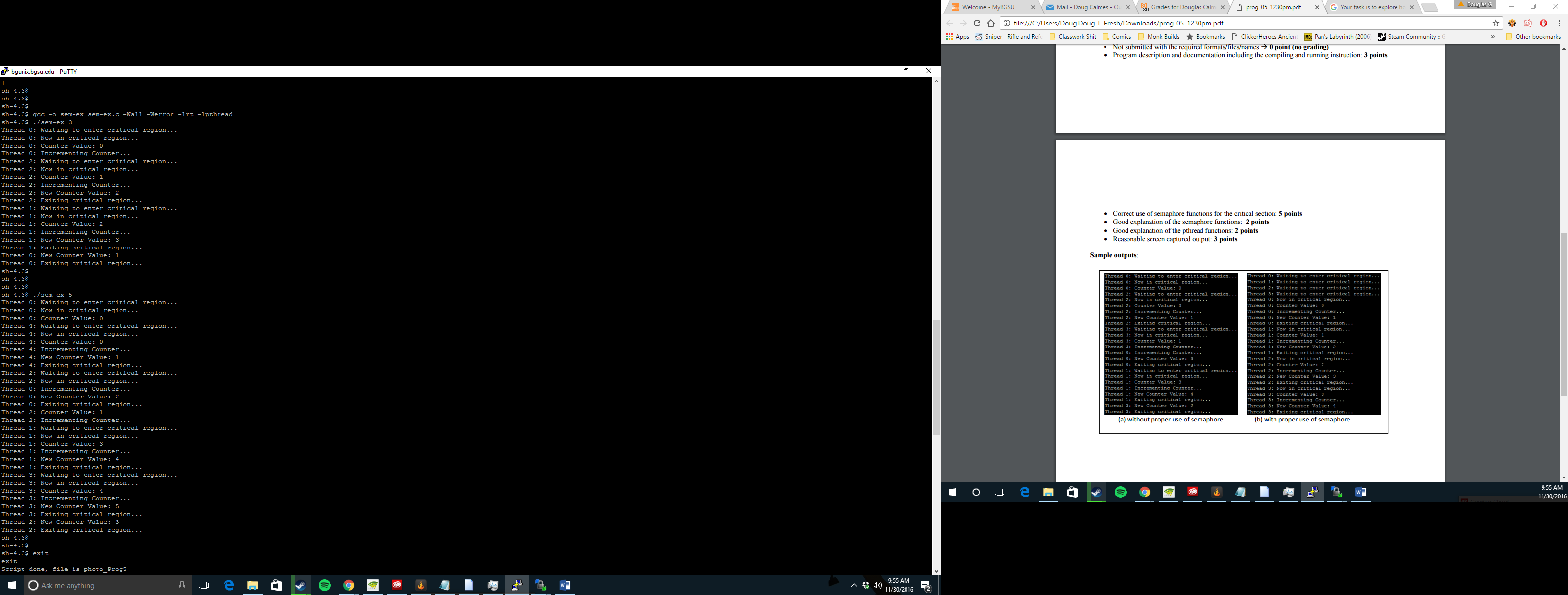
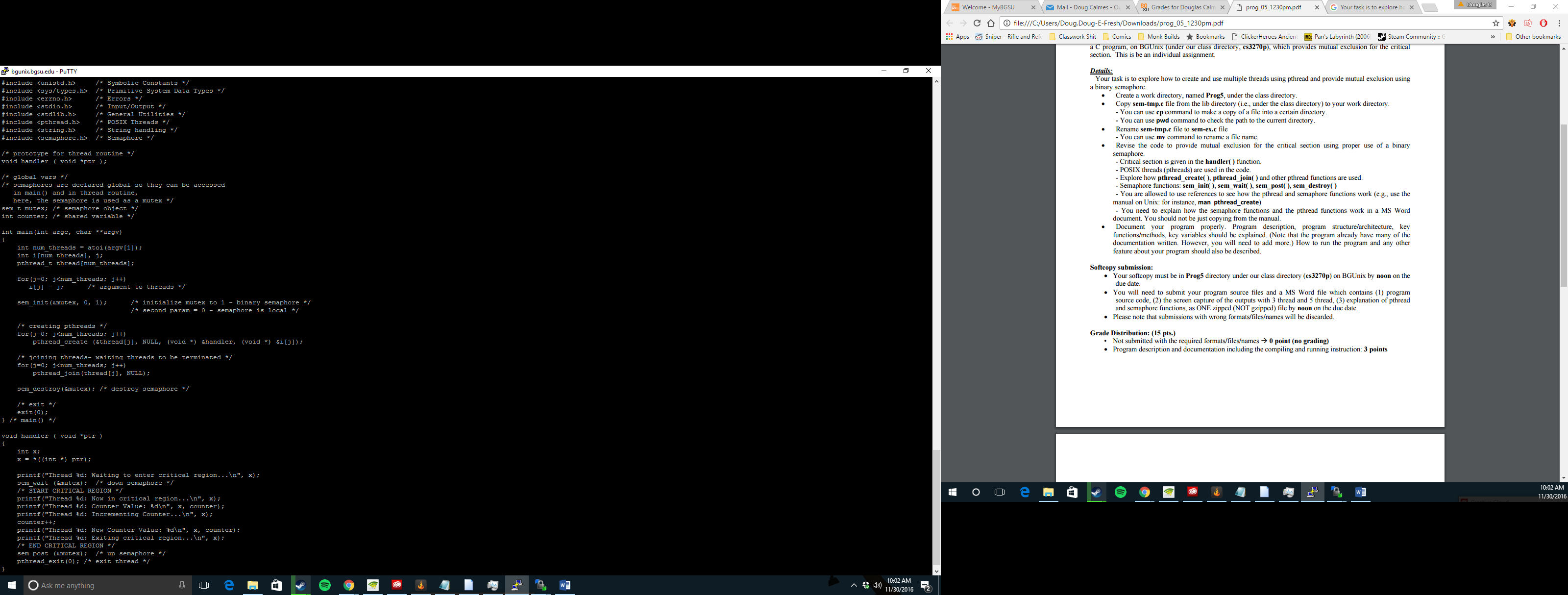
Program 5 Source Code without proper use of semaphore



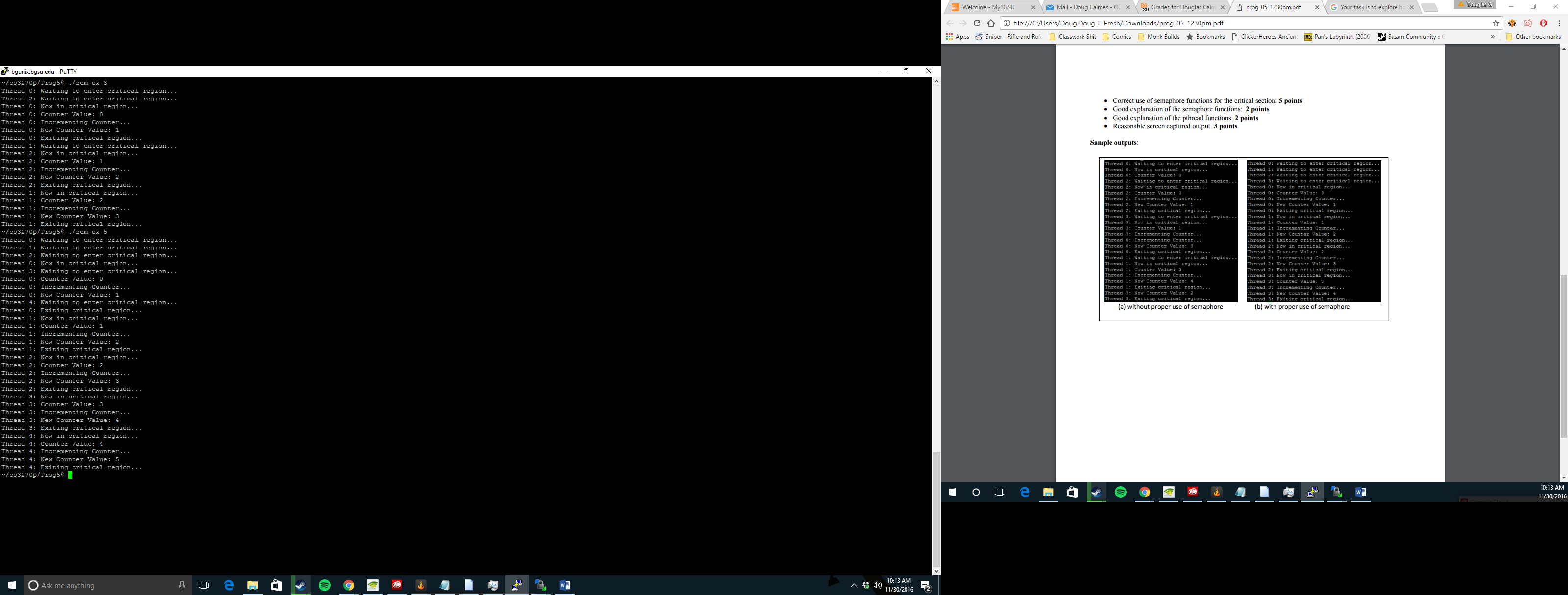
Outputs without proper use of semaphore



Program 5 Source Code with proper use of semaphore



Outputs with proper use of semaphore



Code Explanation

pthread\_create ( )

This is used to create a new thread within a process. The attributes of these threads are identified by attr. Default attributes will be used if attr is NULL. This changes if the attributes are modified later, speaking of attr. The attributes of the thread are not affected by this. Pthread\_create ( ) will store the ID of the created thread in the location referenced by thread upon a successful completion.

The thread is created executing start routine with arg as the only argument. It should be noted that if the start routine returns, the effect carrys out as if there was an imbedded call to pthread\_exit ( ) using the return value of start routine as the exit status.

pthread\_join ( )

This function suspends processing of the calling thread until the target thread has completed. The thread must be a member of the current process and must not be a thread in which is detached. While there are two or more threading waiting for the same thread to complete, they will all be suspended until the thread has terminated. It is them when one thread will return successfully and the others will return with an error. This function will not block processing of the calling thread if the target thread has been terminated.

If the funtion call returns successfully, the value will be passed to pthread\_exit ( ).

If the function calling thread is cancelled, then the target thread will remain joinable. It is important to note that the calling thread may set up a cancellation cleanup handler on the thread.

sem\_init ( )

This function is used to initialize the unnamed semaphore denoted to by sem. The semaphore can be used in subsequent sem functions and will remain usable until the semaphore is destroyed.

sem\_wait ( )

This function locks in the semaphore referenced by sem. It does this by performing a semaphore lock operation on that semaphore. If the semaphore is 0, then the calling thread will not return from the call to this function until it locks or the call is interrupted. With a successful return, the semaphore will remain locked until sem\_post is executed successfully.

sem\_post ( )

This function unlocks the semaphore referenced by sem. It does this by performing a semaphore unlock operation. If the semaphore value is positive, then no threads were blocked waiting for the semaphore to become unlocked. This caused the semaphore to be incremented. If the value of the semaphore is 0 one of the blocked threads will return successfully.

sem\_destroy ( )

This function is used to destroy the unnamed semaphore indicated by sem. It can only destory semaphores that are created by sem\_init. It is safe to destroy any initialized semaphore that has no blocked threads.