Carter Nettesheim
ECE5780
Chapter 18 problems

1) There could be access problems if multiple tasks try to access the resource at the same time
Incorrect data could be at the resource if accessed by multiple tasks

2) Memory mapping can share a memory space between multiple tasks. For example, a mouse position needs to share it's position to multiple applications. This memory can be mapped to a certain memory location to allow multiple tasks to use it. This is a fast efficent way to share a resource, however, some unsafe properties come with it.

3) A critical section is a portion of code that has to be executed with no interruptions. This means other threads or interrupts cannot interfere
A contending critical section is a critical section being used by multple interrupts or tasks at the same time. This leads to weird results.

4) Procure operation "reserves" the resource
Vacate operation "releases" or "frees up" the resource

5) A semaphore is essentially like an OPEN sign at a restaurant. The semaphore allows the resource to be used if the sign is turned to "open," then allows a certain amount of cusotmers in before it closes. When customers leave, more customers are allowed in.

6) The first figure shows a semaphore with multiple slot options available. This means multiple users can use the resource at the same time. The second figure shows a mutex implementation, where the sememphore only allows 1 user to use the resource at one time.

8) A named semaphore has a dedicated task, such as a binary semaphore. This semaphore allows only 1 task to use a resource at a time. Unnamed semaphores are generic and can vary in what task it has.

9) A mutex has lower overhead, supports priority inheritence and only allows one task to use a resource at a time. Meaning no threat of accessing the same memory at the same time.