Mutal Exculsion

Data Races

- A problem that can occur when two or more threads are attempting to access the same memory location, and one of those threads are writing to that memory locations.
- Data races sometimes may not be detected since when a concurrent program runs it behaves differently each time. This makes it very hard to detect data races.

Mutal Exculsion

- Crictical Section of a concurrent program is a code segment that accesses a shared resource. This section must be protected.
- Need to develop defence mechanisms in order to prevent this from happening. In programs you will use a mutex (lock) to prevent data races from happening. One thread/process can have access to the lock at time as they are using the shared recourse.
- Gaining access to the lock is an atomic action,it is a single action that takes places relative to the other threads. Cannot be intreuptted by concurrent threads.
- If a thread attmepts to gain access to the lock, while another thread has the lock, will be put into the blocked state, and can wait until it is avaiable.
- Threads can get blocked at any time (OS can stop the thread in the middle of exectution), so you will want to keep the protected sections of code short to prevent other threads from being blocked for very long periods of time.