This is a list of questions for you to evaluate if you satisfy the prerequisite requirements of [CS 571](https://tddg.github.io/cs571-spring22/). You are expected to know (or at least be able to quickly find) the answers to most of these questions if you plan to take the course:

* What is an OS kernel? Is an OS kernel running all the time?
* What is a race condition? How to prevent race conditions?
* What is a deadlock? How to prevent/avoid deadlocks?
* What is the difference between processes and threads?
* How does the Linux kernel implement processes and threads?
* What is a TLB and what is a page table?
* What is the page table structure used by the x86 architecture?
* Can you describe the process of memory address translation?
* What is external fragmentation and internal fragmentation?
* Does each memory access go through the OS kernel?
* What is LRU and how is it implemented in the Linux kernel?
* What is the CPU scheduling algorithm used in the Linux kernel?
* What is a page fault? Can we always expect reduction of page faults by increasing the number of virtual pages in the virtual memory space?
* What is the difference between a virtual machine (VM) and a container?
* What is the difference between paravirtualization and full-virtualization?
* What are system calls? How are system calls supported in virtual machines?
* How do file systems manage disk blocks?
* What is RAID (redundant array of inexpensive disks), how does RAID work, and what are commonly-used RAID configurations?
* How does a flash-based SSD (solid state drive) work? How is it different from a mechanical HDD (hard disk drive)?
* What is a page cache and why is it useful?
* What is RPC (remote procedure call) and how does it work?
* What is MapReduce and what are the problems it solves?

Contact me ([yuecheng@gmu.edu](mailto:yuecheng@gmu.edu)) if you have any other cool prerequisite questions.

\* Acknowledgment: Some questions taken and adapted from CS 523 at Illinois by Tianyin Xu.