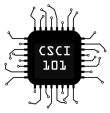
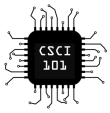




Homework #4 Chapter 6 (13 points)
Due: Friday, October 18th, at start of class
(1 point each unless marked)

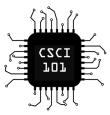
1) List the four main tasks the operating system handles.

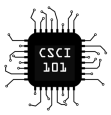
2) Define superusers and give two examples of why we need superusers.

3) My single CPU system appears to execute multiple programs at exactly the same time. How does the operating system do that with just one CPU?

4) We discussed the process state graph in class. Discuss each transition that leaves the RUNNING state; in each case, name the state the transition goes to and briefly explain how the transition happens. In addition, discuss each transition that comes into the RUNNING state. (2 points)

5) A CPU-intensive process and an I/O-intensive process are both running on a heavily loaded single processor machine. What state are they each likely to spend a majority of their time in (outside of the running state), and why? (2 points)





6) Luke notices his system response time is really slow. To improve performance, he decides to reduce the system's time slice (from 170 ms to 1 ms). Is this a good idea? Assume the time to switch from one process to another takes 1 ms. Justify your response.

7) Assume that an individual process in some system spends about 75% of its time waiting for I/O operations to be completed. What percentage of time is the processor doing useful work (called processor utilization) if there are four processes loaded into memory? How many processes should we keep in memory if we want processor utilization to be at least 85%? Justify your answer. (2 points)

8) Suppose my process is BLOCKED, waiting on a file that your process is using, and your process is BLOCKED, waiting on a file that I am using. What is this situation called, and how can it be resolved?

9) Define what an interrupt is and provide one example when an interrupt might occur on your system.

10) Consider a Python program that is stored in your computer's memory. In the context of this program, define the difference between the program's first virtual address and the program's first physical address.