CSC 139 OPERATING SYSTEMS SAC STATE
HM **MIDTERM 2**

Rules for Midterm, 150 points: Work alone and independently; open books, open internet, open notes. Place your name onto your midterm. Each question is worth 10 points. Answer via single words or short phrases where feasible, minimize full sentences.

Alec Resha
MWF 9:00-9:50
11/17/2021

1. Describe >= 4 ***purposes or goals*** of a well functioning OS.
- Prevent Deadlock
- Create Processes
- Delte processes
- Manage process execution

2. Explain the OS term ***livelock***. What causes it? What is the result?
- Multiple processes each claim some of the resources neede by both of them
- CPU switches between the processes rapidly, but both of them are awaiting resources
- Neither process is blocked, they just can't run

3. Explain high-level the key points of ***thread***. Outline what is a "hyper-thread".
- Hyper-thread: More than 1 thread per core
- Threads: smallest unit of programmed instructions, subset of a process

4. Processes run concurrently, ***sharing resources, e.g. memory***. How can the OS prevent a process from accessing data of other processes?
- Can mark the resources as protected while they are in use by another process, then freeing the resources one the process is done

5. Briefly sketch ***Demand Paged Virtual Memory Management***. Explain "demand", "paged", "virtual", "page-size", "page alignment". Contrast briefly with physical memory.
- Demand: If page isn't in memory, a miss occurs and the page is then loaded
- Virtual: Not physical memory, managed by OS
- Page-size: Number of references to memory per page
- Page-alignment: All pages equal in size, first byte is divisible by page size
- 12 bits of 0 on low end

6. Argue whether VMM is an OS technology ***necessary today***, given that memory is cheap.
- It is still necessary, memory is cheap but there are devices that are small so they are forced to use less memory (raspberry pi, etc)

7. Explain in brief phrases: ***program***, ***process, thread, hyperthread, orphan.***
- ***Program:*** Code to be executed
- ***Process:*** Program loaded into runnable form
- ***Thread:*** Breakdown of a process into smaller pieces to run more efficiently
- ***Hyperthread:*** Run multiple threads on each core
- ***Orphan:*** Still running child process of a now finished or terminated process

8. Outline ***relative speeds*** of *storage devices* on a Computer System from fast to slow.
- From Fastest to Slowest:
    - Cache
    - Main Memory
    - SSD
    - HDD

- Tape Drive

9. Devices such as keyboard, printer, display etc. are part of a computer system. How do they receive, share, and communicate information with/from the OS?
- Communication handled by a device driver
- Driver communicated directly with I/O locations
- Can ask the device for changes in status or device will send signals to driver to be sent to OS

10. List and explain in a word or short phrase various (>=5) physical and program-related parameters of ***mass-storage***. What are some mass storage components of a computer?
- Sustainable transfer speed
- Quickly find locations in data
- High capacity
- Reliable storage
- Mass storage components: HDD, SSD (Auxilliary storage)

11. Argue, whether it is ***fair*** that some process **p** is not currently running, though **p** has all resources (except the CPU) needed to execute.
- Yes, if p is switched in or out then it takes more resources and time to repeatedly allocate/deallocate resources instead of waiting for a process to finish
- If p has an error then it hogs the resources, but a large majority of the time process use their resources and finish, freeing up the resources for other processes

12. List process-related responsibilities of a ***multi-user, time-sharing OS***.
- Create processes
- Delete processes
- Sync processes
- Manage processes
- Prevent deadlock

13. Outline high-level the major engineering challenge for designing and implementing an “ideal VMM algorithm” in an Operating System. Is it realistic, absurd, feasible?
- Preventing deadlock
- Checking for live-locked processes then freeing up resources
- Implement a useful priority system so processes don’t get left behind (RR Priority vs FCFS, etc)
- Mostly feasible, livelock can be hard to detect since the processes are still running even though they aren’t making progress

14. What does ***PCB*** stand for? What is essential about a PCB? What is its use? Which information is contained in a PCB?
- Process Control Block; any piece of information relative to one process

15. An Operating System Resource Allocation Graph G shows processes and needed resources. Which graph condition proves that deadlock cannot occur? Which proves that deadlock is possible? Which condition is proof of deadlock?
- If a resource of process a is process b and vide versa and they each have resources allocated that the other needs, deadlock is happening
- If 2 processes require eachother then deadlock is possible
- If no processes need eachother, deadlock cannot occur