CSE3502, Operating Systems
Homework 1

1. What are the three themes of an operating system? Explain each of the function briefly. (5 point)

Virtualization – This process takes a physical resource like processors or memory and transforms it into a virtual form of itself.
Concurrency – This is the process of handling multiple things at once and allocating processing time to each.
Persistence – This involves the scenario where hardware and software data needs to be stored persistently.

2. What are the differences of orphan processes and zombie processes? (5 point)

Orphan processes are when a computer process has successfully finished or terminated.
A zombie process is a process in which has finished execution but still has an entry in the process table (i.e. a terminated state).

3. What are the differences between a monolithic kernel and a microkernel? (5 point)

The two are largely different due to the Monolithic kernel compromising all of the OS space and the microkernel only operating in a the most minimal amount of software space needed.

4. What is system call used for? (5 point)

A system call provides an interface between processes and Operating System.

5. What is a process? What are the two essential parts of a process? How is a process different from a program? (5 point)

No formal definition for it, it depends on the scenario.

- *It can be an instance of a program running*
- *An abstraction that supports the running program*
- *An execution stream of the process state*
- *Sequential stream of execution in its own address space*

Two parts: Sequential execution of instruction, Process state
A program is executable static code, a process is a dynamic instantiation of code and data and such.

6. Given the five-state process model, explain how does a process transit among these states and on what events? (5 point)
Running -> Ready: A thread was descheduled
Running -> Waiting: A thread was blocked or waiting for I/O
Waiting -> Ready: The event the thread was waiting for happened
Ready -> Running: A thread was scheduled
Running -> Terminated: A thread exited or hit a fatal exception

7. What are the differences of threads and processes? (5 point)

In a broader context, a thread is a lightweight process or an instance of a program running and a process is the steps it takes to create an instance.

8. Discuss the advantages and disadvantages of user-level threads and kernel-level threads. (15 point)

User-Thread:

Advantages:

- *User level threads can run on any OS*
- *Scheduling can be application specific*

Disadvantages:

- *Most system calls are blocking*
- *Cannot take advantage of multiprocessing*

Kernel-Thread:

Advantages:

- *Capable of scheduling multiple threads on one processor*
- *Kernel routines can be multithreaded*

Disadvantages:

- *Slower to create*

9. What is the deadlock? (5 point)

Two or more processes waiting on a resource.

10. What are the commonalities and differences between semaphore and mutex? (10 point)

Differences:

- *Semaphore is a signaling mechanism while Mutex is a locking mechanism*
- *Semaphore value can be changed by any process while Mutex can only be released by the thread that originally acquired it*

Commonalities:

- *Operation is held according to sleep and wake approaches*

11. List different ways to avoid race conditions. (5 point)

No two processes are simultaneously in the critical region
No assumptions made about speeds or number of cpu's
No process running outside its critical region it may block another process running in the critical region
No process must wait forever to enter its critical region

12. What are the advantages and disadvantages of busy-waiting and sleep-and-wake approaches for mutual exclusion? (10 point)

Busy-waiting is very efficient by not blocking and instead it constantly polls the lock for availability. Used if you expect a lock to be released quickly.

13. Discuss the goals of CPU scheduling on different computer systems, e.g., batch systems, interactive systems and real-time systems. (5 point)

Batch Systems:

- *Maximize jobs per hour*
- *Minimizes time between submission and termination*
- *Keeps CPU busy*

Interactive Systems:

- *Quick to respond to clients' requests*
- *Meets user expectation*

Real-Time Systems:

- *Meets deadlines to avoid losing data*
- *Highly predictable*

14. Assume that the following processes are to be executed on a uniprocessor system.
Based on their arrival time and CPU burst, calculate the average turnaround time and average response time under the following scheduling policies: (15 point)
a. FCFS
b. Round Robin (quantum = 4 and 6)
c. Shortest Job First (preemptive and non-preemptive)

Process	**Arrival Time**	**CPU burst**
P1	**0**	**12**
P2	**0**	**3**
P3	**2**	**7**
P4	**3**	**5**

-15

Compare the performance of above policies.
(PS: for P1 and P2 in FCFS, we might have two cases: P1 scheduled first; P2 scheduled first)

Please show your calculation steps and fill in the following table:

FCFS	P1 first	P2 first
Avg Turnaround	12	15
Avg response		

RR	Q=4	Q=6
Avg Turnaround		
Avg response		

SJF	Preemptive	Non-preemptive

Avg Turnaround		
Avg response		