TEST QUESTIONS (25->75 pts)

1. Order these options to identify causes and effects of multiple processes and threads.

1. When data is shared by multiple concurrent processes, race conditions can occur
2. From low-level hardware solutions to higher-level software tools, several tools address the critical-section problem
3. There are several challenges to the design of applications that are free of race conditions, including liveness hazards such as deadlocks

A. 1-3-2
B. 3-2-1
C. 2-3-1
D. 1-2-3

2. Which of these is not a classic synchronization problem?

A. Bounded-Buffer
B. Readers and Writers
C. Dining-Philosophers
D. Context-switch

3. On which conditions hold simultaneously can a deadlock arise?

1. Mutual exclusion
2. Hold and wait
3. No preemption
4. Circular wait

A. 1-2-3
B. 2-3
C. 1-4
D. All

4. Which definition(s) is/are not correct?

1.Monitor is a low-level abstraction that provides a convenient and effective mechanism for process synchronization
2.Semaphore is a synchronization tool that provides more sophisticated ways than Mutex locks for processes to synchronize their activities.
3.Mutex (Mutual Exclusion) lock is the hardest way to solve critical section problem

A. None
B. 1-2
C. 1-3
D. 3

5. Given semaphores, which methods should a developer call?
1. wait()
2. acquire()
3. signal()
4. release()
A. 1-2
B. 1-3
C. 2-4
D. 2-3

6. Which methods should a developer call when considering mutex (mutual exclusion) locks?
1. wait()
2. acquire()
3. signal()
4. release()
A. 1-2
B. 1-3
C. 2-4
D. 2-3

7. What is the correct order of synchronizing tools from low level to high level?
A. mutex-semaphore-monitor
B. semaphore-monitor-mutex
C. monitor-mutex-semaphore
D. mutex-monitor-semaphore

8. What are the goals of a typical operating system?
1.Execute user programs and make solving user problems easier
2.Make the computer system convenient to use
3.Use the computer hardware in an efficient manner
A. None
B. All
C.1-2
D. 2-3
9. Which options refer to the components of a computer system?
1.Users (People, machines, other computers)
2.Application programs
3.Operating system
4.Hardware
A. 2-4
B. All
C. 1-2-3
D. 2-3-4

10. Select the activities for which the operating system is responsible in the context of process management.

1.Creating and deleting both user and system processes
2.Suspending and resuming processes
3.Providing mechanisms for process synchronization
4.Providing mechanisms for process communication
A.1-2-3
B.2-3-4
C.1-3-4
D.All

11. Which description is not correct in the context of an operating system?

A. Kernel is the one program running at all times on the computer
B. Bootstrap program is a simple code to initialize the system, load the kernel
C. A trap or exception is a hardware-generated interrupt caused either by an error or a user request
D. An operating system is interrupt driven

12. Which option is not correct given the definitions of process states?

A. running: Instructions are being executed
B. new: The process is being created
C. terminated: The process has finished execution
D. ready: The process is waiting for some event to occur

13. Which explanation(s) is/are true?

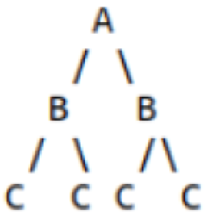
1. Process is a program in execution; process execution must progress in sequential fashion
2. Program counter is stored in Process Control Block (PCB)
3. Stack includes temporary data in a process
4.Heap contains memory statically allocated during run time
A. All
B. 1-2-3
C. 3-4
D. 1-2

14. Which statement shows the process identifier in the C programming language?

A. struct mm_struct *mm;
B. struct task_struct *parent;
C. pid t_pid;
D. struct list_head children;

15. Select the critical description(s) for the context-switch.
1. Context-switch time is overhead; the system does no useful work while switching
2. The more complex the OS and the PCB, the longer the context switch
3. When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a context switch
A. 1
B. 2-3
C. 3
D. All

16. Which system call(s) is/are essential for the process creation?
1. fork
2. exec
3. wait
A. 1
B. 1-2
C. 2-3
D. All

17. 

Which system call should be used to generate this tree representation regarding the process creation? And how many times should this call be used?
A. exec, 7
B. exec, 3
C. fork, 3
D. fork, 7

18. What are the design issues associated with concurrency?
1.communication among processes
2.sharing of and competing for resources such as memory and files
3.synchronization of the activities of multiple processes
A. None
B. 1-2
C. 2-3
D. All

19. Which explanation(s) is/are not correct considering Interprocess Communication (IPC)?

1.Shared memory is a model of IPC

2. IPC is responsible for transferring data/information between address spaces

3. Bounded-buffer assumes that there is an unlimited buffer size

A. All

B. 1-2

C. 3

D. 2-3

20. Which one is the right explanation for the pipes?

A.Ordinary pipes can be accessed from outside the process that created it.

B.Named pipes cannot be accessed without a parent-child relationship.

C.Communication is bidirectional in named pipes

D.Ordinary pipes allow communication in standard producer-consumer style as a bidirectional communication

21. Pick up the multithreading examples?

1.Checking the spelling of words in a document while writing the document

2.Counting the population of the whole country

3.Encoding of an mp3 audio file

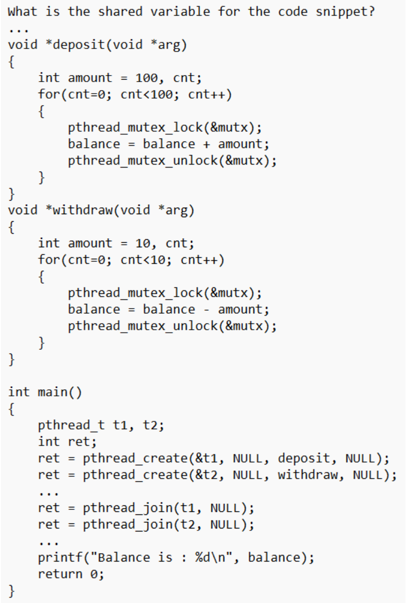
A. All

B. 1-2

C. 2-3

D. 1-3

22.



```
What is the shared variable for the code snippet?
...
void *deposit(void *arg)
{
    int amount = 100, cnt;
    for(cnt=0; cnt<100; cnt++)
    {
        pthread_mutex_lock(&mutx);
        balance = balance + amount;
        pthread_mutex_unlock(&mutx);
    }
}
void *withdraw(void *arg)
{
    int amount = 10, cnt;
    for(cnt=0; cnt<10; cnt++)
    {
        pthread_mutex_lock(&mutx);
        balance = balance - amount;
        pthread_mutex_unlock(&mutx);
    }
}

int main()
{
    pthread_t t1, t2;
    int ret;
    ret = pthread_create(&t1, NULL, deposit, NULL);
    ret = pthread_create(&t2, NULL, withdraw, NULL);
    ...
    ret = pthread_join(t1, NULL);
    ret = pthread_join(t2, NULL);
    ...
    printf("Balance is : %d\n", balance);
    return 0;
}
```

A. amount

B. cnt

C. balance

D. ret

23. Consider the multi-threaded scenarios below, select the option where there exists a race condition.

A. Matrix Multiplication. Each thread is responsible for different portions of the matrices.

B.In a text editor, one thread is responsible for updating the text and the other one is responsible for saving the text to file.

C.Finding the maximum integer in a given array. Each thread is responsible for different portion of the array.

D.Copying multiple files from a folder to another. Each thread is responsible for different files.

24. Which problem(s) is/are directly related to the circular wait?

1. Bounded-Buffer
2. Readers and Writers
3. Dining-Philosophers

A. 1

B. 1-2

C. 3

D. All

25. Regarding the multithreading models, which option is not correct?

A. One-to-One, number of threads per process sometimes restricted due to overhead

B. Many-to-One, Multiple threads may not run in parallel on multicore system because only one may be in kernel at a time

C. Many-to-Many, requires coordination between user and kernel level thread managers

D. Linux uses Many-to-One multithreading models