

QUESTION 1 (15 points):

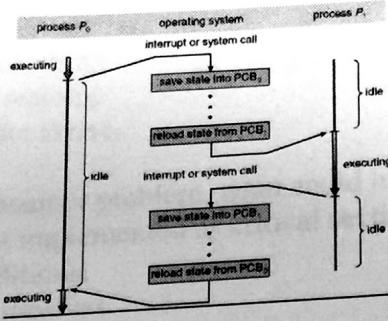
Select the best correct answer from the following:

1. The purpose of an operating system includes:
 - a. To provide an environment to execute user programs efficiently.
 - b. To allocate and manage machine resources as needed. ←
 - c. Supervision of the execution of programs.
 - d. All of the above.
2. Which of the following is an example of resources an operating system will typically manage:
 - a. CPU.
 - b. Memory.
 - c. Storage.
 - d. All the above.
3. Which instruction of the following should operate in kernel mode, and not user mode:
 - a. Reading the system clock.
 - b. Addition.
 - c. Turn off interrupts.
 - d. None of the above.
4. Operating system activities related to memory management includes:
 - a. Deciding where to load processes into memory.
 - b. Deciding whether a user program requested memory variables efficiently.
 - c. Deciding whether appropriate data structures were used in user programs.
 - d. All the above.
5. Some of the major activities an operating system does with regards to secondary storage includes:
 - a. Storing data in specific locations on the media.
 - b. Scheduling disk appropriately.
 - c. Understanding how to retrieve data when needed.
 - d. All the above.
6. Which of the following states indicates that a process is waiting to be assigned to a processor:
 - a. Running.
 - b. Waiting.
 - c. Ready.
 - d. New.

7. The data structure responsible for tracking information associated with processes is the:

- a. Process symbol table.
- b. Process control block.
- c. Process tracking table.
- d. Process identification block.

8. The following diagram is an illustration of:



- a. Process hibernation.
- b. Context switching.
- c. Process priority setting.
- d. Process engaging in a sleep state.

9. In operating systems, a time slice refers to the amount of time:

- a. A process is allowed to exist.
- b. A process is given access to the CPU at a time.
- c. A disk is allowed to fetch data.
- d. A process is allowed to sleep.

10. An I/O bound process refers to a process that:

- a. Requests at least a single I/O request.
- b. Spends more time using the CPU than I/O requests.
- c. Spends more time doing I/O requests than CPU usage.
- d. Requests the most I/O requests amongst its peers.

11. In your opinion, which of the following inter-process communication mechanism would you initially consider as being the most efficient in terms of time:

- a. Remote procedure calls.
- b. Shared memory.
- c. Client server communication using sockets.
- d. Web services.

RPC
RMI
sockets
pipe

12. Which of the following refers to a process that is blocked until a message is received by the corresponding process:

- a. Blocking send.
- b. Blocking receive.
- c. ~~Asynchronous~~ send.
- d. ~~Asynchronous~~ receive.

13. If a process P_i is executing in its critical section, no other processes can be executing in their own critical sections. This concept is referred to as:

- a. Progress.
- b. Relative blocking.
- c. Bounded waiting.
- d. None of the above.

mutual exclusion

14. In a producer-consumer problem, what could happen if the wait and notify operations are not implemented as critical sections?

- a. Race conditions.
- b. Blocking receives.
- c. ~~Deadlocks.~~
- d. All the above.

-1

15. The following could be used as strategies for scheduling processes:

- a. Shortest task first.
- b. Highest priority task first.
- c. Random task first.
- d. First come first serve.
- e. All the above.

QUESTION 2 (20 points):

Answer true or false, circle the correct answer.

1. Certain machine instructions can only be executed in Kernel mode, and not in user mode.	TRUE - FALSE
2. System calls allow user-level processes to request services from the operating system.	TRUE - FALSE
3. A command line interpreter reads commands from the user and executes them.	TRUE - FALSE
4. All operating systems support the same set of systems programs.	TRUE - FALSE
5. An operating system kernel that contains the most basic applications and services is called a monolithic kernel. <i>micro</i>	TRUE - FALSE
6. A process scheduling strategy that always gives preference to shorter processes can cause longer processes to starve.	TRUE - FALSE
7. Some operating systems do not have a long-term process scheduler.	TRUE - FALSE
8. Some operating systems could only allow one user process to run at a time.	TRUE - FALSE
9. If a task executing on a single processor is divided amongst k processors, the task will complete k times faster. <i>Amdahl's Law</i>	TRUE - FALSE
10. Race conditions are more serious in non-preemptive operating systems.	TRUE - FALSE
11. Having more than one CPU can guarantee true parallelism.	TRUE - FALSE
12. Any algorithm can be parallelizable.	TRUE - FALSE

13. Parallel matrix multiplication could best be seen as an example of data parallelism.

TRUE - FALSE

14. The best known process scheduling strategy is the shortest-task-first strategy.

TRUE - FALSE

15. Every user thread must map to a dedicated kernel thread.

TRUE - FALSE

MM, M¹, 11

QUESTION 3 (15 points):

Short answer questions (do not write beyond the provided space):

- Give a reason why system cache is needed:

Not all storage operates at the same speed. This is due to the cost of very fast storage options.
System cache is needed to help data retrieval speeds match CPU consumption speed.

- Some of the major activities an operating system does as relates to process management includes the creation and deletion of processes. State three other activities an operating system does related to process management.

a. ... schedules the order of process execution

b. ... performs accounting to keep track of CPU usage per process (as well as I/O management for the process)

c. ... allows for inter-process communication through OS API like pipes

- An application has 1000 assembly equivalent lines of code. 300 lines of code must execute serially, and 700 are parallelizable. The application will execute on 3 independent processors. What is the maximum speedup it can experience according to Amdahl's law? Reference made to the following:

$$\frac{1}{S + \frac{(1-S)}{N}}$$

S is the percentage of serial lines of code $\rightarrow S = \frac{300}{1000} = 0,3$

N is the number of processors $\rightarrow N=3$

$$\text{speedup} = \frac{1}{\frac{3}{10} + \frac{1-0,3}{3}} = \frac{1}{\frac{3}{10} + \frac{0,7}{3}} = \frac{1}{\frac{3}{10} + \frac{7}{30}}$$

$$= \frac{1}{\frac{3 \times 3}{30} + \frac{7}{30}} = \frac{1}{\frac{9+7}{30}} = \frac{1}{\frac{16}{30}} = \frac{30}{16} = 1,875$$

4. Give an example why the following code (of some process i) is an inappropriate solution to the critical section problem:

```
do {  
  
    while (turn == j);  
        critical section  
    turn = j;  
        remainder section  
} while (true);
```

This can cause deadlocks so it violates the progress requirement

5. Give three examples where a multithreaded implementation of an application is better than a single threaded implementation:

- Server because it needs one thread to keep listening for connections and others to execute requests
- Program that makes I/O requests and manipulates the data. Multiple threads are better because while one thread is waiting for its I/O request to finish, another thread can process the data.
- Age of Empires (best game ever) to assign each game unit (villager, etc) a thread so that units do not have to execute commands sequentially but concurrently

QUESTION 4 (10 points):

Round-robin process schedulers maintain a ready list of all runnable threads (or processes), with each thread listed at most once in the list. The scheduler simply pre-empts processes and adds them to the beginning of the list. If a thread is listed twice in the list, briefly explain how this could cause programs that use synchronization primitives to break on a uniprocessor.

Thread control blocks contain the instruction counter and stack. If a thread is scheduled twice, we might get one instance of a thread that's in the critical section and the other that isn't and this could break the mutual exclusion requirement of that critical section : two threads might be in the critical section and we will have integrity issues.

mutex

↗