17-Week Bridge Practice Exam 4 - Answers

We've compiled some extensive exercises covering *networking* and *operating systems* topics, so that you may work on building your competence around these topics before your exam.

Exercises are not at all reflective of the exam. Both the precise topics covered and the way questions are presented will be different than below. Also, please remember that you are expected to remain familiar with topics covered in previous exams. The below questions primarily serve as exercises to get you going while you study networking and operating systems.

This practice test is light on object oriented material. Make sure to review those sections before your test. There will almost certainly be a class-design problem on your exam.

0.) In computer networking, what are the seven layers of the OSI model? Why is it useful to employ a conceptual model such as the OSI model when developing or describing communications systems?

7 Application - Provide user interface to send and receive the data
6 Presentation - Encrypt, format and compress the data for transmission
5 Session - Initiate and terminate session with remote system
4 Transport - Break data stream in smaller segments + provide reliable + unreliable data delivery
3 Network - Provide logical addressing
2 Data Link - Prepare data for transmission
1 Physical - Move data between devices

One possible answer of many: Conceptual models and other theoretical models are useful because they allow for abstractions. You do not have to know how other layers work in order to write software that operates at a single level.

1.) In the transport layer of the OSI model, both connection-oriented transmissions and connectionless transmissions are supported. Name a protocol that is used for each.

TCP - connection-oriented
UDP - connectionless

2.) What is a subnet mask? How is it calculated?

A Subnet mask is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. The subnet mask is made by setting network bits to all "1"s and setting host bits to all "0"s.

3.) What happens when you enter the command `traceroute 127.0.0.1`? What happens when you enter the command `traceroute nyu.edu`? (Replace `traceroute` with `tracert` in Windows.)

127.0.0.1 is a special-purpose IPv4 address also known as local host. Traceroute will only have 1 step, as the only route from localhost to itself is to stay at localhost.

4.) How does a web cache work? What advantage does it provide?

Web Caching is storing of HTTP responses temporarily for fast retrieval later on. Advantage of web caching: Web caching reduces the number of requests made to the server.

5.) What is a socket?

A socket is like a door. Data are pushed out the socket onto the network layer, and received into a socket from the network layer.

6.) Identify four application-layer protocols and give an example of an application that each protocol services.

HTTP - browser
SMTP/ POP3 - email client
DNS - DNS resolver

7.) What are LANs, MANs, and WANs? Provide an example of each.

Local area network - a group of computers connected together, usually in the same building

Metropolitan area network - A network that covers perhaps several buildings or campuses in a single city

Wide area network - Not really restricted to a geography, but in practice may be on the level of a state or country

8.) When would it provide an advantage to use a persistent HTTP connection, rather than a non-persistent HTTP connection?

Reduced latency in subsequent requests (no handshaking).
Reduced CPU usage and round-trips because of fewer new connections
Enables HTTP pipelining of requests and responses.
Reduced network congestion (fewer TCP connections).
Errors can be reported without the penalty of closing the TCP connection.

9.) What is data encapsulation?

Data encapsulation is the process of taking data from one protocol and translating it into another protocol, so the data can continue across a network. For example, an HTTP packet contained within a TCP/IP frame is a form of encapsulation.

10.) What can happen when the physical layer consists of cables that are longer than the prescribed length for that physical medium?

Signal loss can occur.

11.) What are routers? At which OSI layer do routers operate?

A router is a networking device that forwards data packets between computer networks. Routers perform the traffic directing functions on the Internet.

Routers operate at the network layer, generally speaking.

12.) How do cookies work?

An HTTP cookie is a small piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing. Cookies remember stateful information and overcome the stateless nature of HTTP.

Cookies are set using the Set-Cookie HTTP header, sent in an HTTP response from the web server. This header instructs the web browser to store the cookie and send it back in future requests to the server.

13.) What is the difference between a program and a process?

A process invokes or initiates a program. It is an instance of a program that can be multiple and running the same application.

14.) How is switching between threads in the same process different from switching between threads in different processes?

The main distinction between a thread switch and a process switch is that during a thread switch, the virtual memory space remains the same, while it does not during a process switch. Both types involve handing control over to the operating system kernel to perform the context switch. The process of switching in and out of the OS kernel along with the cost of switching out the registers is the largest fixed cost of performing a context switch. Also, when you change virtual memory spaces, the processor's Translation Lookaside Buffer (TLB) or equivalent gets flushed, making memory accesses much more expensive.

15.) Your boss is worried about mutual exclusion. What could you present as a solution ?

Semaphore, spinlocks, many other solutions

16.) What are the properties of an ideal process scheduler? Does this ideal process scheduler exist in the real world?

An ideal process scheduler is absolutely fair, gives a process only as much time as it needs, perfectly predicts when a user needs a response, minimizes context switching.

This is impossible because OS would have to be omniscient, always know what processes need, and be perfectly and objectively fair.

17.) What is the principle of locality and why is it important?

Principle of locality: Programs access a small proportion of their address space at any time

The most important program property that we regularly exploit is locality of references: Programs tend to reuse data and instructions they have used recently. So we can store them in such a way, such as a cache, that makes reuse easier.

18.) When does a page fault occur?

A page fault occurs when a program accesses a page that has been mapped in address space, but has not been loaded in the physical memory.

19.) You are designing a compiler. You want to implement a feature that checks for mismatched or missing brackets. What data structure might be useful here?

A stack would be a useful data structure here, as brackets should be popped off in the reverse order in which they were pushed. Last in, first out (LIFO)

20.) Where are red-black trees used in the real world? Hint: think operating systems.

Red-black trees are used in the Linux scheduler.
Also, std::set and std::map in the C++ standard library.

21.) MS-DOS allowed all user processes to access the kernel directly, without OS control and supervision. Why is this not allowed by more modern, widespread operating systems?

This is the major reason why MS-DOS crashed so much. Third-party programs had too much control and could crash the entire system.

22.) What are the three strategies for resolving deadlocks? Describe each one.

Deadlock Prevention: this strategy attempts to prevent deadlocks from being able to ever occur. It hinges on removing one of the 4 conditions necessary for creating a deadlock: mutual exclusion, hold-and-wait, no-preemption and circular wait.

Deadlock Avoidance: in this deadlock strategy, we say that deadlocks do have the ability to occur, but we will avoid them if there is the danger of one occurring. In order to implement deadlock avoidance, the OS needs to be aware of the entire state of the system at any given point.
Deadlock Detection: in this strategy, deadlocks are able to occur and allowed to occur. They are expected. The idea is that the OS will periodically check the system for any deadlock, and if one has occured, it will somehow deal with it. This is the method used most commonly today.

23.) What is a critical section? Why are critical sections important?

A critical section is a segment of code that needs to be run as an atomic instruction – that is, it is a piece of code that needs to come with a guarantee that it will never be interrupted once it begins running. It will be able to proceed from beginning to end without being stopped and restarted at a later time. Critical sections are important because if the code is interrupted for any reason (e.g. hardware interrupt, clock interrupt), asynchrony could occur.

24.) How is a thread different than a process?

Threads all share the same address space, processes do not. So sharing is easier amongst threads.

Threads are more lightweight than processes, meaning they are easier to create.

There is no 7-state model for threads: they can just exist in the ready, running or blocked states. Processes use the new, exit and two suspended states.

Threads handle execution, processes handle resource grouping

Threads can be implemented on the user-level or kernel level, or in a hybrid implementation. If threads are implemented on the user level, they can't be interrupted like processes and need to yield voluntarily.

25.) Give an example of an event that would cause a process to:

a. **Go from blocked state to ready state**

An I/O operation that the process was waiting on, completes.

b. **Go from running state to ready state**

Scheduler picks another process to run.

c. **Go from ready state to running state**

Scheduler picks this process to run.

26.) Why does an operating system need a scheduler?

An operating systems needs a scheduler to determine fair and efficient distribution of OS resources (however we choose to define fair and efficient).

27.) What is preemption? Why is it significant? When does it occur?

Preemption is the act, performed by an OS, of stopping a running process, saving all the information about its running state when it was stopped (saving all its registers, stack pointers, etc.), and using that information to restart it at a later point from where it left off.

Preemption is at the heart of computer multitasking, which is the modern paradigm of how computing is done. Preemption allows us to create the illusion of parallel processing (sometimes called pseudo-parallelism) by having the OS rapidly switch between running programs (or processes/threads), as many as hundreds or thousands of times per second, allowing each program to carry out some instructions before being preempted and exchanged with another program (which can include the OS itself).

28.) What is a context switch?

A context switch refers to the process of storing the state of a process or thread so that it can be restored at a later time from the exact same point. This allows multiple processes to share the CPU and is essential to the concept of multitasking.