1. Name and briefly describe the services of each layer of the TCP/IP protocol stack.
   1. **Application: protocol**
   2. **Transport: TCP/UDP**
   3. **Network – datagram**
   4. **Link – routers, frames**
   5. **Physical – transmission, mediums**
2. What is ARP and how does it work?
   1. **Address resolution protocol**
   2. **broadcasts IP address looking for the assigned MAC address. Maps IP address to physical address, uses table to search for matches, if no match then**
3. What is the main feature of non-blocking I/O?
   1. **Main feature of non-blocking IO is typical input and output calls, and accept, return EWOULDBLOCK if they cannot complete immediately, while connect return EINPROGRESS if it cannot complete immediately.**
4. Give a complete description about the code at the given line number in the following code segments.
5. Specify the order of the following events [1-8] for a client connection to a concurrent server.
6. **Server blocks on a listen fd**
7. **Client uses connect() to request a connection**
8. **Server creates new connection fd with client**
9. **Server child is created**
10. **Child closes listen fd**
11. **Server closes connection fd**
12. **Child closes connection fd**
13. **Server closes listen fd**
14. What function is used to wait for a thread to terminate and by a thread to obtain its ID or terminate itself?
    1. **pthread\_join - wait for thread termination**
    2. **pthread\_self – ID**
    3. **pthread\_exit – terminate**

**thread vs. child process:**

**child process is an exact copy of the calling process**

**thread - a stream of instructions that can be scheduled as an independent unit. Share same addresses in processes, lightweight.**

**process contains two kinds of information: resources that are available to the entire process such as program instructions, global data and working directory, and schedulable entities, which include program counters and stacks. A thread is an entity within a process that consists of the schedulable part of the process.**

1. Which memory structures, stack, program counter, file descriptors, errno, etc… are shared by all threads in a process, and what function does it call to indicate that resources associated with it can be released on termination?
   1. **Threads share a common point in memory (everything) but stack, stack pointer, and registers.**
   2. **pthread\_exit**
2. Write a code fragment that creates a thread that executes a function called f. the null pointer is passed as an argument to this function.
3. Describe how traceroute works.
   1. **works by sending UDP datagram with gradually increasing TTL value, starting with TTL value of one. The first router receives the packet, decrements the TTL value and drops the packet because it then has TTL value zero. The router sends an ICMP Time Exceeded message back to the source.**
   2. **What it does: show you the path a packet of information takes from your computer to one you specify. It will list all the routers it passes through until it reaches its destination, or fails to and is discarded. In addition to this, it will tell you how long each 'hop' from router to router takes.**
4. What happens if an I/O operation is called on a non-blocking socket and it’s not able to immediately complete the I/O operation? What happens to an accept or connect operation in the same situation?
   1. **There would be an error return, EWOULDBLOCK**
   2. **Same EWOULDBLOCK, if new connection is not available**
   3. **Connect: allows other processes while TCP’s three-way handshake takes place, instead of being blocked in the call. Still return EINPROGRESS.**
5. For the given source code, describe numbered lines.
6. What is a raw socket? How does it differ from a cooked socket? What can it be used for?
   1. **Allows direct sending and receiving without using TCP or UDP sockets, internet socket.**
   2. **Normal sockets - cannot read/write ICMP or IGMP protocols with normal sockets Ping tool cannot be written using normal sockets. Read and write IPv4 datagrams with protocols not processed by the kernel**
   3. **It’s used to generate/receive packets to/from specific user applications**
7. What is subnet-directed broadcasting and what must be done for a socket to send a broadcast packet and why?
8. Describe an advantage and disadvantage of using threads instead of child processes to accomplish concurrent tasks.
   1. **Much quicker to create a thread than a process and they share data easily.**
   2. **No security between threads, if one thread blocks, all threads in task block.**
9. Describe the data contents and purpose of packets sent to a ping server.
10. **Describe the three essential factors that permit race conditions to occur?**
11. What is ransomware and what would be good advice for someone who is a victim? Why is this bad advice? What can be done to address the threat of ransomware? What is the purpose of preserving a copy of the encrypted data of a ransomware victim?
    1. **Malware threat, block access to a computer system or encrypts files until a sum of money is paid.**
    2. **Disconnect from the internet, avoid paying ransom, research, remove ransomware from the device.**
    3. **Use backups that are remote and unconnected, up-to-date antivirus**
    4. **Recover from shadow copies or to decrypt**
12. (sockets) what is a socket, how does it relate to an IP address and a port number, and how is it used in a computer application to do client/server networking programming.
    1. **A socket pair defines two endpoints of the connection, local IP address and port, and foreign IP address and port. The two values that identify each endpoint, an IP address and a port number, are often called a socket.**
    2. **socket address is the combination of an IP address and a port number**
13. (TCP and UDP) what kind of client/server network application would you select TCP for and why? What kind of client/server network application would you select UDP for and why?
14. (Java/Python vs. C) What are the major similarities and difference in coding a TCP client/server application Java/Python compared to fulfilling the same requirements but coding the application C instead.
    1. **Python - stable, secure slower high-level language, focuses more on the “what” than the “how”, wants indentation, is interpreted, abstracted from lower level, errors that aren’t discovered until runtime, dynamic typing, usually simpler, as in more readable and writable.**
    2. **C - fast low-level language, detailed, able to rewrite certain parts that are automatic in Python, memory copying between socket buffers, is compiled, little abstraction from computer architecture, pointers and memories map closely to processors, static, can do whatever python does and more,**
15. (Network Programming) what major items do you need to address when designing and implementing a client/server network application that you do not need to consider when designing and implementing a stand-alone software application?