Jeffrey Hughes

0026991582

Final Exam Homework - CS252 Spring 2016 - Gustavo

1 - 6 are on the back.

7. What are the four parameters that a computer needs to be able to get connected to the internet and what are they used for?

* IP address - Current IP address
* Subnet mask - Used to send packets on LAN
* Default router - Used to send packets outside LAN
* DNS Server - Used to resolve host names

8. How does a computer know when it can deliver a packet directly and when it has to pass a packet to a router?

* By referring to a routing table and doing a bitwise-and operation between the destination IP and the subnet mask of each route entry.
* If it matches a target network, it will send to the router specified in the entry.
* If the target network is local, packet is sent directly.
* Otherwise, it is sent to the default router.

9. What does ARP mean and how does it work?

* Address Resolution Protocol.
* It converts an IP address into a hardware address when a router or host delivers a packet directly. It accepts an IP address of a computer in locally connected network and outputs the Ethernet address of the computer. The bindings (IP address, hardware address) are kept in an ARP table or an ARP cache.

10. What does DNS mean and what it is used for?

* Domain Name Server
* Domain name (i.e. google.com) -> IP (128.210.224.45)

11. What does DHCP mean and how does it work?

* Dynamic Host Configuration Protocol.
* It is a server that assigns each computer the IP address, subnet mask, DNS server, and default router.

12. What does UDP mean?

* User Datagram Protocol
* It is mainly used for broadcasting and for real-time data applications.

13. What does TCP mean? What are the 6 features of TCP?

* Transmission Control Protocol
* Adaptive retransmission
* Cumulative acknowledgements
* Fast retransmission
* Flow control
* Congestion control
* Reliable connection and shutdown

14. When should you use TCP and when should you use UDP?

* TCP is used when a reliable connection is needed. UDP is needed when a computer needs to broadcast messages to multiple computers or when it needs to use real-time data applications where data needs to be sent without delay. Retransmissions will add to the delay so TCP is slow for those applications.

15. What does NAT stand for? Assume that a packet <A, 4563, X, 80> is sent from a host behind a NAT box to a webserver X. Describe the steps for the translation (6 steps) since it goes from the host A, through the NAT box, to X and then back from X to the NAT box to A.

* Network Address Translation.
* Host A wants to establish connection with webserver X in the Internet.
* NAT box is default router so A sends TCP packet to it.
* NAT box chooses a random unused port (1234) and substitutes the source port with it. It also changes the source IP into its own IP address (B). Packet (B, 1234, X, 80) then sent to X.
* The old source IP (A) and port (4563) are added into the NAT table for use when packets come back.
* When packet is back from X, NAT box looks up the destination port (1234) in NAT table and substitute the destination IP and port with original values (A, 4563).
* Packet is forwarded to A. Everything will be transparent to both ends. The mapping will be used until connection is closed or timed out.

16. Explain why NAT boxes can be used as firewalls to prevent unwanted connections. Also explain why it is not normally possible to run web servers behind a firewall and how this problem can be solved.

* NAT box only passes packets if connection is initiated inside the network. If packet arrives to NAT box and does not have existing entry in NAT table, packet will be discarded. This will protect the private network from hacker attacks. It is often not possible to run web servers behind a firewall because connections will start from outside the network and there is no mapping in NAT table yet. Static entries can be added manually to the NAT table so that NAT box knows where to forward the incoming packets.

17. Write a simple client program "echo-client host port string" that sends a string "string" followed by "\r\n"to "host : port" and then it reads the server's response and prints it to stdout.

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <netdb.h>

#include <assert.h>

int main(int argc, char \*\* argv) {

char \* host = argv[1];

int port = atoi(argv[2]);

char \* echostring = argv[3];

struct sockaddr\_in socketAddress;

memset((char \*)&socketAddress, 0, sizeof(socketAddress));

socketAddress.sin\_family = AF\_INET;

socketAddress.sin\_port = htons((u\_short)port);

struct hostent \* ptrh = gethostbyname(host);

memcpy(&socketAddress.sin\_addr, ptrh->h\_addr, ptrh->h\_length);

struct protoent \* ptrp = getprotobyname("tcp");

int socket = socket(PF\_INET, SOCK\_STREAM, ptrp->p\_proto);

if (connect(socket, (struct sockaddr \*)&socketAddress, sizeof(socketAddress)) < 0) {

perror("connect");

exit(1);

}

write(socket, echostring, strlen(echostring));

write(socket, "\r\n", 2);

char buf[1024];

int n = recv(socket, buf, sizeof(buf), 0);

while (n > 0) {

write(1, buf, n);

n = recv(socket, buf, sizeof(buf), 0);

}

close(socket);

}

18. Write a simple iterative server "echo-server port" that waits for incoming requests in "port" and once it receives a string delimited by "\r\n" it will reply with the same string plus "\r\n" and close the connection.

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <netdb.h>

int main(int argc, char \*\* argv) {

int queueLength = 5;

int port = atoi(argv[1]);

struct sockaddr\_in socketAddress;

memset((char \*)&socketAddress, 0, sizeof(socketAddress));

socketAddress.sin\_family = AF\_INET;

socketAddress.sin\_addr.s\_addr = INADDR\_ANY;

socketAddress.sin\_port = htons((u\_short)port);

int masterSocket = socket(PF\_INET, SOCK\_STREAM, 0);

int optval = 1;

int err = setsockopt(masterSocket, SOL\_SOCKET, SO\_REUSEADDR, (char \*)&optval, sizeof(int));

err = bind(masterSocket, (struct sockaddr \*)&socketAddress, sizeof(socketAddress));

err = listen(masterSocket, queueLength);

while (1) {

struct sockaddr\_in client;

int alen = sizeof(client);

int clientSocket = accept(masterSocket, (struct sockaddr\*) &client, (socklen\_t\*) &alen);

const int maxLength = 1024;

char buf[maxLength + 1];

char newC, lastC;

int curLen = 0;

int n;

while (curLen < maxLength && (n = read(clientSocket, &newC, sizeof(newC))) > 0) {

buf[curLen] = newC;

curLen++;

if (lastC == '\r' && newC == '\n') {

break;

}

lastC = newC;

}

buf[curLen] = '\0';

write(clientSocket, buf, strlen(buf));

close(clientSocket);

}

}

19. Enumerate 5 of the 12 questions in "Joel's Test".

* Do new candidates write code during their interview?
* Do you use the best tools money can buy?
* Do you fix bugs before writing new code?
* Do you make daily builds?
* Do you have a spec?

19. What is XP programming?

* Extreme Programming. It is a practical methodology for software development. It has a list of rules that have been proven successful in software development. It encourages an iterative approach.

21. From XP Programming, mention 4 items from the Planning List, 4 Items from the Coding List, 4 Items from the Designing List, and 4 Items from the testing List.

* Planning
  + User stories
  + Release planning
  + Frequent small releases
  + Measure project development
* Coding
  + Have customers available
  + Follow coding standards
  + Code unit tests first
  + Use pair programming
* Designing
  + Keep it simple
  + Refactor
  + Use CRC cards
  + Choose system metaphor
* Testing
  + All code must have unit tests
  + All code must pass unit tests before integration
  + When a bug is found, create a test
  + Create acceptance tests

22. Explain 5 uses of the source control system.

* Keep track of changes by multiple programmers.
* Merge code easily.
* Evaluate level of contribution of others.
* Backup of sources.
* Find out what changes broke the daily build.

23. Describe the advantages and disadvantages of centralized vs. distributed source control systems.

* Centralized - Advantages:
  + One centralized, common source.
  + Easy to obtain latest source code.
* Centralized - Disadvantages:
  + If server crashes, no one can do anything
* Distributed - Advantages:
  + Programmers can work for a long time without submitting to common code.
  + Easy to make changes since everyone has a local repository.
* Distributed - Disadvantages:
  + Hard to obtain one true source that works for all

24. Describe the 4 types of tests, who writes these tests in the organization, and when do they run.

* 1. Unit tests
  + Group of tests that test a specific class.
  + Written by programmers.
  + Run after writing a class. Sometimes even before writing any code.
* 2. System tests
  + Test a specific subsystem of product or multiple classes involved in a feature.
  + Written by QA and programmers.
  + Run in daily build.
* 3. Regression tests
  + Test written after bug is fixed to verify that it doesn't break anything else.
  + Written by programmers.
  + Run after a bug is fixed.
* 4. Acceptance tests
  + Evaluate quality of software before it is released to tell whether it is ready for prime time.
  + Written by QA.
  + Run before a software release.

25. Explain why it is important to have a bug tracking system.

* To keep track of all the bugs that are reported and to assign priority and severity to them. This will make sure that the software has a reduced number of bugs and improve the quality of the software.

26. Explain the difference between Priority and Severity in a bug.

* Priority is how important is the bug to the software developers and the organization. Severity is how the bug affects the user.

27. Mention 5 cases when you can apply refactoring.

* Long method
* Long parameter list
* Duplicated code
* Large class
* Inconsistent names

28. What is a Software Pattern, what are the parts of a software pattern? What is the name of the book that introduced software patterns and the authors?

* Software patterns are reusable design solutions for recurring problems.
* Pattern name
* Synopsis
* Context
* Forces
* Solution
* Consequences
* Implementation
* Related patterns
* The book is "Design Patterns: Elements of Reusable Object-oriented Software" by Gamma, Helm, Johnson, and Vlissides.

29. Describe the Proxy Pattern and 2 applications.

* Proxy pattern is to force method calls to an object to occur through a proxy object.
* Act as logger object
* Act as a cache

30. Describe the Command Pattern and two applications.

* Command pattern encapsulates commands in objects so you can control selection and sequencing, queue, undo and manipulate them.
* Graphical editor that has do/undo
* Word processor that has do/undo

31. What is the difference between Code Instrumentation Profiling and Statistical Sampling Profiling.

* Code instrumentation profiling adds instructions into the code before a method starts and after a method returns to measure the execution of a method. Statistical sampling samples the program at regular intervals to find out what the program is doing.

32. Explain why Optimizing should be left until the very end in the software cycle and why you should use an execution profiler before attempting to optimize a program.

* Optimizing should be left to the very end because it makes the code difficult to read and understand. Execution profiler should be used to find out which part of the code takes more time and needs to be optimized. In general, only small portions of code need to be optimized.

33. Assume the following table called "customers":

|  |  |  |  |
| --- | --- | --- | --- |
| **CompanyName** | **ContactName** | **Address** | **City** |
| Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin |
| Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå |
| Centro comercial Moctezuma | Francisco Chang | Sierras de Granada 9993 | México D.F. |
| Ernst Handel | Roland Mendel | Kirchgasse 6 | Graz |
| FISSA Fabrica Inter. Salchichas S.A. | Diego Roel | C/ Moralzarzal, 86 | Madrid |
| Galería del gastrónomo | Eduardo Saavedra | Rambla de Cataluña, 23 | Barcelona |
| Island Trading | Helen Bennett | Garden House Crowther Way | Cowes |
| Königlich Essen | Philip Cramer | Maubelstr. 90 | Brandenburg |
| Laughing Bacchus Wine Cellars | Yoshi Tannamuri | 1900 Oak St. | Vancouver |
| Magazzini Alimentari Riuniti | Giovanni Rovelli | Via Ludovico il Moro 22 | Bergamo |
| North/South | Simon Crowther | South House 300 Queensbridge | London |
| Paris spécialités | Marie Bertrand | 265, boulevard Charonne | Paris |
| Rattlesnake Canyon Grocery | Paula Wilson | 2817 Milton Dr. | Albuquerque |
| Simons bistro | Jytte Petersen | Vinbæltet 34 | København |
| The Big Cheese | Liz Nixon | 89 Jefferson Way Suite 2 | Portland |
| Vaffeljernet | Palle Ibsen | Smagsløget 45 | Århus |
| Wolski Zajazd | Zbyszek Piestrzeniewicz | ul. Filtrowa 68 | Warszawa |

Write the result of the following queries (You can use a description when the number of rows in the resulting table is larger than 5, otherwise write down the whole resulting table).

a) SELECT \*FROM customers

* The entire table

b) SELECT ContactName FROM customers

* The entire table but only has 1 ContactName column

c) SELECT CompanyName FROM customers WHERE ContactName LIKE Liz%

* The Big Cheese

d) SELECT CompanyName, ContactName FROM customers WHERE City LIKE Portland

* The Big Cheese | Liz Nixon

e) Write a query to get the companies that are in Spain

* SELECT CompanyName from customers WHERE City LIKE Madrid OR City LIKE Barcelona

f) Write a query to get all the companies that start with R or W

* SELECT CompanyName from customers WHERE CompanyName LIKE 'R%' OR CompanyName LIKE 'W%'