

CS252 Final Review Homework

CS252 Final Review Homework

Answer this final review homework and return it at the beginning of the final exam.

1. Write a program "grepsort arg1 arg2 arg3" that implements the command "grep arg1 | sort < arg2 >> arg3". The program should not return until the command finishes. "arg1", "arg2", and "arg3" are passed as arguments to the program. Example of the usage is "**grepsort hello infile outfile**". This command will print the entries in file **infile** that contain the string **hello** and will append the output sorted to file **outfile**. Use pipes. Do error checking. Notice that the output is appended to arg3.

2. Complete the procedure `runCommand(command, outputBuffer, bufferSize)` that executes a command in a different process and stores its output in `outputBuffer`. `command` is the name of the program with no arguments. See how `main` uses `runCommand()`. `runCommand` will return 0 on success or -1 otherwise. Use a pipe to communicate the parent and the child process running `runCommand()`. Have the parent read from the pipe and write into the `outputBuffer`.

```
int
runCommand( char * command, char * outputBuffer, int maxBufferSize)
{
```

```

}
int
main()
{
    // The output of "ls" will be stored in buffer
    char buffer[ 1024 ];
    if ( runCommand( "ls", buffer, 1024 ) < 0 ) {
        perror("runCommand");
        exit( -1 );
    }
}
```

```

    printf( "Is: %s\n", buffer );
    exit( 0 );
}

```

3. Add the necessary code to the insert() and removeFirst() functions to make them synchronized. removeFirst() will have to wait if the list is empty. insert() will have to wait if there are already 20 elements in the list. **Use semaphores.** Add also the variables you need.

```

struct List {
    int val;
    int next;
};
struct List * head = NULL;
// More variables

main()
{
    // DO any initializations here

}
void insert( int val )
{
    List tmp = new List;

    tmp->val = val;

    tmp->next = head;

    head = tmp;
}
Struct List * removeFirst()
{
    List tmp = head;

    head = tmp->next;

    return tmp;
}

```

4. Using C++ and Semaphores write a class SynchronizedStackSemaphores of int values where pop() will block if the stack is empty and push will block if the stack is full. Write the member variables that you think are necessary. Implement the stack with an array of int's and allocate it dynamically in the constructor. Hint: Use the "Bounded Buffer Problem" with semaphores as an example in your implementation.

```

#include <synch.h>
#include <pthread.h>

class SynchronizedStackSemaphores {
    // Add your member variables here
    int top;
    int * stack;
}

```

```

    public:
        SynchronizedStackSemaphores(int maxStackSize);
        void push(int val);
        int pop();
};

SynchronizedStackSemaphores::SynchronizedStackSemaphores(int maxStackSize) {
    top = 0;
    stack = new int[maxStackSize];

}

void SynchronizedStackSemaphores::push(int val) {

}

int SynchronizedStackSemaphores::pop(){

}

}

```

5. From lab3, assuming you have a procedure *void dispatchHTTP(int slaveSocket)* that processes the request and closes *slaveSocket*, write the loop server code for a) iterative server, b) concurrent server using fork, c) concurrent server creating a thread after each request, and d) pool of threads, in the procedures indicated. Each procedure receives as argument the master socket already initialized and ready to be used inside accept.

```

void iterativeServer( int masterSocket) {

}

void forkServer( int masterSocket) {

}

}

void poolOfThreads( int masterSocket) {

}

}

```

<https://docs.google.com/document/d/e/2PACX-1vTffLIWCeL8T2fEVqrObscPUKb4NSxliJ6r93IAiOkHSQgwyBzARRgEpLXeTnBYfXG1yeoj...> 4/8

8. How does a computer know when it can deliver a packet directly and when it has to pass a packet to a router?
9. What does ARP mean and how does it work?
10. What does DNS mean and what it is used for?
11. What does DHCP mean and how does it work?
12. What does UDP mean?
13. What does TCP mean? What are the 6 features of TCP?
14. When should you use TCP and when should you use UDP?
15. What does NAT stand for? Assume that a packet $\langle A, 4563, X, 80 \rangle$ 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.
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.
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.

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.
19. Enumerate 5 of the 12 questions in "Joel's Test".
20. What is XP programming?
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.
22. Explain 5 uses of the source control system.
23. Describe the advantages and disadvantages of centralized vs. distributed source control systems.
24. Describe the 4 types of tests, who writes these tests in the organization, and when do they run.
25. Explain why it is important to have a bug tracking system.
26. Explain the difference between Priority and Severity in a bug.
27. Mention 5 cases when you can apply refactoring.
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?

29. Describe the Proxy Pattern and 2 applications.

30. Describe the Command Pattern and two applications.

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

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.

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/ Morazarzal, 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
- b) SELECT ContactName FROM customers
- c) SELECT CompanyName FROM customers WHERE ContactName LIKE Liz%
- d) SELECT CompanyName, ContactName FROM customers WHERE City LIKE Portland
- e) Write a query to get the companies that are in Spain
- f) Write a query to get all the companies that start with R or W

Published by [Google Drive](#) – [Report Abuse](#) – Updated automatically every 5 minutes
