CS35 Final Review

Structure:

- simple questions
- harder questions
- write some code (python)
- modify/fix code (C) & patch 2-3 hours

150 min, so 150 points

Sample Questions:

The following questions are worth 5 points each.
simple questions like these probably not on Jiwen's exam.....

1. What's the difference between sed and grep? be concise.
grep finds. sed edits

2. What does the following script print out?

#!/bin/grep is
Hello
this
is
a test!

what abut #! /usr/bin/env python *(python is the argument)*
#! /usr/bin/bash

this
is

3. What's a symbolic link (often referred to as a soft link)? How does this differ from a non-symbolic link (often refered to as a hard link)?
files are represented as iNodes
a soft link's file contents is the name of the file pointed to
a hard link's file has the same inode as the original file! (same data block on HDD)
hard links don't waste space

4. When you don't know how to do something in Linux, where should you first look (other

than online)?
manual pages
apropos
whatis

5. What's stored on a thread's (or process' for single threaded applications)stack?
local variables/buffers
function arguments
return addresses
old base pointers

[something about buffer overflow definitely on exam]

The following questions are worth 7 points each.
6. As part of a class project you and a partner share access to an svn repository hosted on the class server. One night, without your knowledge, your partner adds a backdoor (a secret piece of code allowing remote control of the project) to the project repository. However, you do not realize this until after the project is turned in. How can you show (with reasonable certainty) that you did not add the back door to the project? List the specific commands you would use and what useful information they would provide.
git blame $file

we can pick svn or git
[Jiwen doesn't like this question; it's knowledge-based]

7. In 2009 an attack on SSL (Secure Socket Layer) Certificates was published online that allowed spoofing of domain names. This attack allowed a malicious site to publish valid security certificates which appeared to be for a different site. For example, www.goodsite.com.evilsite.com (a subdomain of evilsite.com) would claim to be www.goodsite.com by having a non-printable null character ("www.goodsite.com\0.evilsite.com") in its name. Given that every domain name is exactly 32 characters (with null byes on the end when less than 32 characters are used), eliminate this spoofing vulnerability from the following function:

```
/* Returns 0 when the certificate is valid for the site and non-zero otherwise*/
int isValidForSite(char* domainName, char* certificateDomainName)
{
        return strcmp(domainName, certificateDomainName);
}
```

Write your corrected function below:

use a for loop:
isValid(char* d, char* c)

```
{
        for (int i = 0; i < 32; i++)
        {
                if(*(d+i) == *(c+i))
                        continue;
                if (*(d+i)=='\0')
                        if(*(d+i+1) != '\0')
                                return false;
        }
}
```

OR use memcmp:

```
memcmp(domainName, certificateDomainName, 32);
```

strcmp reads until it encounters a nullbyte. BADDD!!!
strcmp (dest, src)
strcpy (dest, src)
the two functions above run until they find a null byte

8. SSH is a widely used network protocol that allows for secure (in most typical situations) communication between two systems. SSH allows for both Public Key (where a client's public key is stored on the host) and Password based authentication which you saw in Lab 7. List a set of circumstances where Public Key authentication would be the most secure method and a set of circumstances where Password based authenticated would be the most secure.

prevent other users from accessing .ssh folder:
chmod -r . 700
ssh keys are protected by passphrase
avoid typing password--prevent keyloggers

write the answer point-by-point (by situation)
"we will give you more credits" --jiwen

9. which of the following bash commands would be most appropriate to compile a project and print one line summary of the build status? Justify your answer by critiquing all three commands.

a. make || echo "Build failed!" && echo "Build succeeded"
b. make && echo "Build Succeeded" || echo "Build failed!"
b is the correct answer
c. make || echo "Build succeeded" && echo "Build failed!"

"||" means do the left *or* the right, depending on return value

Jiwen doesn't know what this question means, so it won't be on the final.

10. Write a `sed` command that replaced every phone number in ./contact.html with its digit-only form. The format to consider is (ddd)-ddd-dddd. For example, (555)-555-5555 becomes 5555555555. Ignore phone numbers formatted incorrectly (for example, too many digits).
sed -d [()-]
sed 's/[()-]//g'

or to make sure it only acts on phone numbers, use \1 through \9

```
 sed 's/(\([0-9]\{3\}\))-\([0-9]\{3\}\)-\([0-9]\{4\}\)[[:space:]]/\1\2\3/g' < contact.html
```

The following questions are worth 10 points each.
11. You have a file called one_mb which is exactly 1 megabyte in size. You want to create from it a file of exactly 128 megabytes in size. Please write a shell script to do this with at most 9 lines and no loops, if statements, recursion, or any other logic control structures. Each command, including parameters, must be less than 100 characters in length. (max 9 lines)
cat one_mb >> one_mb

12. In the following question, write your answers in Python.
a. Make a class Point 3 that represents a 3-dimensional point. A constructor is optional, but you must implement the function distance2(...) that will compute the distance squared (e.g. (x_1 - x_2)^2 + (y_1-y_2)^2 + (z_1-z_2)T between the current point and given point and return it.

b. Write a function createCoordinates(...) that sequentially reads lines from a file in the form "x y z" (ex. "5 3 4", "2 -1 1") where all numbers are integers. The function should return an array of type Point3 containing all points read. The file name is a function parameter with default value "points.txt" (you can call createCoordinates(...) without a file name specified):

c. A bounding sphere is defined as a sphere which fully encloses a set of geometry. Create a function boundingSphere(...) that takes an array of type Point3 and prints out the coordinates of a bounding sphere's center followed by its radius that encloses all points in the array. you simply need *a bounding sphere* not the smallest possible bounding sphere. (Hint: This can be done with an arbitrary center point and a single pass of the array):

13. The following C function counts the number of unique positive integer divisors given a number *n* has (not including 1 or *n*). For example, if the number 12 is given the function will return 4 since 12 can be divided by 21, 3, 4, and 5. speed up this function by parallelizing it and doing all work in 3 child threads. Use pthread_create and pthread_join

to manage your threads. You may add additional functions and global variables if needed. Do not attempt to improve the algorithm (i.e. still use exhaustive search). You may assume that all operations are atomic, meaning you can perform operations such as g_divisorCount__ without worrying about synchronization.

[code omitted]

Write your new function(s) below:

14. Given the following buggy C program to compute factorials (main.c):

```
#include ,stdio.h>
int factorial(int n)
{
        return n* factorial (n-1);
}
[Rest of code omitted]
```

And the following patch to correct the buggy program (called fix.patch):
[patch omitted]

Notice the typo of "Fractorial" on line 29 on the patch which should be "Factorial"> Please make a patch that can be applied to "fix.patch" and fix the typo (you do not need to use all lines) on the next page.

[Jiwen doesn't understand why this question is here]

Our Exam:

not many knowledge-based questions (i.e. stuff that we can copy from notes)
we *will* have coding things
fewer questions (5-6 large questions)