Name:______________________ UID:______________________

CS 35L: Software Construction Lab
Section 1
Final Examination

FALL QUARTER 2011

THURSDAY, DECEMBER 8, 2011, 11:30AM – 2:30 PM

Examination Requirements:

1 Please write your response clearly in the allocated space. Unreadable answers will not be graded. It may help to first formulate and write your answers on the scratch paper attached in the last page.

2 This examination is open-book, open-notes. Any printed or hand-written materials are permitted. Electronic devices are not permitted.

3 If you have confusions on questions, raise your hand and TA will come to you and clarify the question for you. Any questions other than clarification of the exam will not be answered.

Problem 1 Warm-up [15 Points]

Please answer the following questions using the knowledge you learned from CS 35L.

(1) Suppose you want to create a directory and necessary parent directories. You don't know the exact command and that command may take a number of options. Try to explain how will you search for the command and the proper options. [5 Points]

If I don't know the exact solution, I will use the command apropos to search for keywords in the ma
specific, I'll do $apropos directory because directory seems to be the most related terms. When I g
commands to look into, I will use the man page to look at each of them, read the description of eac
the possible options that might be useful to solve this question.

(2) Write a regular expression to match all the valid function definitions that take a single argument in a python script. [5 Points]
The python function is defined as:
def <identifier>(<identifier>):
A valid identifier in python is starting from alphabetical characters or underscore and follows alphabetical characters or digits or underscore. To make things simple, we assume that there are no white spaces before and after each token except the one after "def" and there are no comments in the code.

```
^def [A-Za-z_][A-Za-z_0-9]*\([A-Za-z_][A-Za-z_0-9]*):$
```
^ means we want to match from the beginning of the text.
[A-Za-z_] means all letters and underscore
[A-Za-z_0-9] means all letters, digits, and underscores
\(and \) are necessary because we have to remove their special meanings
$ means we want to match at the end of the text.

(3) You and Bob are colleagues and you need to send some secret documents. You want to send Bob a file with signature but also encrypted using public key encryptions. In this process, which key would you use to encrypt a message, which key would Bob use to decrypt a message, which key would you use to create the signature and which key would Bob use to verify it? [5 Points]

Before I send anything to Bob, I will use my private key to sign the document, meaning that a signat
using my private key. Depending on the types of the signatures, there will be either one or several o
example, when I choose to use detached signature, there will be the original file I want to send to Bo
file created by me certifying that I initiate the conversation at a specific time. If the file is tampered so
out when verifying the signature using my public key. I then use Bob's public key to encrypt the mes
his private key to decrypt the files. Again, I use my private key to generate the signature, and Bob us
verify the signature.

Problem 2 Programming in Collaborative Environment (20 Points)

You and your friends teamed up to do a course project. You want to utilize the open source tools you have learnt in CS 35L to manage the collaboration. Consider the following scenarios and briefly mention which tool will you choose, commands to complete the specific task and the reasoning of your choice (e.g. What will the tool do? What's the effect of your command?)

(1) As the team leader, you have written up three documentations for the project. "**Contact.txt**" records the contact information of your team members, which should be filled in by your teammates and be readable for everyone in the class. "**README**" defines some instructions to your teammates, and you don't expect any one except you to change it. "**MeetingNotes.txt**" consists notes of internal group meetings, which should be accessible to the team members only.

Assume all of your files are updated to a Linux server, where each student has a Linux user account and belongs to a user group representing the team. Explain how to use Linux commands to manage the access permissions of above files. (Your project folder is at /projects/team-1/ and you are the owner of the folder. You can change the permission settings of the folder and any files under it).
[7 Points]

For "Contact.txt", teammates in the group, teammates should be able to read and write (rw-), corresponding to permission bit 6 for the group. For other people in the class, they only need the read access, so for others, r-- is enough, which corresponds to 4. For me, since this is a txt file, we don't want to make it a executable, so i will only give myself rw- which corresponds to 6. Therefore, we should use the command

```
$chmod 664 Contact.txt
```

to grant corresponding privilege to different groups of users. One thing to take care of is to make sure the file "Contact.txt" actually belongs to the group. If not, we can use the command $chgrp team Contact.txt to change group ownership.
For "README.txt", I assume it should be written by me, so I should have the rw- access, corresponding to permission bit 6. For group and others, only r-- is needed, corresponding to 4. Therefore, we will use the command $chmod 644 README.txt.
For "MeetingNotes.txt", we will use the command $chmod 660 MeetingNotes.txt

(2) Your team use Git to manage the code. After the latest sync, you find out that your program has some bugs and failed in a test case. You remember that the program works well last week, so there is something wrong with latest updates. Explain how to find the bug in the code using Git commands and other necessary tools.
[7 Points]

Since we believe that the bug was introduced at some time in the past week, we want to take a look at the commit logs. To do that, we use the command $git log, and we will get the specific information of each commit, including time, author(s), and the description of the commit. We can then use the git blame command to see what changes were made, and who made the changes. It would also be helpful if we can use a GUI. For example, if the project is hosted on GitHub, we can see the repo network to visualize the commits and changes to files (similar to the output by diff, but with colors, easier for human to read)

We can also use Git diff to compare between commits, to understand what is being changed.

To debug the code, we can use gdb. Commands for debugging:
step: execute one line and stop before next line
next: Similar; if next line is a function call, execute the entire function
until: Similar; jump loops and functions

(3) Suppose you have found out that the bug is in "foo.c" and fixed it. You want to have a "code review" by sending your team an email enclosed your change in the code. What command will you use to generate the content of email?
Suppose your teammates have acknowledged your change and you want to get your fix into your repository. Explain what Git commands will you use to check in your code and how other members sync up with your change. (Suppose you already have a remote repository set up.)
[6 Points]

After I fix it, I will use the command $git diff to make a diff file between the files in the working stage and the file from the last commit. I'll do that before I commit the modified code to the repo. After they approve it, I will then use $git add foo.c and $git commit -m "bug in foo.c fixed" to commit the changes to the repo, and use $git push to push the changes to the remote server.
OR, I can create a commit for the fix first in my own branch, and use
$ git format-patch -1 HEAD > diff.txt
To create patch. When it is approved, we use git merge to merge the content.

Problem 3 Data Analysis using Shell Scripts [20 Points]

You work for a sports consulting company who provides statistic analysis for NBA players. Despite that you have many advanced data mining tools, many tasks can be done simply using shell commands. Try to design and implement your script to satisfy the following customers' demands.

Your data is a tab-separated file formatted as following:
Position College Height Weight Points-per-game Minutes-per-game Draft-Year Player-Name
For example, the record of Vince Carter and Alan Iverson looks like the following:

```
<Guard>/<Forward>  UNC        6-6  215  22.2  36.4  1998    Vince Carter
<Guard>            GeorgeTown 6-0  165  26.7  41.1  1996    Alan Iverson
```

The file is **nba.txt**. (Hint: Using the text processing tools we introduced in class such as grep, sed, tr etc. You can either write a command or a script containing multiple commands)

(1) One high school basketball star get offer from many different schools. He wants to know which school is better in terms of number of their graduations playing in NBA. Explain how would you use simple Linux command to compare two schools. Your script takes two arguments that are school names, and exists with status 0 if the first school has more NBA players and 1 otherwise.
e.g. ./compare.sh UCLA USC should exit with code 0 since UCLA is better than USC. ☺ [6 points] (There is more space to write on next page)

two arguments of school names; 0 means s1 > s2, vice versa.

commands needed (listed below)

```
#!/bin/bash
SCHOOL1=$(cat nba.txt l grep $1 l wc -l)
SCHOOL2=$(cat nba.txt l grep $2 l wc -l)

if (( $SCHOOL1 > $SCHOOL2 )); then
  exit 0
else
  exit 1
fi
```

(2) One NBA team manager is looking for a player who can play two or more positions (e.g. <Guard>/<Forward>, <Center>/<Forward>. We assume that all the positions are valid in the record. We won't have incorrect positions or duplicate positions in one record) Explain how to find them out? [7 points]

```
cat nba.txt l grep -E "(<[^>]*>/?){2,}" l wc -l
```

3.
-k, --key=KEYDEF
sort via a key; KEYDEF gives location and type
KEYDEF is F[.C][OPTS][,F[.C][OPTS]] for start and stop position, where F is a field number and C a character position in the field; both are origin 1, and the stop position defaults to the line's end. If neither -t nor -b is in effect, characters in a field are counted from the beginning of the preceding whitespace. OPTS is one or more single-letter ordering options [bdfgiMhnRrV], which override global ordering options for that key. If no key is given, use the entire line as the key. Use --debug to diagnose incorrect key usage.

therefore, use sort -k5 -r
pts per game, mins per game, draft year, name of the player
Then use $awk '{print $4, $5, $6, $7, $8}', this command only prints from the 4th column to the 8th column, eliminating the first four columns.
Therefore, the command should be
sort -k5 -r l awk '{print for (i=4;i<=8;i++)print $i}'

(3) A news reporter wants to write a story about best scorers drafted in year 2003. Explain how to generate a sorted list of those players (the output should contain the points-per-game, minutes-per-game, draft year and the name of the player. Output is sorted by points-per-game in descending order). [7 points]
(There is more space to write on next page)

Problem 4 One Day of System Administrators [45 Points]

Suppose you are working for the student union to manage their web server. The manager assigned you some jobs to be done. Use tools learned from CS 35L to complete the following tasks:

(1) In the Christmas party held by the students union, there is a lottery session. Instead of using lottery tickets, you want to implement a program to randomly pick up a winner from the registered email address. Implement a **python script** lottery.py to do so. The script should taken an argument which is a text file containing email addresses of all the registered guests, if no argument is given, it will try to read from "guest-emails.txt". Each line is an email address following a tab and the guest name. (e.g. johnsmith@ucla.edu   John Smith).

The program will output a congratulation message:

```
$ ./lottery.py guest-emails
"Congratulation, John Smith! You won the lottery! Our secret gift has been sent to your email: johnsmith@ucla.edu"
```

Part of the code is given. Please fill in the #TODO part of the code (See next page). If you want to change the given code, please comment out the original code by prefixing a "#" and writing your code right below it.

If you cannot figure out writing the program, try to briefly explain your idea. [15 Points]

Quick python refs:
Suppose s is a string, s = "abc\n"

s.find(a)	returns the index of the first occurrence of substring a in s e.g. s.find("b") # returns 1
s.strip()	remove the preceding and trailing space characters e.g. s.strip() # returns "abc"

```
#
# lottery.py
# Find the lucky winner by random picking from registration records

import random, sys

def random_pick(tuples):

    # Random pick an item from the list tuples
    return random.choice(tuples)

def read_tuples(filename):

    # return a list of tuples by parsing the file
    # Each element of the list is a tuple of two strings: (email, name)
    # e.g. ("johnsmith@ucla.edu", "John Smith")
    tuples = []
    f = open(filename, 'r')
    for line in f.readlines():
        #TODO parse each line into tuples

    f.close()
    return tuples

def print_message(tup):

    #tup is the tupple containing winner's email and name
    # e.g. ("johnsmith@ucla.edu", "John Smith")
    #TODO print the congratulation message

def main():
    if(len(sys.argv) > 1):
        print "Too few arguments"
    #TODO complete the main function
```

```
lottery.py

"guest-emails.txt"

johnsmith@ucla.edu

John Smith

f = open(filename, 'r')
for line in f.readlines():
   #TODO parse each line into tuples
   index_tab = line.find("    ")
   tu = (line[0:index_tab], line[index_tab + 1:])
   tuples.append(tu)
def print_message(tup):
   print "Congratulation, {0}! You won the lottery! Our
secret gift has been sent to your email:
{1}".format(tup[0], tup[1])

def main():
   if(len(sys.argv) > 2): # argv[0] is the file name
print "Too Many arguments"

#TODO complete the main function
     if (len(sys.argv) == 1):


people = read_tuples("guest-emails.txt")
     else:

people = read_tuples(sys.argv[1])
```

(2) Some students reported that their personal info in the student union website was stolen. It is suggested that there may be some security hole in your code. You are asked to examine the existing code on your server. Try to analyze the following code snippets and point out if there is a security hole. If yes, please point out the security hole and explain briefly how to exploit it and how to prevent it. If you cannot find the hole in this piece of code, please briefly explain what are other possible reasons why students lost their privacy information. [10 points]

```
# the following code is for the "retrieve SSN number" function
# the function will return a string contains the SSN
# the code is written in python
username = response.POST["username"]
#get the username from the http POST request
#e.g. http://service.com/?username=xxxxxx, and username will be "xxxxxx"
string query = "SELECT SSN from SSN-table WHERE username = " + username
dbm = DatabaseManager() #initialize a data base manager
dbm.makeQuery(query)    #issue the query to the database
```

User input is always dangerous. The attackers can write some SQL commands as the username, and the python program will concatenate the command to the sql query, which allows the attacker to query information he is not authorized to request. We can prevent this incident by adding some senity checks on the user data. For example, if there are any special characters in the string, we will deny the request and don't make the databse query.

(3) One senior system administrator argues that the security hole may be resulted from your C code using function memcpy. However, you are confident that your work won't be attacked because of the way you write the Makefile. Briefly explain what you did to prevent the attack. [5 Points]

We used the stack protector of gcc, known as -fstack-protector. Such operations cause some performance overhead, since extra code will be added to check for buffer overflows, such as stack smashing attacks. This is done by adding a guard variable to functions with vulnerable objects. This includes functions that call alloca, and functions with buffers larger than 8 bytes.

(4) One of your daily tasks is to generate a visiting count histogram by analyzing the http access log. You need to count the occurrence of a list of urls to know which part of your site is particularly popular, e.g.:
http://studentunion.com/class/
http://studentunion.com/recreation/
…
Assume we already have the urls extracted from the log, and the following function performs the counting job.

```
/* Count the occurences of urls in url_from_log
 * url_list is the list of candidate urls you need to count
 * counters[i] indicates the occurrence of url_list[i]
 * m and n is the size of url_list and url_from_log
 * correspondingly
 */
void count_urls(char** url_list, int m
                char** url_from_log, int n
                int* counters, ) {
    int i, j;
    for(i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            if (strcmp(url_list[i], url_from_log[j]) == 0)
                counters[j]++;
}
```

Your server has 8 cores, and you want to exploit the multicores to speed up your job. Please implement the multithread version of count_urls. If you cannot provide the complete code, please provide a design of the program to illustrate your ideas.
[15 Points]

Modified version:

```
#include <pthread.h>

static int* counters_ptr_global;

struct tparam {
   int start;
   int end;
};

typedef struct tparam Tparam;

void* thread(void* vargp) { /* Thread routine
*/
    Tparam* param = (Tparam*)(vargp);
    int start = param->start;
    int end = param->end;

    for (int j = start; j <= end; j++) {


if (strcmp(url_list[i], url_from_log[j]) == 0)

counters_ptr_global[j]++;

}
    return NULL;
}
```

```
void count_urls(char** url_list, int m,
            char** url_from_log, int n,
int* counters) {
counters_ptr_global = counters;
int i, j;
pthread_t tids[8];
    for (i = 0; i < m; i++){
      for (j = 0; j < 8; j++) {
Tparam param;
param->start = j * 8;
param->end = j * 9 > n ? n : j * 9;
            pthread_create(&tids, NULL, thread,(void*)
(&param))
      }

      for (int i = 0; i < 8; i++) {
      pthread_join(tids[i], NULL);
      }
    }
}
```