CS35L – Software Construction Lab
Fall 2013
Final Exam

Instructor/TA – Paul Eggert/Sharath Gopal
Date: 9th December 2013
Time: 8AM to 11AM
Total points – 100
Duration – 3 hours

Student Name: ______________________________ UID: _________________________________

Instructions:

I) 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 back of each sheet. The space provided on the front of each sheet, should have just the answer.

II) This examination is open-book, open-notes. Any printed or hand-written materials are permitted. **Electronic devices are NOT permitted.**

III) 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.

1) [5 points] Regular Expression - In each of the following cases, you are given 2 regular expressions R1 and R2. You have to say if they generate the same set of strings or not. If they generate the same set of strings then say YES **and** provide an example of a string generated by them. Otherwise say NO **and** provide an example of a string which is generated by one, but not by the other expression. (The symbol '|' means the regular expression before it OR after it. The symbol '*' means zero or more occurrences of the preceding regular expression. Parentheses '()' are used for grouping regular expressions.)

- R1 is **(a|b)*** R2 is **(a*b*)***

  (a|b)* is same as (a*b*)*

- R1 is **a|ba** R2 is **(a|b)a**

  a|ba is NOT same as (a|b)a

  a|ba: a, ba
  (a|b)a: aa, ba

- **R1 is (ab|a)*a R2 is a(ba|a)***

  (ab|a)*a a(ba|a)* are same

  (ab|a)*a: aba, abababa, a, aaaaa
  a(ba|a)*: aba, abababa, a, aaaaa

2) [2 points] Shell - What is the use of the $PATH environment variable?
(Example - PATH=/usr/local/cs/bin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin)

$PATH specifies the directories in which executable programs are located on the machine that can be started without knowing and typing the whole path to the file on the command line.

PS: How to add PATH:
export PATH=$PATH:/user/Desktop/.......
However, PATH is gone when you restart the program

3) [3 points] Shell - You just created a shell script named buildwords.sh in your HOME folder. When you try to run the script as ./buildwords.sh, you get an error

bash: ./buildwords.sh: Permission denied

What could be a possible reason for this? How would you go about fixing this issue?

You do not have the execution permission of buildwords.sh
chmod 755 buildwords.sh
OR: chmod u+x
so that everyone can read and execute this file

4) [3 points] Make - You have written a 'C' program in 'srot.c' that does not use any special libraries. Write a simple Makefile that has two rules. The first is the default rule that generates the executable file 'srot', and the other is a rule to clean the build (remove the 'srot' executable).

```
CC = gcc
make: srot.c
    $(CC) srot.c -o srot

clean:

rm srot
```

```
5.
#!/bin/bash
if gcc --version I grep -F "4.7.2"
then
  echo 'CC = gcc' > Makefile
  echo 'srot: srot.o' >> Makefile
  echo '\t$(CC) -o srot srot.c' >> Makefile
  echo 'clean:' >> Makefile
  echo '\trm srot.o srot' >> Makefile
else
  echo 'This software requires gcc 4.7.2 version'
fi
```

5) [8 points] Shell - You just wrote a srot.c that compiles only with gcc version 4.7.2. You are giving this code to your friend, but you don't know if he has the right version of gcc. Therefore, in addition to the file srot.c, you have to give him a shell script named 'configure.sh' that generates a 'Makefile' only if the 4.7.2 version of gcc exists on his system. The output of the command

```
gcc --version
```

```
gcc (GCC) 4.7.2
Copyright (C) 2012 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

can be used for getting the version number. If the right version is found, then your script should create a Makefile that you wrote for the previous question 4. This should be created on the fly. If the required version is not found on your friend's system then it should print a message 'This software requires gcc 4.7.2 version', and NOT generate the Makefile.

System calls are special type of function that:
Used by user-level processes to request a service from the kernel
Changes CPU's mode from user mode to kernel mode to enable more capabilities
Is part of the kernel of OS
Verifies that the user should be allowed to do the requested action and then does the action (kernel performs the operation on behalf of the user)
Is the only way a user program can perform privileged operations

A process can be authorized to perform dangerous operations via system calls
If the kernel code does NOT perform all required safety check

6) [2 points] System calls - What are system calls? Why should you use them judiciously?

7) [4 points] System calls - Here is a simple implementation of putchar(), where the write() system call is used to write a character to STDOUT

File: myputchar.c

write() is a system call that slows the function significantly

```
#include<unistd.h>
int putchar(int c)
{
   unsigned char cC = (unsigned char)c;
   int rv = write(1, &cC, 1);
   return rv == 1 ? c : EOF;
}
```

Discuss a drawback of the putchar() implementation in the context of a program using it to write a huge file to STDOUT.

8) [10 points] You are required to allocate/deallocate memory to a 3-dimensional array using malloc() and free() functions. You are given a “pointer-to a-pointer-to a-pointer” as shown below. The sizes for the 3 dimensions are 100, 200 & 300 respectively. Write 'C' code that allocates memory using malloc(), followed by code to deallocate the same memory using free().

float ***M;

```
float ***M;
int i,j,k;
M=(float ***) malloc (sizeof(float **) * 100);
for(i=0;i<200;i++)
{M[i]=(float **)malloc (sizeof(float *)*200);
for(j=0;j<300;j++)
{ M[i][j]=(float *) malloc(sizeof(float)*300); }
}
for(i=0;i<200;i++)
{
 for(j=0;j<300;j++)
{ free(M[i][j]); }
free(M[i]);
}
free(M);
```

9) [5 points] System administration – You are an administrator of the SEASNET servers and a new student named Alice has applied for an account on lnxsrv01.seas.ucla.edu. How would you go about creating an account for her on the server? What instructions would you give her, so that she can login to her account to use the SEASNET computing facilities? How would she run applications like Firefox on the server? List the commands that you would use for each.

9.
Log in as root

Add user:
useradd <USERNAME>
Set password:
passwd <USERNAME>

After this the new user's entry is stored in '/etc/passwd' file

By default 'useradd' command creates a user's home directory under /home
To specify the home directory of creating this user:
useradd -d /some/NewDirectory <USERNAME>
Then the directory is created inside NewDirectory

For Alice:
To login, type:
ssh <YOUR_USERNAME@lnxsrv01.seas.ucla.edu>

To run an application, simply type the name of the application:
firefox

10) [5 points] Crypto - Alice wants to send a secret document that can be read only by Bob. The document contains sensitive information such as credit card numbers, SSN, etc. Trudy is able to sniff the communication link that is there between Alice and Bob. How can Alice make use of public-key cryptography to make it difficult for Trudy to get the actual contents of the document? Explain this at a conceptual level.

Let Bob generate an ssh key pair and send his public key to Alice so that Alice can use it to encrypt the secret document. A file encrypted with public key can only be decrypted with private key, so only Bob can decrypt the file.

If Bob does not have a keypair, let him generate an ssh key pair:
ssh-keygen
First specify the file to store the key (default: ~/.ssh/id_rsa)
NOTICE that if we do NOT use the default location, we need to specify the absolute path: /u/cs/ugrad/me/someFile instead of ~/someFile !!

Then set the name and passphrase for the key

now the *.pub file is generated in the corresponding directory, Bob can share this with anyone who want to send him secret documents

11) [6 points] The make-log.txt of a student contains the following output of her/his ray tracer

```
time ./srt 1-test.ppm >1-test.ppm.tmp && mv 1-test.ppm.tmp 1-test.ppm
real   0m49.987s
user   0m46.851s
sys    0m0.020s
```

You have to write a single line command (pipes are ok), that reads in a make-log.txt file and extracts just the number of seconds in the "real" time. For example, in the above case it should print just 49.

```
#!/bin/bash
cat $1|grep real|grep -Po "(?<=m).*(?=\.)"
```

12) [6 points] C Programming - The signature of the gets() function is given below

```
char *gets(char *s);
```

The documentation of gets() says the following
"reads a line from STDIN into the buffer pointed to by s until either a terminating newline or EOF, which it replaces with '\0'."

You are an administrator of a system and have written and installed software that uses gets() to get input from a user on the system, and store it in an array (pointed to by 's') on the stack. Discuss any vulnerability that has been introduced in your software. How can the user exploit it?

Potential buffer overflow
The function may overrun the buffer's boundary and overwrite adjacent memory locations
data is put in a fixed-length buffer

If the function does not check array bounds, attackers simply supply an input with length greater than buffer can hold.

13) [12 points] You are given starter code to sort a set of spheres. The spheres have to be sorted in ascending order according to radii. If there is a tie, then they have to be sorted in ascending order according to their center's distance to the origin. The comments in the starter code provide more description and direction. Rewrite the main() and CompareSpheres() to sort the spheres. If you are reusing some of the starter code, then you don't have to rewrite it.

```
#include<stdio.h>
#include<math.h>
enum { SPHERE_COUNT = 20 };

typdef struct
{
   float X, Y, Z;
} Point3D;

typedef struct
{
   Point3D center;
   float radius;
} Sphere;

float Distance(Point3D a, Point3D b)
{
   //Returns the distance between 2 points
   return sqrt(pow(a.X – b.X, 2) + pow(a.Y – b.Y, 2) + pow(a.Z – b.Z, 2));
}

int CompareSpheres(void *r1, void *r2)
{
   //You need to implement this compare function
}

int main(int argc, char** argv)
{
   Sphere *s = (Sphere*) malloc(sizeof(Sphere) * SPHERE_COUNT);
   if(!s) {
            fprintf(stderr, “Cannot allocate memory!\n”); exit(-1);
   }

   // Assume the following function call loads values into the spheres in 's'.
   // Your task is just to call qsort and write the compare function
   LoadSpheres(s);

   //Replace the following call with your qsort() call
   qsort(....);

   free(s);
   return 0;
}
```

```
float Distance(Point3D a, Point3D b)
{

//Returns the distance between 2 points


return sqrt(pow(a.X – b.X, 2) + pow(a.Y – b.Y, 2)

+ pow(a.Z – b.Z, 2));
}

int CompareSpheres(void *r1, void *r2) {
   return(*(struct Sphere*)r2->radius) - (*(struct Sphere*)r1-
radius);
}

int main()
{
//...

qsort(s,SPHERE_COUNT,sizeof(Sphere),CompareSpheres);
}
```

14) [6 points] What are program exceptions? Explain how you would handle them in python? (You have to provide some sample code and talk about execution flow when an exception occurs, and also when it doesn't occur).

An exception is an error that happens during execution of a program.
When the error occurs, python generate an exception that can be handled,
which avoids your program from crashing
A finally clause is optional and is executed before leaving the try statement

Python:
exceptions are caught in try blocks and handled in except blocks.
If an error is encountered, a try block code execution is stopped and
transferred down to the except block

```
try:
print 1/0
except ZeroDivisionError:
print "Error: Division by zero"
finally:
print "End of program"
```

15) [15 points] You are given a file that contains the requests by customers for 1800 phone numbers, and it is listed "first come first served". An example request file can be as follows:

CALLUCLA
CALLUCSD
CALLUCSF
...etc

where the first request is for a number 1800-CALLUCLA. The number-to-letters mapping on a phone is given as follows : 2(ABC), 3(DEF), 4(GHI), 5(JKL), 6(MNO), 7(PQRS), 8(TUV), 9(WXYZ). You have to write a **python script** that takes this request file as argument, and outputs only the "serviceable" requests. A request is not "serviceable" if the phone number, that it maps to, has already been requested previously in the list (Example: Since CALLUCSF and CALLUCSD map to the same number, you have to print only the one that was requested earlier). You have to assume that there are a huge number of requests in the file. Hence, your accesses/lookups should be efficient.

```
import sys
def translatePhoneNum(phone_num):
    s=""
    for c in phone_num:
        if c in "ABC":
            s+='2'
        elif c in "DEF":
            s+='3'
        elif c in "GHI":
            s+='4'
        elif c in "JKL":
            s+='5'
        elif c in "MNO":
            s+='6'
        elif c in "PQRS":
            s+='7'
        elif c in "TUV":
            s+='8'
        elif c in "WXYZ":
            s+='9'
        else:
                break
    return s
if __name__=="__main__":
    if(len(sys.argv)!=2):
        print "Error: Invalid number of arguments!"
        sys.exit(1)
    filename=sys.argv[1]
```

```
if __name__=="__main__":
    if(len(sys.argv)!=2):
        print "Error: Invalid number of arguments!"
        sys.exit(1)
    filename=sys.argv[1]

    validPhones=set()
    f=open(filename,'r')
    content=f.readlines()
    for line in content:
        phone_num=translatePhoneNum(line)
        #Input line has different size from output phone number
        #subtract 1 because readline() reads a line and append a '\n'
        if(len(phone_num)!=len(line)-1):
            continue

        else:
            validPhones.add(phone_num)

    for n in validPhones:
        print n
```

Benefits of using VCS:
Have a backup of files by storing older versions
Ability to restore previous versions

16) [2 points] GIT - State any 2 benefits of using a version control system.

Files are shared among multiple computers
Everyone in a dev team has an independent version of source code, thus can work
independently
Thus developers can work in parallel
Copy/Merge/Undo changes
Tagging and branching
Maintaining multiple versions of code

17) [2 points] GIT - State any 2 advantages of a distributed version control system, when compared to a centralized version control system.

Benefits of DVCS over CVCS
Perform actions when working offline:
Committing new changesets can be done locallyEverything but pushing and pulling can be done without an
internet connection
Faster speed: only needs to access the hard drive, not a remote server.
Easy merging and branching

18) [4 points] GIT - You are working on code that is being tracked by GIT on your local machine. You just made some modifications to files that were existing before, and also added a new header and a source file (foo.h and foo.c).

- How would you know what changes have been made since the last commit?

  a) git status

- How would you commit the changes that you just made to your code?

  First, run "git add FILE", where "FILE" is the name of file(s) you want to commit
  Or run "git add ." to add all files under the current repository

  Second, run "git commit -m "COMMIT_MESSAGE" ", where the quoted "COMMIT_MESSAGE"
  is the commit message you write

- How would you refer to the commit you just made, in the future?

  Use the commit's SHA hash (or "commit id")
  Run "git log" to get the SHA of this commit