Assignment 1 (120 points)
Fall 2014
Shlomo Hershkop
CIS 542
Out: Sept 9, 2014
Due: Midnight Sept 23

This assignment is broken into 2 sections. A system section which is geared towards implanting some part of a systems solution and a programming section which is meant to re-enforce basic programming concepts covered in class. Please attempt the assignment as early as possible and reach out when you need help…

In general for programming assignments you will need to submit code files and a “readme.txt” file which has your name and email on top and summarizes each of the included files which you are submitting. For the system part a short write up with screen shots (when applicable) is required to show work.

System Assignment 1:

Goal: Learn VM and SSH basics

1) Get a basic virtualization system installed and working on your local system. I had suggested virtual box from oracle, but you are free to adopt the instructions for vmware.

    1) Download the latest ubuntu server iso file. If you don’t know what an iso file is please look it up on wikipedia etc.

    2) create a virtual machine with the iso so that you can run a server version of ubuntu. During the install process it will ask you for user/password make sure you make a note of it. Also it should ask you about setting up ssh which you should choose during the install process.

    3) once you have a running vm, please install any necessary packages needed to accomplish your assignments. For example to install the “nmap” program, you would login to the vm, “sudo su” to become root. and then “apt-get install nmap” to install the nmap program from the central repository. You will need an internet connection for this to work.

       Please write up your experiences in setting this up. Also please browse on your hard drive to the actual virtual machine and look at how much space the files are actually taking up vs the allocated space you setup during the install process. Make sure to understand what you are looking at.

    4) log into the vm and type “ifconfig”. you should see which ip address the vm has been allocated. This will be needed for later steps

5) run “lsb_release -a” and see what it tells you about your system. run “df -h”, “uptime”, “ps -aux” and look at the ouput. Please lookup each of these commands so you are familiar with them.

6) On your main box you will need to ssh (download ssh or equivalent if necessary) and ssh to your vm

7) please look at https://help.ubuntu.com/community/SSH/OpenSSH/Keys for example on setting up ssh keys on your vm.
for windows you will need to download cygwin so you can generate your ssh key pairs on your main machine. Basically you will need to login into your vm to get the local ip address (from earlier step with “ifconfig”) and then copy the public portion over to the vm

8) make sure you can ssh over to the vm with the ssh key pair (ie no password for login). You will need to use ssh-add to add your key into memory before trying to ssh over.

   please write up your experiences on this and note any difficulty you encountered. Also note any resources used to solve problems etc.

2) Programming Assignment 1

This is a very quick review of c and implementation of c programming. Remember to include a readme.txt file with your information in addition to the makefile and .c and .h files.

1) Very basic: Learning Makefile:

   - Stanford CS Education Library has a nice short tutorial on Makefiles. See Unix Programming Tools, Section 2, available at http://cslibrary.stanford.edu/107/.

   - The manual for make is available at http://www.gnu.org/software/make/manual/make.html. Reading the first couple of chapters should be sufficient for basic understanding.

   There are a few rules that we will follow when writing makefiles in this class:

   1) Always provide a "clean" target to remove the intermediate and executable files.
   2) Compile with gcc rather than cc.
   3) Use "-Wall" option for all compilations.
   4) Use "-g" option for compiling and linking.

2) Simple C program:

   Write a program in C which prints out a welcome message and asks the users to enter their name from standard input. Once you read in their name, you want to print it out backwards. Also print out how many characters in their name. Please be aware that a user can have unlimited middle names. Make sure your program works with 0,1,2..n middle names.

3) Simple logic and random

   write a simple c program which randomly chooses a number between 0-100.
   Print the starting time. Then please prompt the user to guess the number. The user has 10 tries to guess the correct number. At each time, you need to tell them if the correct number is very less, less, more, much more (can use 10 as a cutoff to decide how close, i.e. within 10 is close, else not close)
   If the user guesses the number end otherwise after 10 turns, the user looses.
   Print out the ending time and then calculate the time difference. The time difference should be encoded as a function call which takes start and end time and returns the difference in seconds.

4) Prime numbers:

Here is a play program using prime numbers
I am providing some code, the rest you should be able to create on your own.

You will be creating step3.c, prime.h, and prime.c and create appropriate rules in your makefile

prime.h:
```
#include<stdio.h>
#include<stdlib.h>

int is_prime(int n);
```

prime.c
```
int is_prime(int n){
 int k, limit;

 if (n==2)
          return 1;
 if (n % 2 == 0)
          return 0;
 limit = n/2;
 for( k = 3; k <= limit; k+=2 )
          if ( n % k == 0)
                    return 0;
return 1;
}
```

add comments to the above code to explain how the program works.

In addition complete step3.c by adding a main method so that it passes in a specific number to the function above and gives you a listing and total number of primes up and including the specified number. Example setting the number to 999 should loop (for loop) and will show all 168 primes that are less than 1000. feel free to extend the code any way you want.
Note: Please remember, only pass .c files to your gcc command.