CSCE 212H, Spring 2010, Matthews
Lab01 Assignment: Representation of Floats
Assigned: January 21 Due: January 27

Manton Matthews

January 21, 2010

1 Overview

The purpose of this assignment is to become more familiar with bit-level representations of numners and bitwise operators. We will begin by doing a few “unix calastenics” on the Linux workstations. Then you will be ask to answer a series of questions and solve a series of programming “exercises.” The portions of this assignment are:

1. Unix Calastenics
   - ssh
   - changing your password
   - making a subdirectory
   - changing permissions
   - copying files
   - submitting assignments
2. Bitwise operations in C
3. IEEE floating Point representation
4. BigExp float Implementations

2 Logistics

All labs will be submitted electronically using “dropbox,” i.e. https://dropbox.cse.sc.edu. Note the ’s’ in https. It stands for secure http. Login there using your Unix password. Remember you have a Unix password and an Engineering domain password. If there are any corrections or modifications to assignments these will be sent out via email and posted to the website.

2.1 CSE Secure Website

The Department Website has information on the Departement academic programs, faculty etc. For this lab we are interested in the secure web site https://www.cse.sc.edu that contains information that we would like to make available to students but not to the rest of the world. To get access the this site you must supply your ”Unix” login-name and password. Usually the name is the same as your name in the ”engr” domain, but your password will be different. Go to this site (https://www.cse.sc.edu **Note carefully the ”s” in https!**) and login.

Passwords

Your initial password is set to a function of your student ID. This is not secure at all and your should change your password when you first login. A good password:

- is long enough to make it difficult to guess by brute force,
- is not an english word (This would be succeptible to dictionary attacks.), and
- contains a mixture of uppercase, lowercase and special characters.

For detailed instructions on changing your password see https://www.cse.sc.edu.

2.2 Important Links on CSE Secure Site

There are a large number of important links on the secure site and I encourage you to explore for your self. However, there are a few that I would like to point to for help in our labs.

- Dropbox/Center - for assignments
- GradeBook Center - our version of ”blackboard for assignemnts/grades”
- Password
- CSE Computing: Lab policy, Workstations, Host Usage
- Remote Access
- Networked Resources

3 Start Up Instructions

Start by signing on to your engineering domain account. You then need to run the program SSH, Secure SHell, (Start ⇒ Programs ⇒ CSE_Apps?). This program allows for secure connections to remote computers. All transmissions including the initial username and password are encrypted for security. This particular program is already available in the labs but it is also available for downloading and installing on your personal machines from The secure CSE site FTP (upper right menu) ⇒ Windows. More details on SSH may be found using google.

As part of making a connection your need to specify the remote computer. This can be done by giving an IP address or a domain qualified name. For our purposes we will be connecting to the Linux machines in 1D41 or 1D43. The machine names are are given below.

You will also need to give your Unix "login name."

During our labs it will be a good idea for us to logon to different machines. To simplify this I have listed a collection of workstation names.

```
 IP ADDRESS      HOSTNAME                     MACHINE TYPE

Lab 1D39
129.252.130.165   apollon.cse.sc.edu               # Linux lab Precision 380
129.252.130.166   demeter.cse.sc.edu               # Linux lab Precision 380
129.252.130.167   eos.cse.sc.edu                   # Linux lab Precision 380
129.252.130.168   moira.cse.sc.edu                 # Linux lab Precision 380
129.252.130.169   clotho.cse.sc.edu                # Linux lab Precision 380
129.252.130.170   gorgon.cse.sc.edu                # Linux lab Precision 380
129.252.130.171   pluto.cse.sc.edu                 # Linux lab Precision 380
129.252.130.172   helios.cse.sc.edu                # Linux lab Precision 380
129.252.130.175   kronos.cse.sc.edu                # Linux lab Precision 380
...               ...                              # Sun Machines are older

Lab 1D45
129.252.130.147   cerberus.cse.sc.edu            # Linux lab Optiplex 270
129.252.130.148   hermes.cse.sc.edu                # Linux lab Optiplex 270
129.252.130.149   hebe.cse.sc.edu                  # Linux lab Optiplex 270
129.252.130.153   minerva.cse.sc.edu            # Linux lab Precision 650
129.252.130.154   thetis.cse.sc.edu                # Linux lab Precision 650
129.252.130.155   erida.cse.sc.edu                 # Linux lab Precision 650
129.252.130.156   embedded.cse.sc.edu              # Linux lab Precision 650
129.252.130.157   tartarus.cse.sc.edu             # Linux lab Precision 370
129.252.130.158   circe.cse.sc.edu                 # Linux lab Precision 370
129.252.130.159   metis.cse.sc.edu                 # Linux lab Precision 370
129.252.130.160   kalypso.cse.sc.edu               # Linux lab Precision 370
129.252.130.161   hestia.cse.sc.edu                # Linux lab Precision 370
129.252.130.162   chimera.cse.sc.edu               # Linux lab Precision 380
129.252.130.163   hades.cse.sc.edu                 # Linux lab Precision 380
129.252.130.164   skylla.cse.sc.edu                # Linux lab Precision 380
```

You may save this session so that future connections to this machine are easy.

3.1 Basic Unix Commands

Basic Commands Dealing with the Hierarchy	
ls	list the contents of the current working directory
cd dir	change working directory to the directory dir, .. means parent
pwd	prints the current working directory
mkdir file	creates a new directory
date	prints the date
Simple File Commands	
mv file1 file2	renames file1 with name file2 (old file2 destroyed)
rm file	removes file
cp file1 file2	makes a copy of file1 and names it file2
cat file	catenate file, writes file to the standard output (equivalent of MS-DOS type)
vim file	invokes the vim editor on ”file” (vi)
pico file	invokes the pico editor on ”file”
gcc prog.c	invokes the c compiler on the program prog.c creates a.out(cc)
less file	page through the file one screen at a time(more)
sort file	sorts the file
grep pattern files	print the lines that match the pattern from the files
wc file	wordcount counts the characters, words and lines in file
find path options actions	search subdirectory path for files
e.g. find . -name ”*.c” -print	search current subtree print filenames ending in “.c”
Printer Commands	
lpr file	prints file
lpr -Pln	print file on the printer named ‘ln’
a2ps file	**converts an ascii text file to Postscript for printing on laser printers
Assistance Commands	
man cmd	prints on-line manual entry for the command cmd
man -k subject	gives list of manual entries related to the keyword subject
Process Manipulating Commands	
ps	print list of current processes
kill -sig pid	send the signal sig to the process with id pid
history	prints the hsitroy list of commands
alias newname=oldCommand	gives an oldCommand a new name (usually shorter)
exit	terminate shell
Communication Commands	
mail [user]	read mail [send mail to user]
ssh machine	create a secure connection to another machine
sftp machine	secure file transfer to/from another machine
Environmental Commands	
passwd	allows the user to change his password
quota	display disk usage and quotas
du	disk usage for a specific subtree
df	print the amount of free disk space
env	print the environment variables
set	print the values of the shell variables
Who is Who Commands	
who	who is logged in currently
w	who is logged in and what are they doing

4 Permissions

Each file in the UNIX system has a set of permissions associated with it describing how and by whom it may be accessed. There are there types of access to files (1) read access, (2) write access and (3) execute access. For instance a user could have read permission, but not write or execute permission. To encode these 3 bits of information requires a 3-bit binary string or an octal digit. In this encoding the READ permission bit is the highest bit, the WRITE permission bit is the next highest, and the execute permission bit is the lowest. For instance read and execute but not write permission would be represented by 101 = 5 octal.

For each file there are 3 sets of permissions each represented as an octal digit. These are the permissions for the owner, the permission for the other users in the owner's group, and the permission for all other users. To see the permission information associated with a file one can use the command "ls -l" as in the example below. The output of the ls - l translates the permissions into a symbolic representation of 10 characters. The leading character tells what kind of file this is. Here 'd' means a directory and '-' means an ordinary file. There are other possible types and the reader can see the ls manual page for details.

There is a leading character and then 3 groups of 3 characters. The first group is for the owner and the next for the owner's group and finally for all other users. Using regular expression notation each of the groups is "(r | -) (w | -) (z | -)" which of course means the first character is 'r' or '-', the second 'w' or '-' and finally the third 'x' or '-'. If there is an 'r' then there is read permission, and conversely if there is a dash where the 'r' would be then there is not read permission.

To set permissions explicitly one can use the 'chmod' command. Chmod can use a 3-digit octal number.

```
% cd sample
% ls -l testfile
-rw-rw-rw- 1 matthews 37 Dec 9 18:49 testfile
% chmod 750 testfile
% ls -l testfile
-rwxr-x— 1 matthews 37 Dec 9 18:49 testfile
```

Setting up your 211 Directory

To create a subdirectory for your CSCE 211 files use the command "mkdir 211" and then use "chmod 0700 211" to protect this subdirectory. Then use the command "cd 211" to move into this subdirectory and "mkdir Lab1" to create a subdirectory for the files of Laboratory 1. Finally move into the Lab1 directory with "cd Lab1."

Copying the set up Files

There is a directory "/class/csce212-501" that contains files for the various labs during the course. There are several subdirectories for files, most notably Examples and Lab1. To get started copy the file showBytes.c using the command

```
cp /class/csce212-501/Code/show_bytes.c .
```

The “.” in the previous command refers to the current working directory. After copying the file you can use “ls” to show the files in the current directory and verify that the copy command worked.

5 IEEE Floating Point Representation

The 32 bit IEEE floating point representation has three components:

1. the sign bit (1 bit)
2. the exponent (8 bits)
3. the mantissa (23 bits with implied leading 1 for 24 bits)

$Question$ 1 Using showBytes what is the hex presentation of the float 212.0?

$Question$ 2 Without using show bytes what is the representation of -212.0?

$Question$ 3 Without using show bytes what is the representation of 212.75? *Hint .75 = .5 + .2* = $2^{-1} + 2^{-2}$

$Question$ 4 Find the smallest float x such that $x + 1 \;==\; x$.

$Question$ 5 How many different positive floats have exponent field =0xF0?

The next few questions ask for you to supply a function. To write these functions you will need to use the “bit-wise and” to create masks and shifts to rearrange extracted fields. As an example the code

```
unsigned int ui, msb;
union {
  float f;
  unsigned int ui;
} un;
un.f = f;
msb = un.ui && 0xFF000000 >> 24;
msb = msb && 0x000000FF;
```

extracts the most significant byte. The second masking is required because the sign is replicated.

$Question$ 6 Write a function int getSign(float f) that will extract the sign bit

$Question$ 7 Write a function int ExtractExp(float f) that will extract the exponent field and convert it to the actual exponent.

6 Math library

There are libraries that are available to use in constructing programs. In C under UNIX these are typically stored in /usr/lib. For instance the command

```
tar tvf /usr/lib/libm.a
```

There are several notes on this command

- tar - which stands for tape archive, is the library maintainer for UNIX
- the arguments (options) are tvf
- the option t prints the table of contents of the archive
- the option f indicates that the archive is in the file whose name follows the f
- in the name of the library "libm.a" the m indicates the math library.

The UNIX manual prints documentation on commands. "man sin" prints information on "sin."

> *Question* 8 Write a program that will print a "sin" table of 10 values equally spaced between 0 and pi/2.

7 Big Exponent Floats

The purpose of this assignment is to become more familiar with the IEEE 754 floating point standard. In this we will implement a new 32 bit float that uses 15 exponent bits and only 16 bits for the fraction.

In C we will use the declaration

```
typedef BigExp  unsigned int;

BigExp b,f,g;
```

In the latter sections of the lab we will be implementing functions for addition and multiplication. For built-in types these would use the operators '+' and '*', but for our purposes we will just use the function prototypes given below:

```
BigExp addBigExp( BigExp x, BigExp y);
BigExp multBigExp( BigExp x, BigExp y);
```

8 BigExp Questions

There are certain representational and demographics questions first.

1. What is the BIAS?

2. What is the representation in hex of the smallest positive normalized BigExp number?

3. What is its value (i.e. the value represented by the previous BigExp (hex) representation?

4. What is the representation in hex of the largest normalized BigExp number?

5. What is its value, i.e. the value represented by the previous BigExp (hex) representation?

6. $\cdots$ the smallest positive BigExp? (denormal)

7. What is its value, i.e. the value represented by the previous BigExp (hex) representation?

8. $\cdots$ the largest positive denormal BigExp?

9. What is its value, i.e. the value represented by the previous BigExp (hex) representation?

10. What is the biggest gap between two BigExp numbers?

11. What is the representation of - ∞?

12. Are there more BigExps or ints?

13. Are there more BigExps or floats?

9 Implementation of Extract Functions

Write a collection of functions to extract the components of a BigExp number.

- int getSign(BigExp f)
- int getExp(BigExp f) (careful on denormals).
- int getMantissa(BigExp f)

10 Implementation of Addition Function

Write a function addBigExp that will take two BigExp's as arguments and produce there sum, implementing the default rounding.

```
BigExp addBigExp( BigExp x, BigExp y);
```

Notes on this implementation include:

- Alg: shift to align exponents, add, adjust exponent, round to even
- Overflow → ∞
- denormal addition must be handled as a special case.
- Handling signed BigExps must be done in cases, i.e. separate code.

11 Implementation of Multiplication Function

```
BigExp multBigExp( BigExp x, BigExp y);
```