CS35L Labs/Homework's

# Homework 2

## BuildWords

#!/bin/bash

# deletes at top of HTML File and to first english word

sed '/<!DOCTYPE/,/<\/font><\/td>/d' |

# removes lines after last table line to end of html file

sed '/<\/table>/,/<\/html>/d' |

# removes all the english words by deleting first part of column

sed '/<tr>/,/<\/td>/d' |

# removes carriage returns

sed '/\r\n/d' |

# translates all uppercase to lowercase

tr '[:upper:]' '[:lower:]' |

# removes all </td> phrases with new lines

sed 's/<\/td>/\n/g' |

# replaces ` with '

sed s/\`/\'/g |

# replace commas with new lines

sed 's/\,/\n/g' |

# delete any html stuff

sed 's/<[^>]\*>//g' |

# deletes any blank spaces

tr -d '[:blank:]' |

# separates words in a sentence

sed 's/ /\n/g' |

# deletes illformed haiwaiian words

sed '/-/d' |

# deletes any misspelled hawaiin language

tr -cs "pk\'mnwlhaeiou" '[\n\*]' |

# sorts the dictionary and remove duplicates

sort -u |

# remove empty lines

sed '/^$/d'

## Lab.log

1. Set Locale to C:

export LC\_ALL='C'

2. Sort words file:

sort -u /usr/share/dict/words >words

3. Download webpage

wget https://web.cs.ucla.edu/classes/fall18/cs35L/assign/assign2.html

4. Reading differences

tr -c 'A-Za-z' '[\n\*]' < assign2.html:

Checks through each character for a non alphabetical character

and replaces it with a return character

tr -cs 'A-Za-z' '[\n\*]' < assign2.html:

Does the same as above but only displays a single empty line

instead of multiple.

tr -cs 'A-Za-z' '[\n\*]' < assign2.html | sort:

Does same as above command but sorts the results alpa

tr -cs 'A-Za-z' '[\n\*]' < assign2.html | sort -u:

Sorts again but only a single instance of each word is displayed

tr -cs 'A-Za-z' '[\n\*]' < assign2.html | sort -u | comm - words:

Takes sorted file from above and compares it to words list, showing

all three columns

tr -cs 'A-Za-z' '[\n\*]' < assign2.html | sort -u | comm -23 - words:

Only ouputs lines that are unique to the sorted file.

5. Downloading Hawaiian webpage

wget http://mauimapp.com/moolelo/hwnwdseng.htm

6. Building Script

emacs buildwords

7. Make script executable

chmod +x buildwords

8. Build function to use with html file

cat hwnwdseng.htm | ./buildwords > hwords

9. Check Spelling

a. Checking for mispelled English

cat assign2.html | tr -cs 'A-Za-z' '[\n\*]' | tr '[:upper:]' '[:lower:]'

| sort -u | comm -23 - words >misenglish

wc -w misenglish

OUTPUT: 38 misenglish

Meaning there are 38 mispelled english words

b. Checking for mispelled Hawaiian

cat assign2.html | tr -cs "pk\'mnwlhaeiou" '[\n\*]' | tr '[:upper:]' '[:lower:]'

| sort -u | comm -23 - hwords > mishawaiian

wc -w mishawaiian

OUPUT: 204 mishawaiian

Meaning there are 204 mishawaiin words

c. Mispelled english, but not hawaiian

cat misenglish | tr -cs "pk\'mnwlhaeiou" '[\n\*]' | sort -u | comm -12 - hwords

>misenglishhawaiian

Results:

halau

po

wiki

d. Misspelled Hawaiian, but not english

cat mishawaiian | tr -cs 'A-Za-z' '[\n\*]' | sort -u | comm -12 - word

s > mishawaiianenglish

## Sameln

#Luke Jung CS35L 904-982-644

#!/bin/bash

#set the dir variable to input of function

dir=$1

#sets IFS variable to new line

IFS=$'\n'

#declares arr variable to be array

declare -a arr

let count=0

#creates 2 variables ot hold different files,

#one for hidden files which should be added first

#and one for normal files

d=`ls $dir | sort`

hd=`ls -a $dir | grep '^\.' | sort`

#for loop to add hidden files first

for hfile in $hd

do

#if file is a link

if [ -L "$dir/$hfile" ]

then

:

#if file is not a file or readable print error

elif [ ! -f "$dir/$hfile" ]

then

:

elif [ ! -r "$dir/$hfile" ]

then

echo "can't read $hfile"

else

#if passes all the tests, add to array and count

arr[$count]="$dir/$hfile"

let count=count+1

fi

done

for file in $d

do

if [ -L "$dir/$file" ]

then

:

elif [ ! -f "$dir/$file" ]

then

:

elif [ ! -r "$dir/$file" ]

then

echo "can't read $file"

else

arr[$count]="$dir/$file"

let count=count+1

fi

done

#for loop to test through each subject in array if same

for (( i=0; i<$count; i++ ))

do

for (( j=i+1; j<$count; j++ ))

do

#cmp function that checks first 2 variables

cmp -s "${arr[$i]}" "${arr[$j]}"

if [ $? -eq 0 ] #if cmp == 0. means they're the same

then

#ln -fp links the first object to the other and deletes it

ln -fP "${arr[$i]}" "${arr[$j]}"

arr[$j]=arr[$i]

fi

done

done

# Homework 3

## Hw3.txt

Q3. If we run this script on an empty file like /dev/null,

we get the following error

Traceback (most recent call last):

File "./randline.py", line 70, in <module>

main()

File "./randline.py", line 64, in main

sys.stdout.write(generator.chooseline())

File "./randline.py", line 34, in chooseline

return random.choice(self.lines)

File "/usr/lib64/python2.7/random.py", line 274, in choice

return seq[int(self.random() \* len(seq))]

# raises IndexError if seq is empty

IndexError: list index out of range

This happens because the randline.py file assumes that the file

is not empty, and thereby chooses some random line non 0.

However since the file has no lines, it accesses a line that's

not there, giving undefined behavior

Q4. When running this command using python3, we get the error

File "randline.py", line 65

except IOError as (errno, strerror):

^

SyntaxError: invalid syntax

Python3 uses tuple assignment with unpacking which means it assigns values

all in one line together, but since there's nothing to assign, it provides

a sytnax error and won't run at all.

## Lab3.txt

1. Grab Coreutils sig and GNU keyring

wget ftp://ftp.gnu.org/gnu/coreutils/coreutils-8.29.tar.xz.sig

wget https://ftp.gnu.org/gnu/gnu-keyring.gpg

gpg --verify --keyring ./gnu-keyring.gpg coreutils-8.29.tar.xz.sig

Didn't get any problems but it said that my key is not certified or

that the signature belongs to it's owner.

2. Download and install coreutils

wget ftp://ftp.gnu.org/gnu/coreutils/coreutils-8.29.tar.xz

tar xf coreutils-8.29.tar.xz

mkdir coreutilsInstall

cd coreutils-8.29

./configure --prefix=/u/ee/ugrad/luke/cs35l/lab/lab3/coreutilsInstall

make

make install

3. Reproduce bug

mkdir test

touch .

touch ..

touch .foo

touch foo

ls -aA shows all 4 files, not just the 2

4. Patch the bug

cd coreutils-8.29

emacs patchfile.diff

-paste patch into this file

patch -p1 <patchfile.diff

5. make

This command remakes files that have ls in it and dir files

because these are the files that the patch fixed.

Q1. The patch slightly decreases performance because it has to check

for multiple options when parsing through the options for the ls command.

Q2. I would probably watch out for making sure

the new ls command works as intended and as

quickly as before the patch. Other applications

shouldn't be affected because we didn't change

anything to their code.

## Randline.py

#!/usr/bin/python

import random, sys

from optparse import OptionParser

class randline:

def \_\_init\_\_(self, filename):

f = open(filename, 'r')

self.lines = f.readlines()

self.len = len(self.lines)

f.close()

def sample(self, count):

j = random.sample(self.lines, count)

for index in range(count):

sys.stdout.write(j[index])

return 0

def write\_repeat(self, count):

for index in range(count):

sys.stdout.write(random.choice(self.lines))

return 0

def repeat(self):

while True :

sys.stdout.write(random.choice(self.lines))

return 0

def filelength(self):

return self.len

def main():

version\_msg = "%prog 2.0"

usage\_msg = """%prog [OPTION]... FILE

Output randomly selected lines from FILE."""

input\_range = False

output\_range = False

numlines = 0

parser = OptionParser(version=version\_msg,

usage=usage\_msg)

parser.add\_option("-n", "--numlines",

action="store", dest="numlines", default=1,

help="output NUMLINES lines (default 1)")

parser.add\_option("-i", "--input-range",

action="store", dest="range",

help="treat each number LO through HI as input line")

parser.add\_option("-r", "--repeat", action="store\_true", dest="repeat", default=False,

help="output ines can be repeated")

options, args = parser.parse\_args(sys.argv[1:])

try:

numlines = int(options.numlines)

if numlines > 1:

output\_range = True

strrange = options.range

if strrange is not None:

input\_range = True

except:

parser.error("invalid NUMLINES: {0}".

format(options.numlines))

if numlines < 0:

parser.error("negative count: {0}".

format(numlines))

if not input\_range:

if len(args) != 1:

parser.error("wrong number of operands")

input\_file = args[0]

generator = randline(input\_file)

lo\_input = 0

hi\_input = generator.filelength()

try:

if int(options.repeat) and not input\_range:

if output\_range:

generator.write\_repeat(numlines)

else:

generator.repeat()

if input\_range:

if len(args) >= 1:

parser.error("wrong number of operands")

indexlist = strrange.split("-")

lo\_input = int(indexlist[0])

hi\_input = int(indexlist[1])

count = hi\_input-lo\_input

items = list(range(lo\_input,hi\_input+1))

random.shuffle(items)

if int(options.repeat):

if output\_range:

for index in range(numlines):

print(random.choice(items))

else:

while True:

print(random.choice(items))

if output\_range and not options.repeat:

if hi\_input <= numlines:

numlines = hi\_input

for index in range(numlines):

print(items[index])

if not output\_range and not options.repeat:

for index in range(count+1):

print(items[index])

else:

if hi\_input <= numlines:

numlines = hi\_input

generator.sample(numlines)

except IOError as err:

sys.parser.error("I/O error({0}): {1}".

format(errno, strerror))

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Homework 4

## Lab4.diff

2018-11-02 Luke Hyun Jung <luke@lnxsrv09.seas.ucla.edu>

\* lib/timespec.h (timespec\_cmp):

Use if statements to avoid using subtraction

To not have integer overflow with large numbers away

--- lib/timespec.h 2005-09-21 23:05:39.000000000 -0700

+++ lib/timespec.h 2018-11-02 22:58:04.758886000 -0700

@@ -45,8 +45,12 @@

static inline int

timespec\_cmp (struct timespec a, struct timespec b)

{

- int diff = a.tv\_sec - b.tv\_sec;

- return diff ? diff : a.tv\_nsec - b.tv\_nsec;

+ if(a.tv\_sec < b.tv\_sec)

+ return -1;

+ if(a.tv\_sec == b.tv\_sec)

+ return 0;

+ if(a.tv\_sec > b.tv\_sec)

+ return 1;

}

# if ! HAVE\_DECL\_NANOSLEEP

## Lab4.txt

Lab 4 Log

1. Grab the tar file from the html file

$ wget

2. Unzip tar file

$ xvzf coreutils-with-bug.tar.gz

3. Create a new directory to install coreutils bugged version

$ mkdir coreutils

$ ./configure --prefix=/u/ee/ugrad/luke/cs35l/hw4/coreutils

4. Install the program using the make command

$ make //when using this command, I got an error when installing.

5. Grab the patch file

$ wget https://web.cs.ucla.edu/classes/fall18/cs35L/assign/coreutils.diff

6. Edit the patch file deleting the ~

$ emacs coreutils.diff

7. Patch the core-utils file

$ mv coreutils.diff coreutils-with-bug/coreutils.diff

$ patch -p0 < coreutils.diff

The patch fixed the problem that two functions had the same name,

so we changed the name of one of them.

8. Make and install patched version

$ make

$ make install

9. Check for bug

$ tmp=$(mktemp -d)

$ cd $tmp

$ touch -d '1918-11-11 11:00 GMT' wwi-armistice

$ touch now

$ sleep 1

$ touch now1

$ TZ=UTC0 ~/cs35l/hw4/coreutils-with-bug/src/ls -lt

--full-time wwi-armistice now now1cd

Doing this, I got the output

-rw-r--r-- 1 luke eeugrad 0 1918-11-11 11:00:00.000000000 +0000 wwi-armistice

-rw-r--r-- 1 luke eeugrad 0 2018-11-03 05:02:37.810876412 +0000 now1

-rw-r--r-- 1 luke eeugrad 0 2018-11-03 05:02:34.291816447 +0000 now

which is wrong, because the newest is supposed to be first

10. Use GDB to find the bug

Before using GDB, we have to create the files in the temp file to

make sure we can access them. We can do this by copying the first

7 commands from #9.

$ tmp=$(mktemp -d)

$ cd $tmp

$ touch -d '1918-11-11 11:00 GMT' wwi-armistice

$ touch now

$ sleep 1

$ touch now1

$ TZ=UTC0 ~/cs35l/hw4/coreutils-with-bug/src/ls -lt --full-time

wwi-armistice now now1

Then we can access GDB and go directly to the ls executable

$ gdb ~/cs35l/hw4/coreutils/bin/ls

The first thing we must find is the function that sorts

the time of each file. We can check all the functions of a

file using the info command

(gdb) info functions

Looking through these functions, I found a function called

sort\_files which seemed promising, so I put a breakpoint there.

(gdb) break sort\_files

Now we can run the full ls function

(gdb) r -lt --full-time wwi-armistice now now1

Starting program: /w/home.18/ee/ugrad/luke/cs35l/hw4/coreutils/bin/ls

-lt --full-time wwi-armistice now now1

(gdb) s

2975 func = sort\_reverse ? rev\_cmp\_mtime : compare\_mtime;

Once I got to the this part of the step funciton, I saw that the function

calls either rev\_cmp\_mtime or compare\_mtime, so I needed to trace thse too

(gdb) break compare\_mtime

(gdb) r -lt --full-time wwi-armistice now now1

Starting program: /w/home.18/ee/ugrad/luke/cs35l/hw4/coreutils/bin/ls -lt

--full-time wwi-armistice now now1

(gdb) s

49 return diff ? diff : a.tv\_nsec - b.tv\_nsec;

(gdb) info locals

diff = 3

diff = <optimized out>

(gdb) s

49 return diff ? diff : a.tv\_nsec - b.tv\_nsec;

(gdb) info locals

diff = -1139919939

diff = <optimized out>

I reduced some of the output because it was unecessary,

but I realized going through the compare\_mtime function,

that we were getting large negative values for the

difference, meaning we have an integer overflow.

This comes from the following:

timespec\_cmp (struct timespec a, struct timespec b)

47 {

48 int diff = a.tv\_sec - b.tv\_sec;

49 return diff ? diff : a.tv\_nsec - b.tv\_nsec;

50 }

Therefore, we're getting an overflow from the int diff variable,

messing up

our order.

11. Avoid subtraction to fix the bug

a. Go to the location of the timespec\_cmp function

$ cd ~/cs35l/hw4/coreutils-with-bug/lib

b. Copy file to make diff file

cp timespec.h/../timespec.h

cd ..

c. edit file to avoid subtracting

static inline int

timespec\_cmp (struct timespec a, struct timespec b)

{

if(a.tv\_sec < b.tv\_sec)

return -1;

if(a.tv\_sec == b.tv\_sec)

return 0;

if(a.tv\_sec > b.tv\_sec)

return 1;

}

I used if statements because it was the easiest and more readable way to

avoid any subtraction.

Then make a patch.diff file

diff -u lib/timespec.h timespec.h >patch.diff

Then make a changelog file

emacs

C-x 4 a

2018-11-02 Luke Hyun Jung <luke@lnxsrv09.seas.ucla.edu>

\* lib/timespec.h (timespec\_cmp):

Use if statements to avoid using subtraction

to not have integer overflow with large numbers away

cp ChangeLog lab4.diff

cat patch.diff >> lab4.diff

Also have to change the patch file to make the paths point directly

to the lib/timespec.h file

12. Patch file and check to see if it worked

$ patch -p0 <lab5.diff

patching file lib/timespec.h

$ make

[luke@lnxsrv09 ~]$ tmp=$(mktemp -d)

[luke@lnxsrv09 ~]$ cd $tmp

[luke@lnxsrv09 /tmp/tmp.iXx5vPuPQT]$ touch -d

'1918-11-11 11:00 GMT' wwi-armistice

[luke@lnxsrv09 /tmp/tmp.iXx5vPuPQT]$ touch now

[luke@lnxsrv09 /tmp/tmp.iXx5vPuPQT]$ sleep 1

[luke@lnxsrv09 /tmp/tmp.iXx5vPuPQT]$ touch now1

[luke@lnxsrv09 /tmp/tmp.iXx5vPuPQT]$ TZ=UTC0 ~/cs35l/hw4/

coreutils-with-bug/src/ls

-lt --full-time wwi-armistice now now1

-rw-r--r-- 1 luke eeugrad 0 2018-11-03 18:14:37.170158198

+0000 now1

-rw-r--r-- 1 luke eeugrad 0 2018-11-03 18:14:36.166141042

+0000 now

-rw-r--r-- 1 luke eeugrad 0 1918-11-11 11:00:00.000000000 +

0000 wwi-armistice

Now this works correctly and the bug is fixed.

7. Check on SEASnet

First, must go to the home directory

So, when we do the function in our home directory, we get the year to be 2054, a year far into the future instead of 1918. I'm guessing this must do with truncating the bits when they're added, since we're getting a high value positive while it's supposed to be a high value negative. This could be due to losing the MSB through truncation.

## Sfrob.c

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

/\*

\* function that checks memory and returns if pointer

\* is null

\*/

void checkmemory(void \*input)

{

if(input == NULL)

{

fprintf(stderr, "Memory Error: %d\n", errno);

exit(1);

}

}

/\*

\* function that checks input and returns if pointer

\* has a error

\*/

void checkinput(void \*input)

{

if(ferror(input))

{

fprintf(stderr, "IO Error: %d\n", errno);

exit(1);

}

}

/\*

\* The encoding given in the spec is just an xor of 42

\*/

char decoder(const char a)

{

return (a^42);

}

/\*

\* function that goes through two strings to compare

\* if one is larger or less

\*/

int frobcmp(char const\* a, char const\* b)

{

while(\*a != ' ' && \*b != ' ')

{

if(\*a == '\0')

a++;

if(\*b == '\0')

b++;

if (\*a == ' ' || decoder(\*a) < decoder(\*b))

return -1;

if (\*b == ' ' || decoder(\*a) > decoder(\*b))

return 1;

a++;

b++;

}

return 0;

}

/\*

\* function that changes casts the input from a single

\* pointer string to a double

\*/

int frbcmp(const void\* a, const void\* b)

{

const char\* input1 = \*(const char\*\*)a;

const char\* input2 = \*(const char\*\*)b;

return frobcmp(input1, input2);

}

/\*

\* print function to print out output and checks if input

\* has an error

\*/

void print(const char\* input)

{

for(;;)

{

putchar(\*input);

checkinput(stdout);

if(\*input++ == ' ')

return;

}

}

int main(void)

{

//initialize variables

char \*input, \*\*output;

size\_t file\_size = 10, buffer\_size = 0, line\_num = 0, line\_size = 0, i;

int eof = feof(stdin);

int space;

//create input array and malloc size of the array

input = (char\*) malloc(sizeof(char) \* file\_size);

checkmemory(input);

//continue to parse through string until gets to end of phrase

while(!eof)

{

char c;

c = getchar();

checkinput(stdin);

space = c == ' ';

eof = feof(stdin);

//chekcs for space and if line size is 0 to prase

if(space && ! line\_size)

continue;

//if our buffer gets to our file size, increase size

if(buffer\_size == file\_size)

{

file\_size \*= 2;

input = (char\* ) realloc(input, sizeof(char) \* file\_size );

checkmemory(input);

}

//continue to add characters to array

if(!eof)

{

input[buffer\_size++] = c;

line\_size++;

if (! space)

continue;

}

//if doesn't pass other cases...

else

{

//if the size doesn't go up, the input is over

//need to just return program

if(buffer\_size == 0)

{

free(input);

return 0;

}

//if there is no space at the end of buffer, add one

if(input[buffer\_size-1] != ' ')

{

input[buffer\_size++] = ' ';

}

//if end of phrase, breaks out of while loop

if(line\_size == 0)

break;

}

//increment the output size and reset line\_size

line\_num ++;

line\_size = 0;

}

//output array

output = (char\*\*) malloc(sizeof(char\*) \* line\_num);

checkmemory(output);

//for loop to add characters to array and are

//separated by spaces for each line

size\_t j;

char\* line = input;

for( i = 0, j = 0; i < buffer\_size; i++)

{

if(input[i] == ' ')

{

output[j++] = line;

line = input + i + 1;

}

}

//quick sort the array

qsort(output, line\_num, sizeof(char\*), frbcmp);

for(i = 0; i < line\_num; i ++)

{

print(output[i]);

}

//clear variables at end

free(output);

free(input);

return 0;

}

# Homework 5

## Lab.txt

To write the different tr files, I used the

knowledge from the previous homework of decoding

and saw that it would mostly be the same process.

I also had to look up the argv parameter in int main

to understand where the options get stored

3. Measuring System Calls

I used a c function to generate a large file with

random text.

#include <stdlib.h>

#include <unistd.h>

int main() {

char c;

for (int i = 0; i < 5000000; i++) {

c = rand()% (127 - 32) + 32;

write(1, &c, 1);

}

}

a)

To copy one file to another I used the command

strace -c ./tr2b 'a' 'x' < output.txt > output1

strace -c ./tr2u 'a' 'x' < output.txt > output2

So here the buffered one did a much better job

b)

strace -c ./tr2u 'a' 'x' < output.txt

buffered one still did better, but took a bunch more write calls than

copying

4. Testing time

time ./tr2b 'a' 'z' < output.txt > output1

real 0m0.250s

user 0m0.155s

sys 0m0.006s

time ./tr2u 'a' 'z' < output.txt > output2

real 0m11.429s

user 0m1.542s

sys 0m9.792s

## sfrob script

#!/bin/bash

normal="\000-\377"

frob="\52\...

export LC\_ALL='C'

if [ $# -eq 1 ]

then

tr "$frob" "$normal" | sort -f | tr "$normal" "$frob"

fi

if [ $# -eq 0 ]

then

tr "$frob" "$normal" | sort | tr "$normal" "$frob"

fi

## sfrobu.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <ctype.h>

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>

int toUpper;

/\*

\* function that checks memory and returns if pointer

\* is null

\*/

void checkmemory(void \*input)

{

if(input == NULL)

{

fprintf(stderr, "Memory Error: %d\n", errno);

exit(1);

}

}

/\*

\* function that checks input and returns if pointer

\* has a error

\*/

int checkinput(int num)

{

if(num < 0)

{

fprintf(stderr, "IO Error: %d\n", errno);

exit(1);

}

return 0;

}

/\*

\* The encoding given in the spec is just an xor of 42

\*/

char decoder(const char a)

{

if(toUpper)

{

return toupper(a^42);

}

return a^42;

}

/\*

\* function that goes through two strings to compare

\* if one is larger or less

\*/

int frobcmp(char const\* a, char const\* b)

{

while(\*a != ' ' && \*b != ' ')

{

if(\*a == '\0')

a++;

if(\*b == '\0')

b++;

if (\*a == ' ' || decoder(\*a) < decoder(\*b))

return -1;

if (\*b == ' ' || decoder(\*a) > decoder(\*b))

return 1;

a++;

b++;

}

return 0;

}

/\*

\* function that changes casts the input from a single

\* pointer string to a double

\*/

int frbcmp(const void\* a, const void\* b)

{

const char\* input1 = \*(const char\*\*)a;

const char\* input2 = \*(const char\*\*)b;

return frobcmp(input1, input2);

}

/\*

\* print function to print out output and checks if input

\* has an error

\*/

void print(char\* input)

{

for (;;)

{

int num = write(STDOUT\_FILENO, input, 1);

checkinput(num);

if (\*input++ == ' ')

return;

}

return;

}

int main(int argc, const char\* argv[])

{

//initialize variables

char \*input, \*\*output;

size\_t file\_size, buffer\_size = 0, line\_num = 0, line\_size = 0, i;

int eof = 0, space, num;

struct stat st;

//create input array and malloc size of the array

num = fstat(STDIN\_FILENO, &st);

checkinput(num);

file\_size = st.st\_size+1;

input = (char\*) malloc(sizeof(char) \* file\_size);

checkmemory(input);

toUpper = 0;

if(argc == 2)

{

const char\* arg1 = argv[1];

if(strcmp(arg1,"-f") == 0)

{

toUpper = 1;

}

}

if(argc > 2)

{

fprintf(stderr, "IO Error: %d\n", errno);

exit(1);

}

//continue to parse through string until gets to end of phrase

while(!eof)

{

char cbuf[2];

num = read(STDIN\_FILENO, &cbuf, 1);

checkinput(num);

char c = cbuf[0];

space = c == ' ';

eof = (!num);

//checks for space and if line size is 0 to prase

if(space && ! line\_size)

continue;

//if our buffer gets to our file size, increase size

if(buffer\_size == file\_size)

{

file\_size \*= 2;

input = (char\* ) realloc(input, sizeof(char) \* file\_size );

checkmemory(input);

}

//continue to add characters to array

if(!eof)

{

input[buffer\_size++] = c;

line\_size++;

if (! space)

continue;

}

//if doesn't pass other cases...

else

{

//if the size doesn't go up, the input is over

//need to just return program

if(buffer\_size == 0)

{

free(input);

return 0;

}

//if there is no space at the end of buffer, add one

if(input[buffer\_size-1] != ' ')

{

input[buffer\_size++] = ' ';

}

//if end of phrase, breaks out of while loop

if(line\_size == 0)

break;

}

//increment the output size and reset line\_size

line\_num ++;

line\_size = 0;

}

//output array

output = (char\*\*) malloc(sizeof(char\*) \* line\_num);

checkmemory(output);

//for loop to add characters to array and are

//separated by spaces for each line

size\_t j;

char\* line = input;

for( i = 0, j = 0; i < buffer\_size; i++)

{

if(input[i] == ' ')

{

output[j++] = line;

line = input + i + 1;

}

}

//quick sort the array

qsort(output, line\_num, sizeof(char\*), frbcmp);

for(i = 0; i < line\_num; i ++)

{

// print(output[i]);

print(output[i]);

}

//clear variables at end

free(output);

free(input);

return 0;

}

## Tr2b.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <errno.h>

void error()

{

fprintf(stderr, "Incorrect Input... %d\n", errno);

exit(1);

}

int main(int argc, const char\* input2[])

{

if(argc !=3)

{

error();

}

const char\* arg1 = input2[1];

const char\* arg2 = input2[2];

if(strlen(arg1) != strlen(arg2))

{

error();

}

size\_t len = strlen(arg1);

for(int i = 0; i < len-1; i ++)

{

for(int j = i+1; j < len; j ++)

if(arg1[i] == arg1[j])

error();

}

int eof = feof(stdin);

int change;

while(!eof)

{

char c;

c=getchar();

eof = feof(stdin);

if(eof)

break;

for(int i = 0; i < len; i ++)

{

if(c == arg1[i])

c = arg2[i];

}

putchar(c);

}

}

# Homework 6

## Log.txt

Multithread Performance Lab:

1. Check sort

which sort

/usr/local/cs/bin/sort

sort --version

sort (GNU coreutils) 8.30

2. Create input file containing 2^24 doubles

od -An -tf -N $(($4\*(2\*\*24))) </dev/urandom > random.txt

Recreate file but make new space with tr command and delete

first line

od -An -tf -N $(((2\*\*24)\*8)) </dev/urandom |

tr -s ' ' '\n' |

sed '/^$/d' > random.txt

wc -l <num.txt

16777217

3. Check Performance

a. No options

time -p sort -g random.txt > /dev/null

real 66.60

user 340.10

sys 1.03

b. 1 thread

time -p sort -g --parallel=1 random.txt > /dev/null

real 324.92

user 324.23

sys 0.67

c. 2 thread

time -p sort -g --parallel=2 random.txt > /dev/null

real 170.02

user 325.56

sys 0.55

d. 4 threads

time -p sort -g --parallel=4 random.txt > /dev/null

real 102.46

user 350.87

sys 0.73

3. 8 threads

time -p sort -g --parallel=8 random.txt > /dev/null

real 63.86

user 361.84

sys 0.91

## Makelog.txt

make clean check

rm -f \*.o \*.tmp 1-test.ppm 2-test.ppm 4-test.ppm 8-test.ppm srt srt.tgz

gcc -std=gnu11 -g -O2 -Wall -Wextra -Wno-unused-parameter -c -o main.o main.c

gcc -std=gnu11 -g -O2 -Wall -Wextra -Wno-unused-parameter -c -o raymath.o raymath.c

gcc -std=gnu11 -g -O2 -Wall -Wextra -Wno-unused-parameter -c -o shaders.o shaders.c

gcc -std=gnu11 -g -O2 -Wall -Wextra -Wno-unused-parameter -o srt main.o raymath.o shaders.o -lm -lpthread

## main.c (pthreads program

//

// main.c

// srt

//

// Created by vector on 11/2/10.

// Copyright (c) 2010 Brian F. Allen.

#include "raymath.h"

#include "shaders.h"

#include <stdio.h>

#include <stdlib.h>

#include <assert.h>

#include <math.h>

#include <pthread.h>

static double dirs[6][3] =

{ {1,0,0}, {-1,0,0}, {0,1,0}, {0,-1,0}, {0,0,1}, {0,0,-1} };

static const int opposites[] = { 1, 0, 3, 2, 5, 4 };

static void

add\_sphereflake( scene\_t\* scene, int sphere\_id, int parent\_id, int dir,

double ratio, int recursion\_level )

{

sphere\_t\* parent = &scene->spheres[parent\_id];

sphere\_t\* child = &scene->spheres[sphere\_id];

/\* start at parents origin \*/

mul( child->org, dirs[dir], (1.+ratio)\*parent->rad );

add( child->org, child->org, parent->org );

child->rad = parent->rad \* ratio;

copy( child->color, parent->color );

child->shader = parent->shader;

scene->sphere\_count++;

}

static int

recursive\_add\_sphereflake( scene\_t\* scene, int parent\_id, int parent\_dir,

int sphere\_id, int dir,

int recursion\_level, int recursion\_limit )

{

const double ratio = 0.35;

add\_sphereflake( scene, sphere\_id, parent\_id, dir, ratio, recursion\_level );

if( recursion\_level > recursion\_limit )

{

return sphere\_id + 1;

}

/\* six children, one at each cardinal point \*/

parent\_id = sphere\_id;

sphere\_id = sphere\_id + 1;

for( int child\_dir=0; child\_dir<6; ++child\_dir )

{

/\* skip making spheres inside parent \*/

if( parent\_dir == opposites[child\_dir] ) continue;

sphere\_id = recursive\_add\_sphereflake( scene, parent\_id, parent\_dir,

sphere\_id, child\_dir,

recursion\_level + 1,

recursion\_limit );

}

return sphere\_id;

}

static scene\_t

create\_sphereflake\_scene( int recursion\_limit )

{

scene\_t scene;

Vec3 color;

sphere\_t\* sphere;

init\_scene( &scene );

// Pantone UC Gold 122

add\_light( &scene, 2, 5, 0, 0.996, 0.733, 0.212 );

// Pantone UCLA Blue (50,132,191)

add\_light( &scene, -5, 3, -5, 0.196, 0.517, 0.749 );

int max\_sphere\_count = 2 + powl( 6, recursion\_limit + 2 );

scene.spheres = realloc( scene.spheres,

max\_sphere\_count\*sizeof( sphere\_t ) );

if( !scene.spheres )

{

fprintf( stderr, "Failed to get memory for sphereflake. aborting.\n" );

exit( -1 );

}

// sphere = &(scene.spheres[0]);

// set( sphere->org, -0.5, -1.0, 0 );

// sphere->rad = 0.75;

// set( color, 0.85, 0.25, 0.25 );

// copy( sphere->color, color );

// sphere->shader = mirror\_shader;

/\* center sphere is special, child inherent shader and color \*/

sphere = &(scene.spheres[0]);

scene.sphere\_count++;

set( sphere->org, 0, -1, 0 );

sphere->rad = 0.75;

set( color, 0.75, 0.75, 0.75 );

copy( sphere->color, color );

sphere->shader = mirror\_shader;

recursive\_add\_sphereflake( &scene,

0, /\* parent is the first sphere \*/

-1, /\* -1 means no dir, make all children \*/

1, /\* next free sphere index \*/

2, /\* starting dir \*/

0, /\* starting recursion level \*/

recursion\_limit );

return scene;

}

static void

free\_scene( scene\_t\* arg )

{

free( arg->lights );

arg->light\_count = 0;

free( arg->spheres );

arg->sphere\_count = 0;

}

/\*\*\*\*\*\*

\* Constants that have a large effect on performance \*/

/\* how many levels to generate spheres \*/

enum { sphereflake\_recursion = 3 };

/\* output image size \*/

enum { height = 131 };

enum { width = 131 };

/\* antialiasing samples, more is higher quality, 0 for no AA \*/

enum { halfSamples = 4 };

/\*\*\*\*\*\*/

/\* color depth to output for ppm \*/

enum { max\_color = 255 };

/\* z value for ray \*/

enum { z = 1 };

int nthreads;

scene\_t scene;

float scaled\_color[width][height][3];

void \* thread\_pixels( void \*thread\_num)

{

int col = \*(int \*) thread\_num;

Vec3 camera\_pos;

set( camera\_pos, 0., 0., -4. );

Vec3 camera\_dir;

set( camera\_dir, 0., 0., 1. );

const double camera\_fov = 75.0 \* (PI/180.0);

Vec3 bg\_color;

set( bg\_color, 0.8, 0.8, 1 );

const double pixel\_dx = tan( 0.5\*camera\_fov ) / ((double)width\*0.5);

const double pixel\_dy = tan( 0.5\*camera\_fov ) / ((double)height\*0.5);

const double subsample\_dx

= halfSamples ? pixel\_dx / ((double)halfSamples\*2.0)

: pixel\_dx;

const double subsample\_dy

= halfSamples ? pixel\_dy / ((double)halfSamples\*2.0)

: pixel\_dy;

/\* for every pixel \*/

for( int px=col; px<width; px+=nthreads )

{

const double x = pixel\_dx \* ((double)( px-(width/2) ));

for( int py=0; py<height; ++py )

{

const double y = pixel\_dy \* ((double)( py-(height/2) ));

Vec3 pixel\_color;

set( pixel\_color, 0, 0, 0 );

for( int xs=-halfSamples; xs<=halfSamples; ++xs )

{

for( int ys=-halfSamples; ys<=halfSamples; ++ys )

{

double subx = x + ((double)xs)\*subsample\_dx;

double suby = y + ((double)ys)\*subsample\_dy;

/\* construct the ray coming out of the camera, through

\* the screen at (subx,suby)

\*/

ray\_t pixel\_ray;

copy( pixel\_ray.org, camera\_pos );

Vec3 pixel\_target;

set( pixel\_target, subx, suby, z );

sub( pixel\_ray.dir, pixel\_target, camera\_pos );

norm( pixel\_ray.dir, pixel\_ray.dir );

Vec3 sample\_color;

copy( sample\_color, bg\_color );

/\* trace the ray from the camera that

\* passes through this pixel \*/

trace( &scene, sample\_color, &pixel\_ray, 0 );

/\* sum color for subpixel AA \*/

add( pixel\_color, pixel\_color, sample\_color );

}

}

/\* at this point, have accumulated (2\*halfSamples)^2 samples,

\* so need to average out the final pixel color

\*/

if( halfSamples )

{

mul( pixel\_color, pixel\_color,

(1.0/( 4.0 \* halfSamples \* halfSamples ) ) );

}

/\* done, final floating point color values are in pixel\_color \*/

scaled\_color[px][py][0] = gamma( pixel\_color[0] ) \* max\_color;

scaled\_color[px][py][1] = gamma( pixel\_color[1] ) \* max\_color;

scaled\_color[px][py][2] = gamma( pixel\_color[2] ) \* max\_color;

/\* enforce caps, replace with real gamma \*/

for( int i=0; i<3; i++)

scaled\_color[px][py][i] = max( min(scaled\_color[px][py][i], 255), 0);

}

}

return NULL;

}

int

main( int argc, char \*\*argv )

{

nthreads = argc == 2 ? atoi( argv[1] ) : 0;

if( nthreads < 1 )

{

fprintf( stderr, "%s: usage: %s NTHREADS\n", argv[0], argv[0] );

return 1;

}

scene = create\_sphereflake\_scene( sphereflake\_recursion );

/\* Write the image format header \*/

/\* P3 is an ASCII-formatted, color, PPM file \*/

printf( "P3\n%d %d\n%d\n", width, height, max\_color );

printf( "# Rendering scene with %d spheres and %d lights\n",

scene.sphere\_count,

scene.light\_count );

pthread\_t threads[nthreads];

int \* threadID = malloc(sizeof(int) \* nthreads);

int num, i;

for(i = 0; i < nthreads; i ++)

{

threadID[i] = i;

num = pthread\_create(&threads[i], NULL, thread\_pixels, (void \*)(threadID + i));

if (num)

{

fprintf(stderr, "Thread couldn't be created.");

exit(1);

}

}

for(i = 0; i < nthreads; i ++)

{

num = pthread\_join(threads[i], NULL);

if(num)

{

fprintf(stderr, "Couldn't join thread");

return 1;

}

}

for (int px = 0; px < width; px++)

{

for (int py = 0; py < height; py++)

{

/\* write this pixel out to disk. ppm is forgiving about whitespace,

\* but has a maximum of 70 chars/line, so use one line per pixel

\*/

printf( "%.0f %.0f %.0f\n",

scaled\_color[px][py][0], scaled\_color[px][py][1], scaled\_color[px][py][2]);

}

printf( "\n" );

}

free\_scene( &scene );

if( ferror( stdout ) || fclose( stdout ) != 0 )

{

fprintf( stderr, "Output error\n" );

return 1;

}

free(threadID);

return 0;

}

# Homework 7

## Randmain.mk

CFLAGS = -O2 -g3 -Wall -Wextra -march=native -mtune=native -mrdrnd

randlibsw.so:

gcc $(CFLAGS) -fPIC -c randlibsw.c -o randlibsw.o

gcc $(CFLAGS) -shared -o randlibsw.so randlibsw.o

randlibhw.so:

gcc $(CFLAGS) -fPIC -c randlibhw.c -o randlibhw.o

gcc $(CFLAGS) -shared -o randlibhw.so randlibhw.o

randmain:

gcc $(CFLAGS) -c randcpuid.c -o randcpuid.o

gcc $(CFLAGS) -c randmain.c -o randmain.o

gcc $(CFLAGS) -ldl -Wl,-rpath=$(PWD) randmain.o randcpuid.o -o randmain

# Homework 8

## Hw.txt

1. If the other team was still monitoring bits, we would still be secure because each bit is encrypted with the public key and when passed over it can be decrypted using the public key. Since the other team doesn't have access to the private key, they would never be able to see what we're sending.

If the keyboard was tapped, then there's a higher risk of getting attacked because they can see the passwords we use for the pass phrase and the actual password we created when we started up the server. But if there

isn't password authentication, then they couldn't access the server regardless.

For usb, if they had access to our .ssh folder, they could extract our private key and decrypt all of our information.

2. The --verify command doesn't really verify that I created the file because the command doesn't check the owner of the file but checks to see if the public key and signature match each other. So if the data was modified and signed by a different public key, then it would still be "verified" making it not the creator.

We could solve this by having the public key be posted somewhere to verify that it matches the signature, and if it doesn't then it means that the signature was tampered with.

## Lab.txt

First thing we did was follow the piazza post about setting up the

beaglebone, installing drivers and connecting it to wifi

1. Connect to beaglebone and wifi

a. connect to beaglebone through usb.

ssh root@192.168.7.2

b. Set up wifi

connmanctl

enable wifi

scan wifi

services

agent on

connect wifi\_2cf7f1068fec\_4352333736302d77696669\_managed\_psk

quit

2. Update database of apt-get and download openssh

a. grab updates for database and apt-get function

apt-get update

apt-get install xauth

apt-get install xvfb

b. install software to for open-ssh, client and server

apt-get install openssh-server

apt-get install openssh-client

3. Generate public and private keys

ssh-keygen

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

0a:9d:cc:5e:5d:2f:45:86:7d:b7:b4:3f:b9:e3:5b:c1 root@304805526.lasr.cs.ucla.edu

The key's randomart image is:

4. Set up user on the server

a. Sets up my user on the beaglebone to be able to log into

useradd -d /home/luke -m luke

passwd luke

b. Give the beaglebone user accessibility from outside

cd /home/luke

mkdir .ssh

chown -R luke .ssh

chmod 700 .ssh

5. Log into partner's beaglebone

a. Log into specified ip address using ifconfig

ifconfig

ssh-copy-id -i luke@10.97.85.49

b. Save a private key to the user so don't need a passphrase everytime.

eval `ssh-agent -s`

ssh-add

ssh luke@10.97.85.49

Now when we log into each other's beaglebone's, we don't have to put

in a passphrase.

Also, we are able to run commands like xterm on the different beaglebones,

as long as we login with the -X command when we ssh.

Therefore, we ahve successfully logged into each other's beaglebones,

both being able to be the client and the serve on both accounts.

# Homework 9

## Hw.txt

2. Create checkout branch

git checkout -b 'quote' 'v3.0'

3. Patch

patch -p1 <quote-3.0-patch.txt

patching file src/cmp.c

patching file src/diff.c

patching file src/diff3.c

patching file src/dir.c

patching file src/sdiff.c

patching file src/util.c

4/5. Change log

emacs

C-x 4 a

I copy pasted the commit description into the emacs ChangeLog, but only

the the ones where the file was changed correctly.

6. Commit Changes

man git commit

git commit -aF ChangeLog

7. Generate formatted patch file

man git format-patch

git format-patch -1 --stdout > formatted-patch.txt

8. Applying partner's

First need to make a new checkout branch called partner

git checkout -b 'partner' 'v3.0'

My partner's name:

Dominic Loftus

203-910-863

Then used the command

git am formatted-patch-dom.txt

Once it's patched, we go through the process of building the new file

./bootstrap

patch -p0 <diffutils-gets.diff

./configure

make

make check

And we see that things are successful.

9. Make Distdir

make distdir

cd diffutils-3.0.1-41d4

emacs ChangeLog

Looking into this file, I see Dominic's comments in the changelog so it

works

10. Gitk

To load a 3rd party application, I had to first download xquartz on my

computer and restart it. I went to their main website

https://www.xquartz.org/ and installed it. I then tried to gitk the

path for eggert's library, but it really wasn't opening so I decided to

just download the git from the website and run the command in there.

I actually got a few errors when installing but it seemed to work.

git clone https://git.savannah.gnu.org/git/emacs.git

gitk

Doing this opened up xquartz window and I could see all the different

commits and changes.

## Lab.txt

1. Get the git for the diffutils from the website.

git clone https://git.savannah.gnu.org/git/diffutils.git

2. Log changes

git log > git-log.txt

3. Get tags

git tag > git-tags.txt

4. Get commit that for patch

a. Find commit number

emacs git-log.txt

C-s 'like this'

Copy the commit number

62ca21c8c1a5aa3488589dcb191a4ef04ae9ed4f

b. Create a patch for the commit

git format-patch -1 62ca21c8c1a5aa3488589dcb191a4ef04ae9ed4f --stdout >

quote-patch.txt

5. Check out version 3.0

git checkout v3.0

6. Patch file

patch -p1 < quote-patch.txt

7. Check status

git status

HEAD detached at v3.0

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

8/9. Use emacs functions to revert changes

I learned you can use the emacs function 'C-x v u' to revert changes.

So I went through each file that wasn't a '.c' file and issued the command

and saved each file.

The files include

NEWS, README, TODO, doc/diagmeet.note, ms/config.bat, ms/config.site,

po/en.po, src/diff.h, src/system.h, and tests/help-version

Now to revert changes from .c files, we have to use a slightly different

command on emacs vc-diff (C-x v =) and revert changes on only comments

using C-u-c-a

The files include

src/analyze.c, src/cmp.c, src/context.c, src/diff.c, src/diff3.c,

src/dir.c, src/ifdef.c, src/io.c, src/sdiff.c, src/side.c, and src/util.c

10. Manually apply rejected patches

I went through the rejected patches using:

emacs src/\*c.rej, seeing that the patches failed because of mismatched

line patterns, and manually inputted them into the corresponding files.

This incldued only diff.c and diff3.c

I found each line by using C-s search and changing ` to '

11. Delete unwanted files

I used the find command to find all the files that have a file ending

.orig and .rej and deleted them using the commands

find -type f -name '\*.orig' -delete

find -type f -name '\*.rej' -delete

12. Create a patch for git

First we need to check if we modified the right amount of files

git status

modified: src/cmp.c

modified: src/diff.c

modified: src/diff3.c

modified: src/dir.c

modified: src/sdiff.c

modified: src/util.c

Then we create the patch for the git using

git diff > quote-3.0-patch.txt

Then I checked if the patch is enough lines using

wc quotewc quote-3.0-patch.txt

326 1776 12384 quote-3.0-patch.txt

which matches the spec

13. Patch the files

First, we open the README-hacking file and go through the commands

the first part is to run

./bootstrap

Then since I'm using lnxsrv09, I wget the patch and change the directory

to library and patch using

patch -p0 diffutils-gets.diff

Then we can make

make

Now to check we input the commands diff . - which gives the result

diff: cannot compare '-' to a directory

and diff --help

which now gives the write quotation mark

14. Checking with source code

First we need to clone the original source code into a new directory

git clone ~eggert/src/gnu/diffutils diffutil-3.0

Then we move the source code from the original and patched to their own

directories to compare

cp diffutil-3.0/src/\*.c diffutils-3.0

cp diffutils/src/\*.c diffutils-3.0-patch

Now we can make patch files for both using the command given

./diffutils/src/diff -pru diffutils-3.0 diffutils-3.0-patch >quote-3.0-test.txt

15.

And make a difference file with the new text file and our original

diff -u quote-3.0-test.txt diffutils/quote-3.0-patch.txt >diff.txt

Looking at the differences, it seems like a lot but it's really

just a lot of the of different paths and not real content

changes. When scrolling through and matching each part, it

seemed to me that every difference is innocuous.