Name: Daniel Schwartz

CS35L: Software Construction Laboratory

Winter 2018
Lab 7

Final Exam

March 22, 2018 11:30 AM – 2:30 PM

100 Points (50% of Final Grade)

Instructions:

1. Please write your response clearly in the allotted space. Unreadable answers will not be graded. You may also use the back side of every page to record your answers for the questions given on that page.
2. Use the extra pages attached towards the end for rough work.
3. If you have any questions, raise your hand and the TA will come to you to clarify the question.
4. If you need to assume anything which is beyond what is stated in the questions, please write your assumption and the corresponding solution. However, your assumption should not contradict the question itself.
5. There will be partial credits for every problem.
6. Please refrain from cheating.
7. Using cell phone during the exam is prohibited.

88

5

Q1 (5 points)

Command 1: echo I am a good boy | tr 'good' 'test'
Command 2: echo I am a good boy | sed 's/good/best/g'
What will each of these commands output? Explain how you got it.

1. I am a tSSt bSy

2. I am a best boy

✓ Sed will replace strings
g does it globally.

The tr command translates bytes, the equivalent from string 1 to string 2
g→t
o→e
o→s
d→t
& o→s overwrites o→e

6

Q2 (8 points)

Write a python function which takes three strings as input – ‘source’, ‘destination’, ‘mystring’ and implements substitution.

[Replace source characters of ‘mystring’ to destination characters]

I assume source and dest are same length
also no repeats in source ☺

Example:

```
#!/usr/bin/python
```

Input: ‘abc’, ‘def’, ‘mystringabcdef’

```
import sys, locale, string
```

Output: ‘mystringdefdef’

```
def main():
Source = args[0]
destination = args[1]
mystring = args[2]

for i in mystring
    for j in source
        if (mystring[i] == source[j])
            mystring[i] == dest[j])

print mystring

if __name__ : "__main__":
    main()
```

6

Q3 (6 points)

Write a bash script which does the following:

1. Copies all the files of the current directory into a new folder called 'mylab' (which should be in the same directory).
2. Creates a new file called 'new_file.txt' inside the 'mylab' directory with text 'CS 35L Final Exam'.
3. Appends a line to the new_file.txt – this line would be the name of the first file you obtain after sorting the filenames inside the 'mylab' directory in ascending order.

```
#!/bin/bash
mkdir mylab
RESULT=`find . -type f`
for FILE in $RESULT
do
   cp $FILE mylab/$FILE
done
touch mylab/new_file.txt
cat " CS 35L Final Exam" >> mylab/new_file.txt
touch temp2
touch temp
ls -a mylab >> temp
cat temp | sort >> temp2
head -n 1 temp2 >> mylab/new_file.txt
rm -f temp
rm -f temp2
```

```
#!/bin/bash
mkdir mylab
result=$(find . -maxdepth 1 -mindepth 1 -type f | sort)
for file in $result
Do
 cp $file mylar/$file
Done
echo “CS 35L final exam” > mylab/new_file.txt
echo $result | head -1 >> mylab/new_file.txt

#Another solution (if we only copy files but not directories)

#!/bin/bash
mkdir mylab
cp * mylab
echo "CS 35L Final Exam" >mylab/new_file.txt
ls mylab|sort|head -1 >>mylab/new_file.txt
```

6

Q4 (6 points)

In a galaxy far far away there lived a Darth Cipher. DC wants to send some information to Earth (Message to be sent from DC to E) using asymmetric encryption (Public and Private key). Darth Cipher would like to request you for help on Twitter. In less than 140 characters explain to him:

a. For broadcasting the message, whose key will be used to encrypt the message to ensure that the broadcasted message is not tampered/rewritten.

DC will use his private key !
to mak message digest and encrypt

Public broadcasting(1 to n), avoid tampering: private key to encrypt
Secure communication(n to 1): public key to encrypt, only private key can decrypt

b. For one-to-one secure communication (DC wants to send the message to E), whose key will be used to encrypt and decrypt the message to ensure no third-party is able to read the message.

DC uses E's 'Public key to encrypt ✓
E uses his private key to decrypt. ✓

3

Q5 (3 points)

How do you search for a keyword within the man page? Does the search in the man page support regular expressions? If yes, give an example.

/ [keyword]

yes.

/'a'

man -k [keyword]
man -K [keyword]
Regex can be turned on using -ew after -k,
man -k -ew [regex]

3

Q6 (4 points)

Does the following code throw any error? If yes, then what error?

Debug this code and write the correct solution (the correct solution may include inserting/deleting some lines in the given code). Also write the gdb commands you would try to debug this code.

```
1.  #include<stdio.h>
2.
3.  struct node{
4.     int val;
5.     struct node *up;
6.     struct node *down;
7.  };
8.
9.  struct stack {
10.    struct node *root;
11. };
12.
13. typedef struct stack stack_set;
14.
15. int main(){
16.    stack_set *set;
17.    set = malloc(sizeof(stack_set));
18.    set->root = NULL;
19.    set = malloc(sizeof(struct node));
20.    set->root->val = 0;
21.    printf("%d", set->root->val);
22.    return 0;
23. }
```

Yes.

segmentation fault ✓

gdb commands:

break 15
run
next
step
list ✓
info frame

19. set = malloc(sizeof(root→node));

→ root

2

Q7 (3 points)

The 'last' shell command shows all the users logged into the system for the last 30 days. Which user logged in the highest number of times in the last 30 days? Write a shell script you will use to get the result.

```
last |awk -f ' '{print $1}| sort -n > temp3
    cat head -n 1 < temp3
    rm temp3
```

last l cut -d " " -f 1lsortluniq -clsort -n ltail -1
The result is like:
360 das

9.

Q8 (9 points)

a) Input file (3 points)
file.txt:

```
apple
orange
mango
apple
grapes
mango
apple
orange
orange
apple
```

3

Task: to count each item

Hint: uniq -c counts the items using some order of the input

Command: cat file.txt | sort | uniq -c

b) (2 points)

a. I forgot the last few commands I ran, which 'shell command' do I use to see them?

history ✓

b. Without using the above 'shell command', how do I run the previously ran shell command which starts with letter 'g'

!g ✓

c) What is the maximum length for a filename under Linux? (1 point)

128 characters ✗

255 characters
//PS: Maximum path length: 4096 chars

d) What are the kinds of permissions/modes under Linux? (1.5 points)

full read/write read/execute read only ✓ write and execute
none write only
execute only

e) What does chmod g+rx do? (1.5 points)

adds read and execute permissions to the group class ✓

f) What is the ssh port number? (1 point)

a number identifying which connection port to use. For example, lnxsrv07.seas.ucla.edu uses port 22 to connect. the port # is which channel is used to transmit data through. ✓

2 5

Q9 (3 points)

Which shell command will you use to copy a file securely from your home directory on the lnxsrv09.seas.ucla.edu to your local machine? (Write the full command; assume your SEASnet account username on the lnxsrv09.seas.ucla.edu is 'user0').

✓ scp user0@lnxsrv09.seas.ucla.edu /HOME/file \
/usr/daniels/Desktop/new_file

My local machine runs Windows so I would use WinSCP. I cannot use a shell command to transfer it... ✓

2

Q10 (3 points)

a. The GDB command "info local"
- A) displays the list of local variables
- B) value of local values for the current stack frame
- [✓ C) displays the list of local variables & value of local values for the current stack frame]
- D) none of the mentioned

Make -j 2 means
A) two jobs will run simultaneously

b. 'Make' keeps track of when files were last compiled and only recompiles those target files for which source files were changed since make was last executed.
- [✓ A) True]
- B) False

✓

c. If make command is executed as "make -j 2", then
- A) two jobs will run simultaneously
- B) only two will be executed
- C) it will give an error
- [✓ D) none of the mentioned]

✗

3

Q11 (3 points)

Patching –

Why not just change the original source code to fix it? Why do we have patches?

Without patches, every time you wanted to change part of software you would have to reinstall an entire file, which may be slow. With patches, you can download a smaller file to fix/things.

4

Q12 (4 points)

List two differences between Softlink and Hardlink.

Softlinks point to the (relative or absolute) path of the file

Hardlinks point to the same inode of a file

If a file pointed to by a softlink is changed, the soft link changes as well

Deleting the file pointed to by a softlink deletes it entirely

✓ If you delete a file pointed to by a hardlink, the hardlink will still have the file's data,

Differences between hard/soft links
A soft link does not contain the data in the target file
A hard link preserves the contents of the file

A soft link points to another entry somewhere in the file system.
A soft link can link to directories, or to files on remote computers networked through NFS

Deleting a file will invalidate the softlink pointing to that file, whereas hardlink remains unchanged

9

Q13 (9 points)

a. Write a bash script to swap two numbers without using the third variable. The two numbers should be taken as input from the command line. (5 points)

```
#!/bin/bash
a=$1
b=$2

b=$1
a=$2
```

```
#!/bin/bash

a=$1
b=$2
a=`expr $a + $b`
b=`expr $a - $b`
a=`expr $a - $b`
```

5

b. Assume you created the above bash script file inside the git repository on your computer. Write down the command(s) you will use to add this file to the remote git server. (Hint: There are 3 main commands) (2 points)

```
git add <filename>
git commit
git push origin
```

use origin if it's a new branch

✓

c. Write the command to create a new branch called mybranch and then switch to it. Write down the command(s) you use to perform the above operations (2 points)

```
git branch mybranch
git checkout mybranch
```

✓

root is the user that has access to all commands and files on a Linux or Unix-like OS
root has ability to modify the system in any way and to grant and revoke access
permissions (the ability to read, modify and execute specific files and directories) for other users

3

Q14 (3 points)

User 1: Play Spotify

Prompt: Permission denied

User 1: su root. Play Spotify

Prompt: Okay

The 'su root' command allows the User 1 to become the root user of the system. What is a root user on any unix-like operating system? How is the root user different from other users?

root user is a "superuser". As root user, you can su to switch to any user without a password. root user has full permissions on the computer. ✓

1

Q15 (4 points)

Give the keystrokes for Emacs commands:

a. Copy a region
C-y

b. Paste
C-p

c. Replace
C-R

d. Undo
C-x C-u ✓

C-x R

a. M-w
b. C-y
c. M-y
d. C-x u

Q16 (3 points)

1.5

a. Write a shell command to create a directory 'finals' and 'submission' inside finals without using mkdir twice. (1 point)

```
mkdir -p finals/submission
```
✓

b. Write the shell command to find all html files in the current directory whose name starts with letter 'a' and ends with a digit followed by a '#' (1 point)

```
find . -name 'a.*([0-9]#)$'
```
X

find . -name "a.*[0-9]#.html" -maxdepth 1
OR: find . -name a*[0-9]\#

c. Write a shell command to replace the character '/' with ',' in a file in-place? (Hint: sed) (1 point)

```
sed 's/\//,/g'
```
0.5

sed -i old 's/\//,/g'

2.5

Q1 Answer the following questions (4 points)

a. Define 'critical section'. (1.5 point)

The part of code that is parallelized.

x

Critical section is the section that cannot be parallelized

2

b. Fill the blanks in the following program: (2.5 points)

```
#include<stdio.h>
#include<stdlib.h>
#include <pthread.h>

void *printMsg(void *thread_num) {
        int t_num = (int) thread_num;
        printf("It's me, thread #%d!\n", t_num); }

int main() {
        pthread_t tids[3];
        int t;
        for(t = 0; t < 3; t++) {
                int ret = pthread_create(&tids[t], NULL, printMsg, (void *) t);
                if(ret) {
                        printf("Error creating thread. Error code is %d\n", ret);
                        exit(1);
                }
        }

        for (t=0; t<3;t++) {
                int ret1 = pthread_join(tids[t], NULL);
                if(ret1) {
                        printf("Error joining thread. Error code is %d\n", ret1);
                        exit(1);
                }
        }
}
```

4-

Q18 (4 points)

a. ______ provide(s) an interface to the services provided by an operating system. (1 point)

A) Shared memory

[✓ B) System calls]

C) Simulators

D) Communication

✓

b. System calls can be run in either user mode or kernel mode. (1 point)

A) True

[✓ B) False]

✓

c. List the compilation steps of a GCC compiler (2 points)

preprocessing
compilation
assembler
linking

✓

3

Q19 How are libraries dynamically loaded? What are two advantages and one disadvantage of dynamic linking? **(4 points)**

They are loaded with a symbol and relocation table. Libraries are placed under a program's control, and the program selectively calls functions within the library.

Advantages:
1. A single library can be shared among multiple programs. This saves memory
2. When shared libraries are updated, executables that use them don't need to be updated.

Disadvantage: Hurts performance to have to load shared objects.

✓ ✓

Disadvantages: It may have lower performance, And the dynamically loaded library may be missing.

1.5

Q20 Unbuffered vs buffered I/O! Which is faster in what applications - when would you use buffered or unbuffered I/O? **(2 points)**

Buffered: Reading in large chunks of data at once. You would use it when you need to access other local data. For example, it is faster to use buffered when reading from a hard drive.

You would use unbuffered if you only want to read/write each byte individually. Each byte is read/written by the kernel through a system call. If you don't want other local bytes to be read or written, use unbuffered. Taking many other bytes in a buffer could use up space in a cache that is not needed.

system calls

In most cases, it is faster to use buffered IO, especially when we are reading or writing a small part at a time. When each read and write is already large enough, we can directly use unbuffered IO with system call, and eliminate the overhead the the buffers.

9

Q21 (9 points)

a. Which shell command is used to display the unix version (1 point)

[✓ A) uname -r]
B) uname -n
C) uname -t
D) kernel

✓

b. What would be the current working directory at the end of the following command sequence? (1 point)

```
$ pwd
/home/user1/proj
 $ cd  src
 $ cd  generic
 $ cd  .
 $ pwd
```

/home/user1/proj/src/generic ✓

c. Which of these is NOT a valid variable in bash (1 point)

A) __ (double underscore)
B) _1var (underscore 1 var)
C) _var_ (underscore var underscore)
[✓ D) some-var (some hyphen var)]

✓

d. What is the output of this program? (3 points)

```
#include<stdio.h>
#include<pthread.h>
#include<fcntl.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
   pthread_exit("Bye");
   printf("CS35L\n");
}
int main()
{
   pthread_t pt;
   void *res_t;
   if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
      perror("pthread_create");
   if(pthread_join(pt,&res_t) != 0)
      perror("pthread_join");
   printf("%s\n",(char *)res_t);
   return 0;
}
```

A) CS35L
[✓ B) Bye]
C) segmentation fault
D) run time error
E) 0

✓

e. In Linux, the static libraries can be created by (1 point) ✓

[✓ A) ar command]
B) as command
C) ap command
D) aq command

f. What does the 'strace' command do? (2 points)

intercepts and prints system calls to stderr or an output file ✓