Sample questions:

Ques 1

Imagine you have a directory inside your home directory called task1 (~/task1). This directory contains all regular files. However, they may be of different or no extensions. Write a shell script to perform the following tasks on ~/task1:

* Find the extension of all such regular files in the directory. Output all the file names and their extensions (if they have) to the stdout buffer. For example, for the file called ‘part1.txt’, you may write a statement to stdout stating “Extension of part1.txt is .txt”. And for a file called ‘part2’, you do not output
* If a file has an extension .tar.gz, untar it.
* If a file is compressed with the gzip compression utility, uncompress the file
* Determine the size of the directory task1 (in human readable format) before and after you uncompress the files

(Your script should do nothing to files having no extensions.)

#!/bin/bash

echo “before uncompressing: ”

du -sh ~/task1 # “disk usage” : Determine size of files/directory.

#We cannot use “ls -l” for listing sizes of directories

for file in $(find ~/task1 -mindepth 1 -maxdepth 1 -type f | grep ‘\.[^/]\*$’)

do

suffix=$(echo $file | sed ‘s/^[^\.]\*\././’) # may have multiple extension

echo Extension of $file is $suffix

if [suffix == ‘.tar.gz’]; do

tar -xzvf $file

done

done

echo “after uncompressing: ”

du -sh ~/task1

Ques 2

What will be the output of the following c program? Please justify your response.

----------

#include <stdio.h>

#include <stdlib.h>

int main() {

string str = "Hello World!";

string s = "hello world!";

string\* ptr = &str;

string\* pt = ptr;

printf("%s",\*pt);

return 0;

}

---------

a) Hello World!

b) hello world!

c) Compile Error

d) Runtime Error

There are no “string” in C.

Ques 3

What is a meant by a memory leak in the C programming language? Give an example of memory leak in a c program. What should you do to avoid this?

Memory leaks means that there are dynamically allocated memory segments(using malloc()) that are not freed using free() method, so until the end of the program execution, the part of the memory cannot be used.

Example:

Int main() {malloc(1000000);}

We should keep track of the dynamically allocated pointers and make sure that they are freed before leaving the main function.

Ques 4

What does the following python script do? Write an equivalent GNU/Linux command which gives the same output as the script. The input to the command should be the same as the variable var1 in the following script.

#!/bin/python

my\_dictionary = { 'a':'b', 'b':'c', 'c':'d'}

var1 = "abc"

var2 = ""

for ch in var1:

if ch in my\_dictionary:

var2+=my\_dictionary[ch]

else:

var2+=ch

print var2

The python script transliterates the input byte by byte with the dictionary.

The equivalent command is

$ tr ‘abc’ ‘bcd’

So if the input to the command to tr command is ‘abc’, the program will return ‘bcd’.

Ques 5

Can macros be used inside a makefile? If yes, give an example. If no, justify your answer

Yes. Macro can be used in a makefile.

For example:

CC = gcc

main.o: main.c

$(CC) -c -o main.o main.c

Ques 6

Which type of library linking technique will you use in the following scenarios:

* Case 1: You have an executable file which needs to be migrated between different servers. It needs to use the same set of library functions in every server. Which type of library linking will you use in this case in order to avoid dependency problems and thereby simplifying distribution? Consider that this executable is being used and stored at only one place in each server. Justify your reason.
* Case 2: Now assume that the same executable needs to be distributed among many users within the same server. Each of the users have their own working space where they would be storing and running this executable. Which type of library linking should be used in this scenario? Assume that this server has all the libraries as required by the application. Also remember that the server has its own data storage limit. Justify your response.
* Case 3: Can you mention one drawback of case 2. What will the most optimum way to reuse this executable among by different users across different servers. You can think of something beyond library linking. Assume all servers have all libraries as required by the application.

#####

**KNOWLEDGE BASE:**

**static linking**

Every program has its own copy

Consumes more space

More portable, can execute on platform with no standard library installed

Tied to a specific version of the lib. New version of the lib requires recompile of source code

**dynamic linking**

less space: only one copy of the shared lib is kept in memory

Faster to compile programs. Size of executable is reduced

Library is shared by multiple programs

Less portable, cannot execute on platform with no standard library installed

Easy to update and fix bugs in standard library

#####

Case1: we should use static linking, because we would like to avoid problem of dependency in dynamic linking and simplify distribution by not needing a separate file.

Case 2: We should use the dynamic linking, because the users have executable in separate workspace, and there is a storage limit of the workspace. Dynamic linking can be used to reduce the size of the executable, and the dynamic library can be shared among the users.

Case 3: One of the drawbacks is the performance issue. The executable need to find the dynamically linked library before using it. In the case of many user with many servers, we can share the entire executable for all the users.

Ques 7

What is a ‘conflict’ in git? When does it occur? Give the sequence of git commands to be executed in order to resolve this.

A git conflict occurs when git is not able to determine what to do with the file when they are merging. A conflict in git happens when both commits modifies certain part of the file.

To resolve the conflict, we check the status of git, which shows the conflict file:

$ git status

We use editor like emacs to clean up the file, and execute the command:

$ git add –all

$ git commit

### Additional ###

A git merging conflict may appear like this:

What pops up in your mind when you think of USC?

<<<<<<< HEAD

# You cannot spell “SUCK” without ‘U’,’S’,’C’

=======

# How many USC freshmen does it take to screw in a light bulb?

# ... None... That's a sophomore course

>>>>>>> final-is-bullshit-branch

#####

Ques 8

Which of the following options will help us determine the exact location of segmentation fault which is occurring because of using a pointer on which free() has been called earlier?

1. Using gdb to debug
2. Generate a core dump and debug the same
3. Dereference and print the pointer at different locations within the code

Select only one option from the following:

a) Only i

b) Both i and ii

c) Both i and iii

d) All three of the above

//gcc cannot show seg fault

Ques 9

What would be some advantages and disadvantages of XYZ.

What is this?

Ques 10

What are regular expressions? Describe what each of the following regular expressions can be used for. Give two examples for each. Also point out why they are not a complete solution and describe how you would make it complete.

a. [A-Za-z]\*@[A-Za-z]\*[.]com

b. www[.][A-za-z]\*[.]com

c. [0-9]\*[.][0-9]\*

Regular expression is a pattern that can match a range of strings.

* 1. This regular expression matches a email address. For example:
     1. [asbdasnjd@gmail.com](mailto:asbdasnjd@gmail.com)
     2. [ASnDn@Boeing.com](mailto:ASnDn@Boeing.com)

This is not complete because it does not allow numeric characters and special characters in the email, and also need at least one letter before and after @, so it can change to

[A-Za-z0-9\_][\*@[A-Za-z0-9\_]\*[.]com](mailto:*@[A-Za-z0-9_]*[.]com)

* 1. This regular expression matches a website URL. For example:
     1. [www.google.com](http://www.google.com)
     2. [www.gmail.com](http://www.gmail.com)

This is not complete because it does not allow numbers and allow empty name in the middle. It can change to:

www[.][A-za-z0-9]+[.]com

* 1. This regular expression matches a decimal. For example:
     1. 0.883274
     2. 43920.3

It is not complete because it allows for the first digit to be 0. We can change to :

(0|[1-9][0-9]\*)?[.][0-9]\*