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
* anything.
* 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

sizeOf=`ls -l | awk '/[0-9]/{print $5}'` # get the file size for each line/file

preSize=0

for num in $sizeOf; do

let preSize="$preSize+$num"

done

filesInCurrentDir=`ls $dir | grep "[.]"`

for file in $filesInCurrentDir; do

extention=`sed 's/^\w\+.//' <<< "$file"`

echo "the extention for $file is: "$extention

if [ "$extention" = "tar.gz" ]; then

tar zxvf $file

fi

done

sizeOf=`ls -l | awk '/[0-9]/{print $5}'`

postSize=0

for num in $sizeOf; do

let postSize="$postSize+$num" #arithmetic

done

echo "post size: $postSize"

echo "pre size: $preSize"

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! **(this would be the answer if there was no compile error)**

c) Compile Error

d) Runtime Error

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?

**When you use malloc to allocate memory for something and don’t end up freeing it at the end of the function. To avoid this, make sure that every time we use malloc or some kind of allocation, we always free it after.**

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 script changes a→b b→c and c→d. The equivalent GNU command is**

cat abc | tr [abc] [bcd]

Ques 5

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

**Yes, macros such as CC and CFLAGS are both macros that help us reduce keystrokes when compiling files. Also environment variables such as $@ and others are macros that help as well.**

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. **Static, if we want to get away from dependencies, we should always use static because all the files will always be there and we don’t have to wait for specific files to be there.**
* 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. **Dynamic, since we have a data storage limit, dynamic linking will reduce the memory usage by a lot since multiple programs can link to the library functions very easily.**
* 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. **A drawback of case 2 are dependencies, it’s very easy to have a single dll file crash and make it crash for everyone else. Also, if everyone has access to this library function, then it can be easily changes and mess up everyone else’s work.**

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 conflict in git is when two different commits change the same file and you must figure out how to merge the differences. The steps to get around this are you must:**

1. **Git pull: grab latest commit to see what you need to fix**
2. **Git merge: merge the differences and see what need to change and fix**
3. **Manually correct and edit each head for conflicts**
4. **Git add: add your changes into the pipeline**
5. **Git commit: commit your own changes**
6. **Include documentation**
7. **Git push: finally push the files back into the repository**

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 **(it should be this answer but honestly all the answers above are doing mostly the same thing.)**

Ques 9

What would be some advantages and disadvantages of XYZ.

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]\*

**Should have a ‘\’ before the [\.] so that the regexp know that we’re meaning a ‘.’ And not any other idea.**

**The first one is an email, the second is a website, and the third is a decimal number.**