HKUSPACE Community College

Shell Programming

Programming Assignment 1

Student Name	**Chan** [redacted]
Student No	201[redacted]
Class	CL-[redacted]

Part A

1. Explain how you generate random PIN with different length. (1 marks)

First, open the “urandom” file in dev folder using “cat” and use “tr” to translate all the character into 0 to 9. Then, use “fold” to limit the digit of the number with $1. Lastly, use “head” to stop looping the number and get the random PIN.

2. Explain how you count the number of files in a folder (1 marks)

“Find” command can search the type of the file in the directory. Use “-type f” to search the file only in the given directory. Then use “-maxdepth 1” to limit not to search any subfolder in the given directory. Last, use “wc -l” to count and show the number of the result from “find”.

Part B

3. Explain how you can make use of the code in Part A to do Part B. (1 marks)

First, “pwd_gen” can generate the PIN with $1 and $2, which PIN length and Key. After the PIN is generated, the program can use “shift_encrypt” to encrypt the PIN into a password.

4. Explain how you can know your file is zipped successfully before you remove it. (1 marks)

Zip the file into a zip archive first. Then, use “$?” to get error value, if the value is zero, the zip is success and delete the file.

Part C

5. Explain how you generate and handle the PINs with leading zeros. (1 marks)

First use a variable to store the $1. Then use same method of “pwd_gen” but translate into all ‘9’ instead of ‘0-9’, as a result, the maximum number are generated. Next, use a for loop start from 0 to the maximum. During looping, use ‘printf’ with format ‘%0${variable}d\n”, the variable is the one mentioned above. PINs with leading zeros are generated.

6. How can you hidden the all error messages when trying the PINs? (1 marks)

First, in the command unzip, use “-t” to test the password is correct or not. At the end of the unzip command, add “> /dev/null 2>&1”. Forward all the output of the testing unzip to the file “/dev/null”, and only return the standard error back to the console. If the error of “$?” is zero, the program will unzip the file with that password instead of testing, and output result.

Part D

7. Explain how you can read in and process the password from the password csv file. (1 marks)

Use “cat” to read the password csv first. Then use “sed” to replace all the character end with “,” and use “tail -n +2” to get the word start from line 2 in the password csv. Last, use a for loop to loop every word of the result above.

8. How can you make sure the file is unzip and placed in next to the original folder but not the present working directory (PWD). (1 marks)

Use a variable to store the path of zip file. Next, use “${variable##*/}” to remove all the character that end with “/”, and get the zip file name. Then, use “sed” to remove the file name in the path, and get the location of the zip file. Last, use “-d” in unzip to change the output next to the original folder instead of working directory.

Part E

9. Explain how you apply the lock breaker script to every files in the specified directory. (1 marks)

First, use “ls” to list all the file in the specified directory. Use a for loop to loop every file name, if the file name is end with “.zip”, then unlock the zip file using same method in “lock_breakerB” with some output change.

10. Explain how you can handle the invalid input of $1 & $2. (1 marks)

Use a “test” with “-f” argument in an if statement to test $1 is a regular file or not. If the “test” result is “false” then error message will output and file “err.log” will be generated with the error. For the invalid input of $2, “-d” argument will be used as $2 is a directory. The rest are the same as handling the invalid input of $1.