SLG OPS102 Final Practice **[ANSWER]**

|  |  |
| --- | --- |
| **A** is a very broad category that includes storage, networking, printers, keyboard and mouse, and so forth. | 1. input/output (I/O) 2. processes 3. sleeping 4. swap 5. thread 6. kernel 7. backslash [\] 8. PATH 9. HOME 10. floating point 11. integer 12. exit code 13. execution code 14. -e 15. -x 16. shell scripts 17. batch files 18. @echo off 19. %NAME% 20. %0 21. exclaimation-marks [!] 22. > NUL 23. /dev/null 24. RANDOM 25. Windows scripting 26. bash |
| **V** is a special device in Windows that acts as a "black hole" for output. |
| On a multitasking computer system, a **B** is a single instance of an executing computer program. |
| In bash scripting, we can apply option **O** to test if filename is executable (accessible if a directory). |
| Windows defaults to displaying each command in a script before executing it. If you do not want each command to be displayed, you can apply **R**. |
| When any process finishes executing, it exits with a numeric value. This can be called the **L**. |
| On Windows scripting, change any variables which will be updated during the execution of the loop by replacing the percent-signs (%) with **U**. |
| DOS and early Windows systems were inherently interactive in nature, and early scripts were called “**Q**”. |
| A process which is ineligible for execution because it is waiting for a resources is called a **C**  process. |
| Arguments to a script are called parameters. **T** contains the name of the script. |
| **D** is when system memory (RAM) is approaching full utilization, the operating system may take some of the least-recently-used areas of memory and place them in storage or compress them in order to avoid running out of free memory. |
| The remainder of the shebang line is interpreted by the **F** as the name of the shell which is to be used to interpret the script. |
| The common environment variables **I** stores user's home directory - used for relative-to-home pathnames. |
| Many modern CPUs can execute more than one **E** on a core meaning that they have hardware support for rapid switching between two or more processes one core |
| Bash can perform **K** arithmetic. |
| A **G** character outside of quotes or inside double quotes instructs the shell to ignore any special meaning that the following character may have. |

**Matching (8 Marks, 0.5 mark each)**

**MCQ (9 Marks, 1 mark each)**

1. What is the default shell in Linux?
   1. bash
   2. csh
   3. ksh
   4. zsh
2. Which line of command will resume the most recent process in the background?
   1. kill
   2. jobs
   3. fg
   4. bg
3. Which symbol is used to start the process in the background in Linux?
   1. ^
   2. $
   3. &
   4. \*
4. What does the following command do?

egrep ‘A+’ filename

* 1. Matches zero or more occurrences of ‘A’
  2. Matches one or more occurrences of ‘A’
  3. Matches zero or one occurrence of ‘A’
  4. Matches A followed by any character

1. Consider the file that contains the following :

Text

Description automatically generated

Which line(s) will be displayed when the following command is issued?

egrep ‘th.\*is’ test.file

* 1. Line 1 only
  2. Line 1 and 4
  3. Line 4 and 5
  4. Line 1 and 5
  5. None of the above

1. What is the regular expression that selects all lines that begin with 2 or more occurrences of the word “this”?
   1. “^(this) (this )+”
   2. “(this this)?”
   3. “(this )+”
   4. “^(this ){2,}”
   5. Both a and d
2. What is the regular expression that selects all empty lines?
   1. ‘\*’
   2. ‘\b’
   3. ‘$^’
   4. ‘^$’
3. Which command will display detailed information about only the hidden files and directories in the current directory?
   1. ls -ld .\*
   2. ls -al
   3. ls -la | grep “.”
   4. ls -lah’
4. To run a bash script, the kernel and shell requires \_\_\_\_\_\_\_\_\_ permission accordingly.
   1. execute [x] and read [r]
   2. read [r] and execute [x]
   3. execute [x] and write [w]
   4. only read [r]
   5. only execute [x]

**Short answers (6 Marks, 3 marks each)**

1. What is the regular expression that selects all lines that begin with 2 or more occurrences of an upper case alphabet, immediately followed by 3 whitespaces, and one or two occurrences of the word this?

"^[A-Z]{2,}\s{3}(this ){1,2}"

1. What is the regular expression that selects all lines that contain 2 to 5 occurrences of “a”, followed by any number of any characters, 0 or 1 occurrence of “x”, any number of any characters and ends with either a digit or the word “bye”?

"a{2,5}.\*x?.\*([0-9]|bye)$"

**Scripting (16 Marks)**

**Question 1 (5 marks)**

Create a windows script called “grades” that does the following:

1. Prompts user 5 times for grades in each course.
2. Compute the sum of 5 grades, and the average.
3. Display the total grade and average.

*Sample Output*

A screen shot of a computer code

Description automatically generated

**Sample script - “grades”**

@echo off

set sum=0

setlocal enabledelayedexpansion

for %%i in (1 2 3 4 5) do (

set /p num=Enter grade for subject #%%i:

set /a sum=!sum! + num

)

set /a average=%sum% / 5

echo Your total grade is %sum%

echo Your average is %average%

**Question 2 (6 marks)**

Create a bash script called “numbers” that does the following:

1. Clears the screen.
2. Prompts user for the input (a number less than 10)
3. Prints error message for incorrect input (containing non-digit characters)
4. Also prints error message for number greater than or equal to 10.
5. Continue to re-prompt for input until valid input is entered.
6. If the input is validated, then print the output as shown below:

*Sample output*

A screenshot of a computer program

Description automatically generated

**Sample script – “numbers”**

#!/bin/bash

valid=false

clear

while [[ "$valid" = false ]]

do

read -p "Type in a number less than 10: " num

if [ $(echo $num | grep "[^0-9]") ]

then

echo "Incorrect data input!"

elif [[ num -lt 1 || num -gt 9 ]]

then

echo "Number must be between 1 and 9!"

else

echo "You entered $num"

valid=true

fi

done

for ((i=$num;i>0;i--))

do

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

do

echo -n "$i"

done

echo

done

for ((i=2;i<=$num;i++))

do

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

do

echo -n "$i"

done

echo

done

**Question 3 (5 marks)**

Create a bash script called “palindrome” that does the following

1. Prompt the user for a word, or a sentence
2. Make sure all white spaces are removed.
3. Check to see if the word is palindrome after trimming off whitespaces.
   1. palindrome is a word that is identical when reversed.
   2. e.g. race car, radar, rotator, civic, bob, eve, dad, was it a car or a cat I saw

*Sample output*

A screen shot of a computer

Description automatically generated

**Sample script – “Palindrome”**

#!/bin/bash

read -p "Enter a word: " word

nw=$(echo $word | tr [A-Z] [a-z] | tr -d ' ')

bw=$(echo $nw | rev)

if [ $nw == $bw ]

then

echo $word is a Palindrome

else

echo $word is not a Palindrome

fi