Czesc1

Lab 1: Exit Status Code Usage

Write a script does an ls for a nonexistend file, folowed by a display of the exit status code. Then create a file and display the new exit status code. In each task, send the ls output to /dev/null.

Lab 2: Working with Files

Write a script that will ask the user for a directory name, which the script will create. Change the working directory to the new directory and tell the user where you are using the pwd command.

Next use the touch to create a few files followed by displaying the filenames. Use echo and redirection to put some content into the files and show the user the content. Finally, say goodbye to the user to end the script.

Lab 3: Environment Variables

Write a script whoch asks the user for a number (1 or 2) which the script will use to determine whether an environment variable will be set to yes or no, then set the environment variable and export it. Tell the user what needs to be entered when the script starts. Check to see if the user put in a nimber. If not, set the environment variable to unknown.

Lab 4: Functions

Write a script that asks the user for a number (1, 2 or 3) which is used to call a function with that number is its name. The function then displays a message with the function number within it, for example, „this message is from function 3."

Lab 5: Arithmetic

Write a script that will act as a simple calculator for add, subtract, multiply, and divide; each operation should be in a function of its own. Any of the three methods for bash arithmetic, let, expr or $((..)), may be used. The user should pass as an argument on the command line a letter (a, s, m or d) and two numbers. The letter will be used to determine whoch operation will be performed on the two numbers entered. In one or more of the three parameters is letf out, then display a usage message. Finally display the answer.

Czesc2

Lab1: Putting Case Statements into Practice

Write a script that will be given a month number as the argument and will translate this number into a month name. The result will be printed to stdout.

Lab 2: Script Arguments and Usage Information
Write a script that takes exactly one argument, a directory name. The script should printthat argument back to standard output. Make sure the script generates a usage message if needed and that it handles errors with a message.

Lab 3: Randomness
Create a script that takes a word as an argument from the user, then appends a random number to the word and display it to the user. Put in a check to make sure the user passed in a word, displaying a usage statement if a word was not passed as an argument
.
Lab 4: Strings
Write a script that will read two strings from the user. The script will perform three operations on the two strings
:
(1) Use the test
command to see if one of the strings is of zero length and if the other is of non-zero length, telling the user of both results.
(2) Determine the length of each string and tell the user which is longer or if they are of equal length.
(3) Compare the strings to see if they are the same. Let the user know the result.

Czesc3

Lab1: Background and Foreground Jobs
Create a job that writes the date to an output file thrice, with a gap of 60 seconds and 180 seconds. Check whether the job is running and bring it to foreground job. Stop the foreground job and make it run in the background. Finally, kill the background job and verify its status.

Lab2: Scheduling a One-Time Backup
Create job using ***at*** to back up files in one directory to another 10 minutes from now

Lab 3: Scheduling Repeated Backups
Set up a ***Cron*** job to backup the files in one directory to another every day at 10 am. Put the commands in file called mycron.

Lab 4: Using the sleep command
Create a small script or use the command line to create a reminder that a meeting is starting in 15 minutes. The reminder should appear 10 minutes from now