**Shell Scripts (Batch File)**

**==================**

**A “program” file that contains one or more LINUX commands**

**lab6**

**===**

|  |
| --- |
| **#!/bin/bash**  **echo Your name is**  **who**  **#print today’s date followed by current directory name**  **echo Today is**  **date**  **echo Directory is**  **pwd**  **echo You have the following files**  **#use ls command to list**  **ls –x** |

**Steps:**

**=====**

**[1] The above commands are placed in a file – using vi, emacs, etc. eg: lab6**

**[2] lab6 must be made executable**

**chmod +x lab6**

**[3] Run the file lab6 (specify current directory)**

**. / lab6**

**You will see the output of all the commands in the script.**

**#!/bin/bash is a magic first line that tells which shell is used eg:**

**#!/bin/csh**

**#!/bin/ksh**

**#!/bin/bash**

**etc**

**Comment lines:**

**===========**

**Lines start with # is a remark(comment line) and are not executed**

**Variables**

**=======**

**You can also use variables**

**Define variable ‘ i ’ to take 5**

**let i=5 i=5**

**Note: NO spacing before and after =**

**Define variable ‘ count ’ to take -138**

**let count=-138**

**To access the value of a variable precede the variable name by ‘ $ ‘**

**echo $i $count**

**Display the value of variables i and count**

**size=$count**

**Define variable ‘ size ’ to take on the value of variable ‘ count ’ (which is -138)**

**lab6a**

|  |
| --- |
| **#!/bin/bash**  **let x=12**  **echo The value of x is $x** |

**program output . /lab6a**

|  |
| --- |
| **The value of x is 12** |

|  |  |
| --- | --- |
| **Command** | **Output** |
| **let x=12** |  |
| **echo The value of x is $x** | **The value of x is 12** |
| **echo “The value” of x is $x** | **The value of x is 12** |
| **echo “The value of x is “ $x** | **The value of x is 12** |
| **echo “$x “** | **$x** |
| **echo $x** | **12** |
| **echo “The value of $x is “ $x** | **The value of $x is 12** |
| **echo “The value of x is “ x** | **The value of x is x** |
| **echo “The value of $x is $x”** | **The value of $x is $x** |

**read command**

**===========**

**Can be used to input a string from user**

**input/read the characters string from keyboard**

**program wait until the user hit the <Enter> key**

**read dt**

**read the characters entered from keyboard and assign to variable dt**

|  |  |  |
| --- | --- | --- |
| **lab6b** |  | **Program output . / lab6b** |
| **#!/bin/bash**  **#Example: Ask the user to enter a number**  **#and print the number multiplied by 300**  **echo Please enter a number**  **read x**  **let b=$x\*300**  **echo The value of $x multiplied by 300 is $b** |  | **Please enter a number**  **6 (user types “6 <Enter>”)**  **The value of 6 multiplied by 300 is 1800** |

**Shell variable arithmetic**

**==================**

**Shell variables are always stored as characters. To do arithmetic, you use the “expr” command**

**+ add 2 integers**

**- subtract second integer from first integer**

**\* multiply the 2 integers**

**/ divide first integer by second integer**

**% remainder of division first integer by second integer**

**let n=2+1 n=`expr 2+1`**

**let size=5-3 size=`expr 5-3`**

**let iz=4 iz=4**

**expr $iz+6**

**expr 4\*5 error \* meta character**

**expr 4\\*5 produce correct output 20**

|  |  |  |
| --- | --- | --- |
| **lab6c** |  | **Program output . / lab6c** |
| **#!/bin/bash**  **#Example: Input 2 numbers and print the**  **#total sum of the 2 numbers**  **echo Please enter first number**  **read num**  **echo Please enter second number**  **read za**  **let toty=$num+$za**  **echo The total sum is $toty** |  | **Please enter first number**  **3 (user types “3 <Enter>”)**  **Please enter second number**  **47 (user types “47 <Enter>”)**  **The total sum is 50** |

**Number(integer) Comparison**

**=====================**

|  |  |  |
| --- | --- | --- |
| **Shell** | **C++/Java** | **Numerical Comparison** |
| **- e q** | **= =** | **equal to** |
| **- n e** | **! =** | **not equal to** |
| **- l t** | **<** | **less than** |
| **- g t** | **>** | **greater than** |
| **-g e** | **> =** | **greater than or equal to** |
| **- l e** | **< =** | **less than or equal to** |

**Combining multiple comparison**

**========================**

**More than one comparisons above can be combined together**

|  |  |  |
| --- | --- | --- |
| **Shell** | **C++/Java** | **Combining Comparison** |
| **- a** | **& &** | **boolean AND** |
| **- o** | **| |** | **boolean OR** |
| **- !** | **!** | **boolean NOT (negate/inverse)** |

**Comparisons are used in if, if-else, if –elseif, case/switch, while, repeat, etc statements**

**conditional if**

|  |  |
| --- | --- |
| **if [condition#1]**  **then**  **command list if condition#1 is true** |  |
| **else**  **command list if condition#1 is false** | **optional** |
| **fi** |  |

|  |  |
| --- | --- |
| **if [condition#1]**  **then**  **command list if condition#1 is true**  **elif [condition#2]**  **then**  **command list if condition#2 is true**  **elif [condition#3]**  **then**  **command list if condition#3 is true**  **::::**  **elif [condition#n]**  **then**  **command list if condition#n is true** |  |
| **else**  **command list if condition#1,2,3,,,,n are false** | **optional** |
| **fi** |  |

**Notes:**

**[1] “fi” is “if” spelt backwards**

**[2] condition is using the comparisons above eg:**

**[ $a –gt 100 ]**

**[ $sz –eq $wqd ]**

**[ $num –ne $v3 –a $num –ne $v5 –a $num –ne $gj ]**

**etc**

**Example: if**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **lab6d** |  | **Program output . / lab6d** |  | **Program output . / lab6d** |
| **#!/bin/bash**  **# Input a number and print**  **#“Big number” if it is more than 24**  **echo Please enter a number**  **read mn**  **if [ $mn –gt 24 ]**  **then**  **echo Big number**  **fi** |  | **Please enter a number**  **8 (user types “8 <Enter>”)** |  | **Please enter a number**  **33 (user types “33 <Enter>”)**  **Big number** |

## Example: if - else

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **lab6e** |  | **Program output . / lab6e** |  | **Program output . / lab6e** |
| **#!/bin/bash**  **# Input 2 numbers and print the larger**  **# number**  **echo Please enter a number**  **read mn**  **if [ $mn –gt 24 ]**  **then**  **echo Big number**  **fi** |  | **Please enter a number**  **8 (user types “8 <Enter>”)** |  | **Please enter a number**  **33 (user types “33 <Enter>”)**  **Big number** |

**Example: if - else**

**Input an examination mark and print the corresponding grade according to:**

**marks grade**

**70 –100 A**

**56 – 69 B**

**40 – 55 C**

**0 – 39 D**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **lab6f** |  | **Program output . / lab6f** |  | **Program output . / lab6f** |
| **#!/bin/bash**  **# Input marks and print the grade**  **echo Please enter marks**  **read mk**  **if [ $mk –ge 70 ]**  **then**  **echo Grade A**  **elif [ $mk –ge 56 ]**  **then**  **echo Grade B**  **elif [ $mk –ge 40 ]**  **then**  **echo Grade C**  **else**  **echo Grade D**  **fi** |  | **Please enter marks**  **21 (user types “21<Enter>”)**  **Grade D** |  | **Please enter marks**  **63 (user types “63<Enter>”)**  **Grade B** |

**Assignment:**

**The total marks is final exam(60%) and coursework(40%). The student automatically fails (grade D) if either or both these components marks is less than 40%.**

**String test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number test** | **String test** |  |  |  |
| **- e q** | **=** | **equal to** | **[ a = b ]** | **a equal to b** |
| **- n e** | **!=** | **not equal to** | **[ a != b ]** | **a not equal to b** |
|  | **- z** | **string length equal 0** | **[ -z a]** | **length of string a is 0** |
|  | **- n** | **string length is not equal 0** | **[ -n a]** | **length of string a is not 0** |

**Example: String test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **lab6g** |  | **output . / lab6g** |  | **output . / lab6g** |
| **#!/bin/bash**  **echo Are you above 21?**  **read kw**  **if [ $kw = “y” –o $kw=”Yes” –o $kw=”yes” ]**  **then**  **echo You are an adult**  **elif [ $kw = “no” –o $kw=”NO” –o $kw=”n” ]**  **echo You are a juvenile**  **else**  **echo You are a kid**  **fi** |  | **Are you above 21?**  **y (<Enter>)**  **You are an adult** |  | **Are you above 21?**  **no (<Enter>)**  **You are a juvenile** |

**while statement**

**The while statement loops (repeats) as long as the test condition is true. The loop terminates when the condition is false.**

**while [ condition ]**

**do**

**command-list to repeat if condition is true**

**::::**

**{ break }**

**::::**

**{ continue }**

**::::**

**done**

|  |  |
| --- | --- |
| **lab6j** | **output . /lab6j** |
| **#!/bin/bash**  **echo “Do you want to try again (y/n) “**  **read prompt**  **while [ $prompt != “y” ]**  **do**  **echo “Do you want to try again (y/n) “**  **read prompt**  **done**  **echo Goodbye** | **Do you want to try again (y/n)**  **y**  **Do you want to try again (y/n)**  **y**  **Do you want to try again (y/n)**  **n**  **Goodbye** |

**Example: Print the total of 5 numbers entered by user**

|  |  |
| --- | --- |
| **lab6k** | **output . /lab6k** |
| **#!/bin/bash**  **let sum=0**  **let count=1**  **while [ $count –le 5 ]**  **do**  **echo Enter value $count**  **read p**  **let sum=$sum+$p**  **let count=$count+1**  **done**  **echo The total sum is $sum** | **Enter value 1**  **3**  **Enter value 2**  **5**  **Enter value 3**  **4**  **Enter value 4**  **1**  **Enter value 5**  **6**  **The total sum is 19** |