CPTR 2224

Linux 1

Overview

- A reliable backup is forgiveness
- These are example scripts. The idea is students would use an editor like kate to type these scripts on your class Linux system.
- Once the scripts are entered, please run and debug them.
- Questions

Script Basics

- The first line must indicate the name of the shell used to interpret the script.
- The file must have execute file permission set.
- The file must contain valid commands.

Example

- #!/bin/bash
- # This script will echo a sentence.
- echo This is a sample shell script.

Example

- #!/bin/bash
- # This script documents files in my home directory
- ls –la ~/* > home_list.txt
- fortune >> home_list.txt

Example

- #!/bin/bash
- # This script documents the md5 hash value of files in the home directory
- md5sum ~/* > ~/md5test1.txt
- md5sum –c ~/md5test1.txt

Example

- #!/bin/bash
- #Name: whatsup.sh
- # 1/19/1997 Patty Stygian
- # Purpose: Display who's logged in and date
- echo "Let us see who is logged in and today's date"
- who   #See who is logged in
- echo "Display date"
- date
- echo "Display seconds since 1-1-1970"
- date +%s

Example

- #!/bin/bash
- #Description Collect info on core dump files
- # and create file with directory size
- #
- touch /home/preuss/core
- find /home –name core –execdir rm '{}' \;
- du /home >>/tmp/home_sizes.txt
- less /tmp/home_sizes.txt

Example

- #!/bin/bash
- #Name:copycorp.sh
- #Purpose: copy specified file to ~/Documents/corp
- CORP01=~/Documents/corp
- mkdir –p $CORP01
- echo 'Enter name of file to copy" #prompt human
- read FILENAME # get filename
- cp $FILENAME $CORP01 #do copy
- ls –l $CORP01 # show success

Example

- #!/bin/bash
- # This requires a file to process.
- echo Enter a filename to process:
- read THEFILE
- echo The number of lines in $THEFILE is :
- wc –l $THEFILE
- echo The number of words in $THEFILE is:
- wc –w $THEFILE
- echo End of processing for $THEFILE

Example

- #!/bin/bash
- # This requires a file name as a parameter.
- echo The number of lines in $1 is :
- wc –l $1
- echo The number of words in $1 is:
- wc –w $1
- echo End of processing for $1

Example

- #!/bin/bash
- # This requires 1 to 4 file names as parameter.
- echo The script you are running is $0
- echo The number of filenames you provided is $#
- echo The number of lines in the file $1 is:
- wc –l $1
- echo The number of lines in the file $2 is:
- wc –l $2
- echo The number of lines in the file $3 is:
- wc –l $3
- echo The number of lines in the file $4 is:
- wc –l $4

Example

- #!/bin/bash
- # This checks for smb.conf file existance.
- if [-f /etc/smb.conf];
  - then
    - echo the samba server appears to be configured.
  - else
    - echo the samba server cannot be started.
  - fi

Example

- #!/bin/bash
- if [$# -ne 2] ;
- then
  - echo you must provide a login name and a database table name.
  - exit
- fi
- #store command line values in shell variables
- db_user=$1
- db_table=$2

Example

- #!/bin/bash
- # This requires filenames as parameters.
- if [$# = 4] ; then
- echo "ready to process four files."
- elif [$# = 3] ; then
- echo "ready to process three files."
- elif [$# = 2] ; then
- echo "ready to process two files."
- elif [$# = 1] ; then
- echo "ready to process one file."
- else
- echo "no filenames to process."
- fi

Example

- #!/bin/bash
- # This requires filenames as parameters
- case $# in
- 4)
- echo "ready to process four files." ;;
- 3)
- echo "ready to process three files." ;;
- 2)
- echo "ready to process two files." ;;
- 1)
- echo "ready to process one file." ;;
- *)
- echo "invalid number of filenames provided." ;;
- esac

Example

- #!/bin/bash
- # This requires filenames as parameters.
- for counter in $@
- do

    - gzip $counter

- done

Example A1

- #!/bin/bash
- # Name: ShrtMenu.sh
- # Purpose: Allow human to print, delete, or quit program
- # Display Menu
  - echo "Please choose either P, D, or Q to "
  - echo " [P]rint a file"
  - echo " [D]elete a file"
  - echo " [Q]uit"

Example A2

- # Get a response from human
  - read RESPONSE
- # Use case to match response to action
  - case $RESPONSE in
    - P|p) echo "Name of file to print?"
    - read FILENAME
    - lp $FILENAME;;

Example A3

- D|d) echo "Name of file to delete?"
  - read FILENAME
  - rm $FILENAME;;
- *)   echo "leaving now";;
- esac

Example B1

- #!/bin/bash
- # This requires a valid login name.
- # This is tar zip the login's home directory
- echo 'Enter a valid login name to process:'
- read THENAME
- if [-d /home/$THENAME];
  - then
    - echo 'This is a valid login name.'
  - else
    - echo 'This is an invalid login name.'
    - exit
- Fi

Example B2

- # we will now create a tar zip of the $THENAME directory
- tar –czvf /home/$THENAME/archive.tgz /home/$THENAME/*
- md5sum /home/$THENAME/archive.tgz > /home/$THENAME/archive.txt
- Echo 'All Done'

Example

- #!/bin/bash
- # This verifies the archive for a login
- echo Enter a valid home directory name to process:
- read THENAME
- if [-f /home/$THENAME/archive.tgz];
  - then
    - echo This is a valid login archive.
    - md5sum –c /home/$THENAME/archive.txt
  - else
    - echo This is not a valid login directory
- fi
- echo 'All Done'

Example

- #!/bin/bash
- # This is an example of required use of quotes.
- cd
- mkdir test one
- mkdir 'test two'
- cd test one
- ls -la
- pwd
- sleep 5
- cd
- cd 'test two'
- ls -la
- pwd
- sleep 5
- cd
- ls -l