UNIX

EXP 2A. Execution of Unix General Purpose Utility Commands

1. man ls
2. whoami
3. whichls
4. Echo
5. History
6. Clear
7. Id
8. Pwd
9. Date
10. Time
11. Uptime
12. Cal
13. Cat>file1
14. Ping 192.168.98.8

Exp 2b. Working with vi editor

Exp 3c. Execution of File System Management: Compression and Archiving Commands like tar, gzip, bzip2, unzip etc.

Gzip and linux gzip

1. Zip fun.zip file1.txt file2.txt
2. Ls
3. Gzip file1.txt
4. Ls

Tar command

1. Tar –cvf fun.tar file1.txt file2.txt
2. Ls
3. Tar –zcvf fun.tar file1.txt file2.txt
4. Ls

Exp 4A. Execution of User Management Commands.

1. Whoami
2. Who
3. W
4. Id
5. Exit
6. Useradd athx
7. Su
8. Sudo –s
9. Passwd
10. Sudo groupadd SEIT
11. Id
12. Id athx
13. Usermod
14. Sudo usermod –G SEIT athx
15. Id athx
16. Ls –l file1
17. Sudo chage –l root
18. Chfn

Exp 5. Execution of Process Management Commands

1. Sleep 50
2. Sleep 50 &
3. Jobs
4. Sleep 50 m%
5. Fg %2
6. Sleep 50 &
7. Fg %2
8. Top &
9. Fg %2
10. Jobs
11. Sleep 10&
12. Sleep 10
13. Sleep 100 &
14. Sleep 100
15. Ps
16. Ps ux
17. Ps 3201
18. Ps 6787
19. Pstree
20. Ps –el |grep terminal
21. Nice –10 gnome-terminal
22. Pidof chrome
23. Killall chrome

Exp 6. Execution of Memory Management Commands

1. Free
2. Free –b
3. Free –m
4. Free –g
5. Free –t
6. Cat /proc/meminfo
7. Df
8. Df –h
9. Df –t
10. Vmstat
11. Vmstat –a
12. Vmstat –f

Exp7. Execution of basic programs of shell scripting

Program 1

#!/bin/bash

#reading data fron the user

read -p ' Enter a : ' a

read -p 'Enter b:' b

add=$((a + b))

echo= $a+$b = $add

sub=$((a-b))

echo $a -$b = $substract

mul=$((a\*b))

echo $a\* $b = $multiply

div=$((a/b))

echo $a / $b = $divide

mod=$((a % b))

echo $a % $b = $mod

((++a))

echo Increment operater when applied on "a" results into a = $a

((--b))

echo Decrement operator when applted on "b" results into b = $b

Terminal : vim Exp7a\_60.sh

Bash Exp7a\_60.sh

Program 2:

#!/bin/bash

echo "Enter Amount:"

read p

echo "Enter time period:"

read n

echo "Enter ROI:"

read r

i= `expr $p \\* $n \\* $r`

i=` expr $i / 100

echo Simple Interest is: $i

Terminal : vim filename.sh

Bash filename.sh

Program 3:

#!/bin/bash

echo ^LEAP YEAR CHECKER

echo -n Enter the year to check:

read check\_year

if [`expr $check\_year % 4` == 0]

then

echo $check\_year is a leap year

else

echo $check\_year is not a leap year

Fi

Terminal : vim filename .sh

Bash filename .sh

Exp8. Execution of advanced programs of shell scripting.

a. Write a shell script to implement menu-driven calculator using case statement.

b. Write a shell script to compare two strings.

c. Write a shell script to read and check if the directory / file exists or not, if not make the directory / file.

Program 1:

#This code is written by Atharva

echo "Enter the first number"

Read a

echo "Enter the second number"

Read b

echo "Enter the operator:"

echo -e "Addition: +\nSubtraction: -\nMultiplication: x\nDivision: /"

read op

case $op in

+) c=`expr $a + Sb

echo "Sum of $a and $b is $c";;

-) c=`expr $a - $b

echo "Difference of $a and $b is $c";;

x) c=`expr $a \\* Sb

echo "Product of Sa and $b is $c";;

/) c=`expr $a / $b

echo "Division of $a and $b is $c";;

\*) echo "Invalid Operator"

exit;;

Esac

Terminal: vim

Bash

Program : 2

#shell script to compare two strings

#This code is written by Athx

read -p "Enter two strings: str1 str2

if [ $str1 == $str2 ]

then

echo "Equal"

else

echo "Un Equal"

Fi

Terminal : vim , bash

Program 3:

#!/bin/bash

#This code is written by Athx

echo "Enter directory name"

read ndir

if [ -d "$ndir" ]

then

echo "Directory exist"

Else

`mkdir $ndir`

echo "Directory created"

Fi

Exp 9. Execution of scripts using grep / sed commands

a. Write a script using grep command to find the number of words character, words and lines in a file.

b. Write a script using egrep command to display list of specific type of files in the directory.

c. Write a script using sed command to replace all occurrences of particular word in given a file.

Program 1.

#!/bin/bash

Echo enter the filename

Read file

Wm `cat $file|wc –w`

C= `cat $file|wc –c`

L=`grep –c “,” $file `

Echo number of character in $file is $c

Echo number of words in $file is $W

Echo number of lines in $file is $l

Grep –l “grep” $file

Terminal : cat 9a.txt

Bash exp9a.sh

Program 2 :

#!/bin/bash

Egrep -r ‘\*sh”

Egrep –r “sample”\*

Egrep –c ‘begin|end$’ 9a.txt

Program 3:

#!/bin/bash

Sed –i ‘s/Apsit/APSHAH/g’new.txt

Exp10.: Execute the following scripts using awk/perl languages:

a) Write an awk script to print all even numbers in a given range.

b) Write an awk script to develop a Fibonacci series (take user input for number of terms). c) Write a perl script to check a number is prime or not

Program 1.

#!bin/bash

Read –p “Number:” num

Awk ‘{for(i=0;i<=$1;i++) if(i%2==0) printi}’<<<$num

Terminal:vim ,bash

Program 2:

#!bin/bash

Echo “how manyno”

Read num

Echo num is $num

Awk ‘ BEGIN{

A=0; b=1;

For(i=0;i<’${num}’;i++)

{

Print a;

C=a+b;

A=b;

B=c;

}

}

Program 3.

#!/usr/bin/perl  
 use strict;  
 use warnings;  
 print "upto what number you want to calculate primes?", "\n";  
 my $n = <STDIN>;  
 my $count=2;  
 my $j=2;  
 my $zero\_remainder\_found = 0;  
 my $remainder = $n % $j;  
 for (my $count=2; $count<=$n; $count++) {  
 $zero\_remainder\_found = 0;  
 for (my $j=2; $j<$count;$j++) {  
 $remainder = $count % $j;  
 if ($remainder == 0) {  
 $zero\_remainder\_found = 1;  
 last;  
 }  
 }  
 if (!$zero\_remainder\_found) {  
 print "$count is a prime number \n";  
   
 }   
 }

Terminal:

Vim

Perl

Exp11. To create symlink and hardlink of the file.

(symlink)

1. Touch jerry
2. Ls –l jerry
3. Sudo Ln –s/home/desktop/Athx/jerry/boot/tom
4. Ls –l tom
5. Ls –il tom
6. Rm jerry
7. Ls –il tom

(hard link )

1. Touch veer
2. Ls –l veer
3. Ls –il veer
4. Sudo ln –s /home/desktop/athx/veer/boot/ak
5. Ls –l ak
6. Ls –il ak
7. Rm veer
8. Ls –l ak