## PRACTICAL – 5

1. **Write a Shell script to accept a string as command line argument and reverse the same.**

if [ $# -ne 1 ]; then

echo "Usage: $0 <string>"

exit 1

fi

string=$1

reverse=""

for (( i=${#string}-1; i>=0; i-- )); do

reverse="$reverse${string:$i:1}"

done

echo $reverse



1. **Write a shell script to calculate the loss percentage of an article, given the cost price and the selling price as command line arguments.**

if [ $# -ne 2 ]; then

echo "Usage: $0 <cost price> <selling price>"

exit 1

fi

cost\_price=$1

selling\_price=$2

if [ $selling\_price -lt $cost\_price ]; then

loss=$(($cost\_price - $selling\_price))

loss\_percent=$(echo "scale=2; ($loss / $cost\_price) \* 100" | bc)

echo "Loss Percentage: $loss\_percent%"

elif [ $selling\_price -gt $cost\_price ]; then

profit=$(($selling\_price - $cost\_price))

profit\_percent=$(echo "scale=2; ($profit / $cost\_price) \* 100" | bc)

echo "Profit Percentage: $profit\_percent%"

else

echo "No loss, no profit."

fi



1. **Write a shell script to accept the name of the user and check out if the same has logged in or not.**

If [ $# -ne 1 ]; then

echo "Usage: $0 <username>"

exit 1

fi

username=$1

last | grep $username > /dev/null

if [ $? -eq 0 ]; then

echo "$username has logged in."

else

echo "$username has not logged in."

fi



1. **Write a shell script to check whether the file whose name is scanned exists and readable.**

if [ $# -ne 1 ]; then

echo "Usage: $0 <filename>"

exit 1

fi

filename=$1

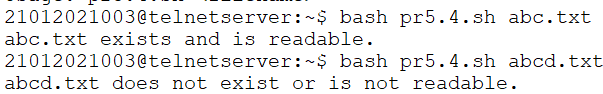
if [ -r $filename ]; then

echo "$filename exists and is readable."

else

echo "$filename does not exist or is not readable."

fi



1. **Write a shell script to check if the input string is a palindrome.**

if [ $# -ne 1 ]; then

echo "Usage: $0 <string>"

exit 1

fi

string=$1

reverse=""

for (( i=${#string}-1; i>=0; i-- )); do

reverse="$reverse${string:$i:1}"

done

if [ "$string" = "$reverse" ]; then

echo "$string is a palindrome."

else

echo "$string is not a palindrome."

fi

****

1. **Write a shell script to accept a number and a word as command line arguments and print the word the given number of times on each line.**

if [ $# -ne 2 ]; then

echo "Usage: $0 <number> <word>"

exit 1

fi

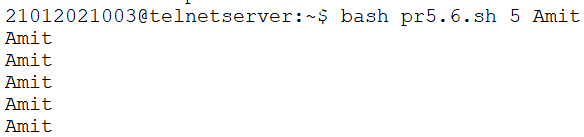
number=$1

word=$2

for (( i=1; i<=number; i++ )); do

echo $word

done

****

1. **Write a shell script to find the file or directory with the maximum size in the current directory.**

max\_size=0

for file in \*; do

if [ -d $file ]; then

size=$(du -s $file | cut -f1)

else

size=$(wc -c < $file)

fi

if [ $size -gt $max\_size ]; then

max\_size=$size

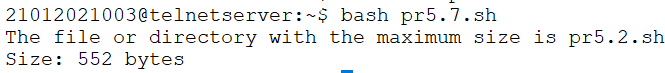
max\_file=$file

fi

done

echo "The file or directory with the maximum size is $max\_file"

echo "Size: $max\_size bytes"

****

1. **Write a shell script to accept two filenames and check if both exist. If the second filename exists, then the contents of the first filename should be appended to it. If the second filename does not exist then create a newfile with the contents of the first file.**

if [ $# -ne 2 ]; then

echo "Usage: $0 file1 file2"

exit 1

fi

if [ ! -f $1 ]; then

echo "File '$1' does not exist"

exit 1

fi

if [ -f $2 ]; then

cat $1 >> $2

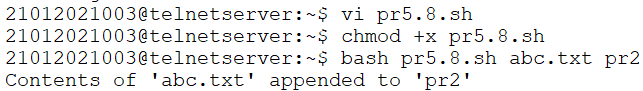
echo "Contents of '$1' appended to '$2'"

else

cp $1 $2

echo "File '$2' created with contents of '$1'"

fi

****

1. **Write a shell script to accept a number in the command line and displays the sum up to that number. By default, the sum up to 50 should be displayed.**

if [ $# -eq 0 ]

then

n=50

else

n=$1

fi

sum=0

for (( i=1; i<=$n; i++ ))

do

sum=$((sum+i))

done

echo "The sum up to $n is $sum."

****

1. **Write a shell script to find the number of ordinary files and directory files in the current directory.**

num\_files=0

num\_directories=0

for item in $(ls -1)

do

if [ -f $item ]

then

num\_files=$((num\_files+1))

elif [ -d $item ]

then

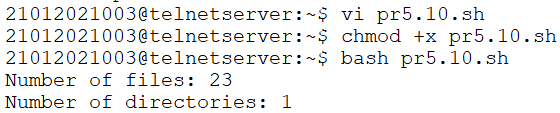
num\_directories=$((num\_directories+1))

fi

done

echo "Number of files: $num\_files"

echo "Number of directories: $num\_directories"

****

1. **Write a shell script to accept an alphabet from the user and list all the files/directory starting with that alphabet in the current directory.**

echo "Enter an alphabet:"

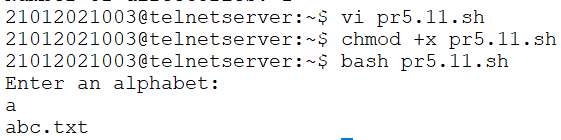
read alphabet

for item in $alphabet\*

do

echo $item

done

****