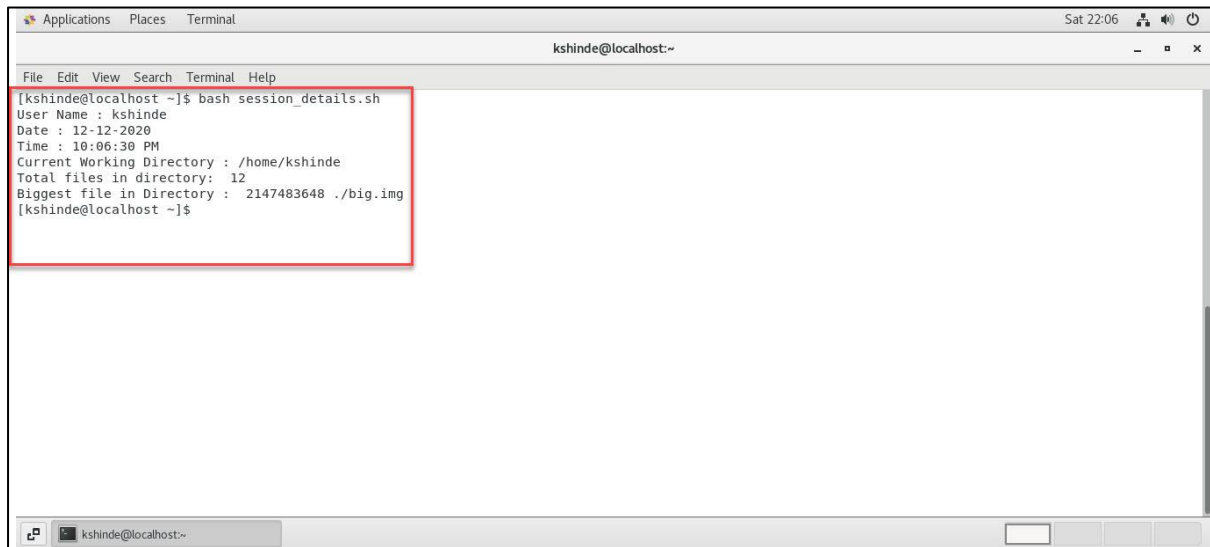


Linux Administration

Assignment Day 8

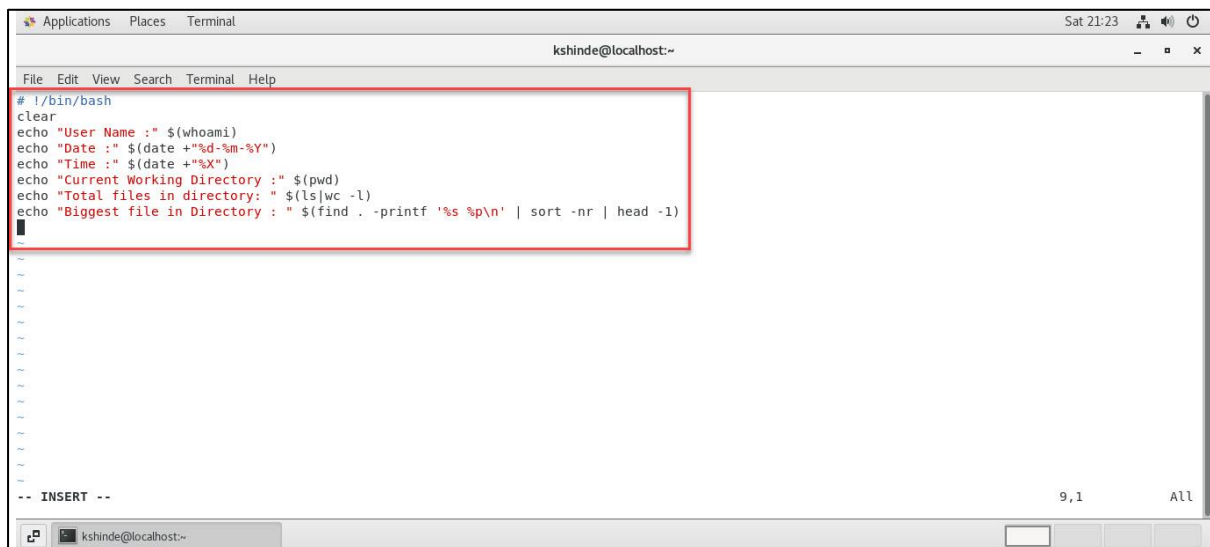
Assignment 0 : Create a simple shell script to tell the user about their session



A terminal window titled 'Applications Places Terminal' with a status bar showing 'Sat 22:06'. The terminal shows the command `[kshinde@localhost ~]$ bash session_details.sh` and its output:

```
[kshinde@localhost ~]$ bash session_details.sh
User Name : kshinde
Date : 12-12-2020
Time : 10:06:30 PM
Current Working Directory : /home/kshinde
Total files in directory: 12
Biggest file in Directory : 2147483648 ./big.img
[kshinde@localhost ~]$
```

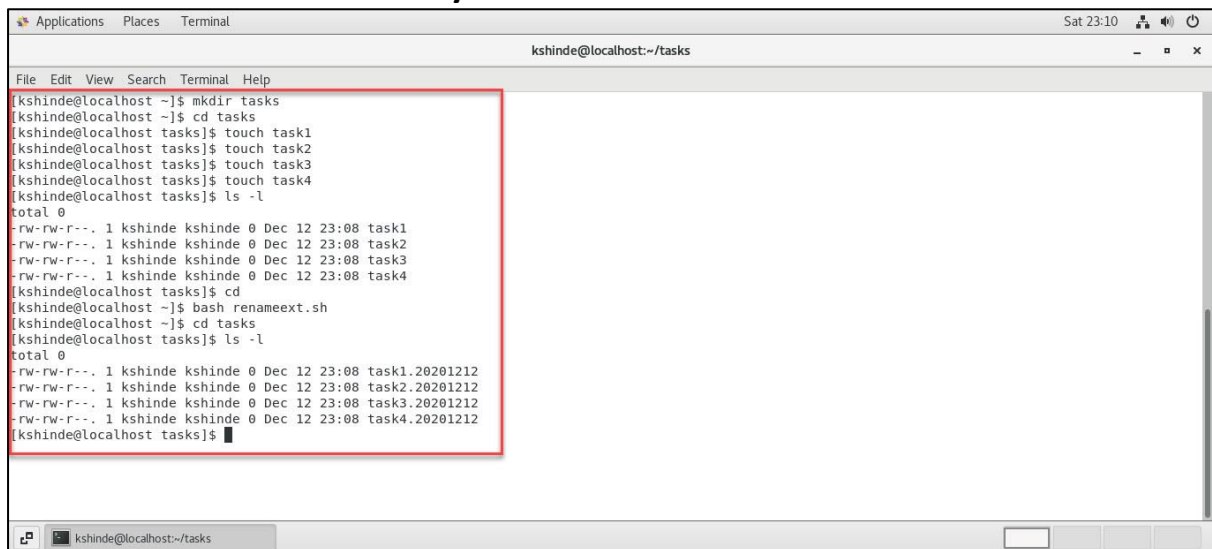
Code :



A terminal window titled 'Applications Places Terminal' with a status bar showing 'Sat 21:23'. The terminal shows the command `# !/bin/bash` and the source code of the script:

```
# !/bin/bash
clear
echo "User Name : " $(whoami)
echo "Date : " $(date +%d-%m-%Y)
echo "Time : " $(date +%X)
echo "Current Working Directory : " $(pwd)
echo "Total files in directory: " $(ls|wc -l)
echo "Biggest file in Directory : " $(find . -printf '%s %p\n' | sort -nr | head -1)
```

Assignment 1 : Create a directory with a few test files in it (the files can be empty). Now write a script that for every file in that directory you rename it to have an extension of today's date in YYYYMMDD format.



```
kshinde@localhost: ~]$ mkdir tasks
kshinde@localhost: ~]$ cd tasks
kshinde@localhost: tasks]$ touch task1
kshinde@localhost: tasks]$ touch task2
kshinde@localhost: tasks]$ touch task3
kshinde@localhost: tasks]$ touch task4
kshinde@localhost: tasks]$ ls -l
total 0
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task1
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task2
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task3
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task4
kshinde@localhost: tasks]$ cd
kshinde@localhost: ~]$ bash renameext.sh
kshinde@localhost: ~]$ cd tasks
kshinde@localhost: tasks]$ ls -l
total 0
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task1.20201212
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task2.20201212
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task3.20201212
-rw-rw-r--. 1 kshinde kshinde 0 Dec 12 23:08 task4.20201212
kshinde@localhost: tasks]$
```

Code :



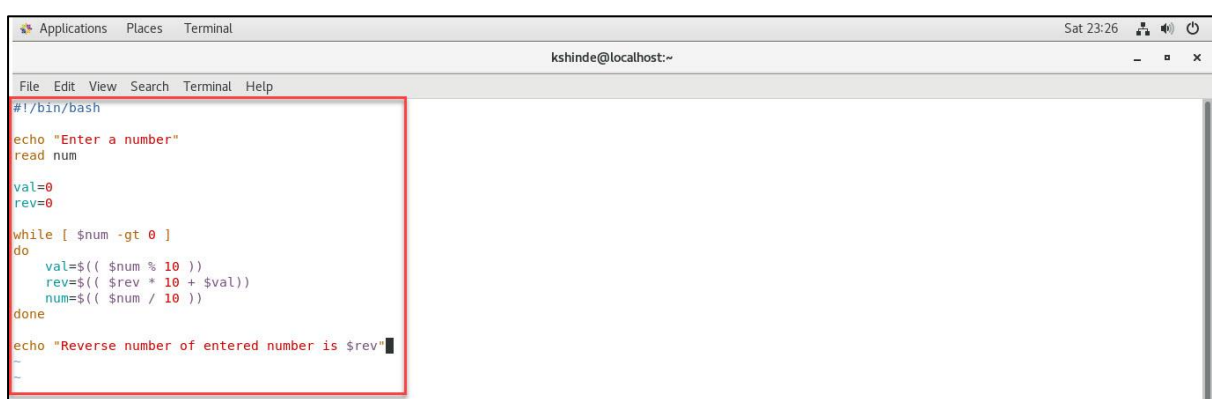
```
#!/bin/bash
cd tasks
for f in *;
do
mv "$f" "${f%.*}.${date +%Y%m%d}"
done
```

Assignment 2 : Write a script that takes a number as an input and reverses it out to the user.



```
kshinde@localhost: ~]$ vi reverse.sh
kshinde@localhost: ~]$ bash reverse.sh
Enter a number
59821
Reverse number of entered number is 12895
kshinde@localhost: ~]$
```

Code :



```
#!/bin/bash

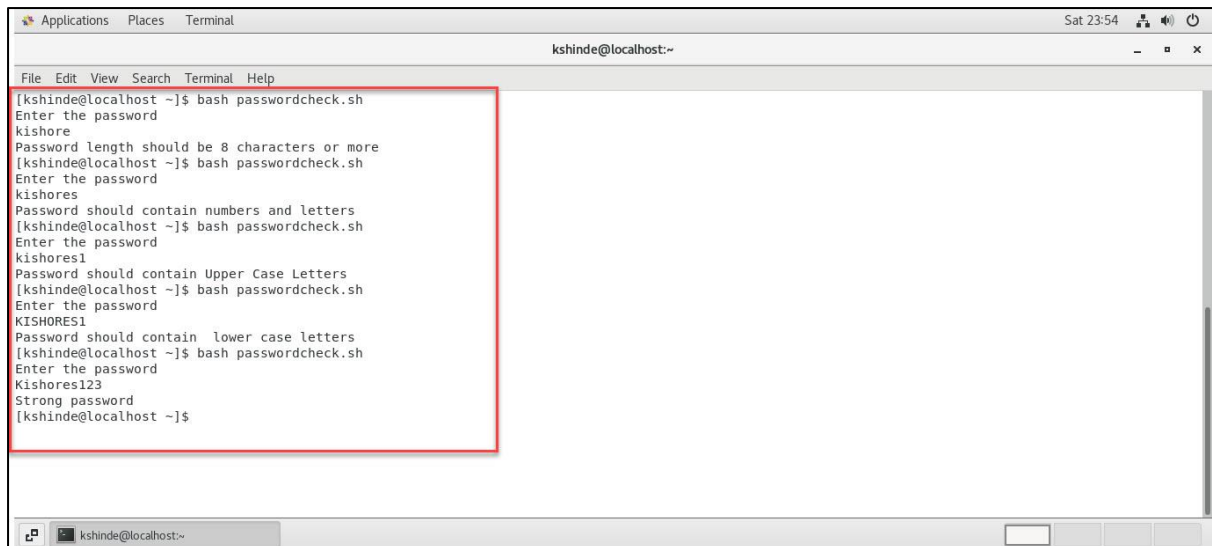
echo "Enter a number"
read num

val=0
rev=0

while [ $num -gt 0 ]
do
    val=$(( $num % 10 ))
    rev=$(( $rev * 10 + $val ))
    num=$(( $num / 10 ))
done

echo "Reverse number of entered number is $rev"
```

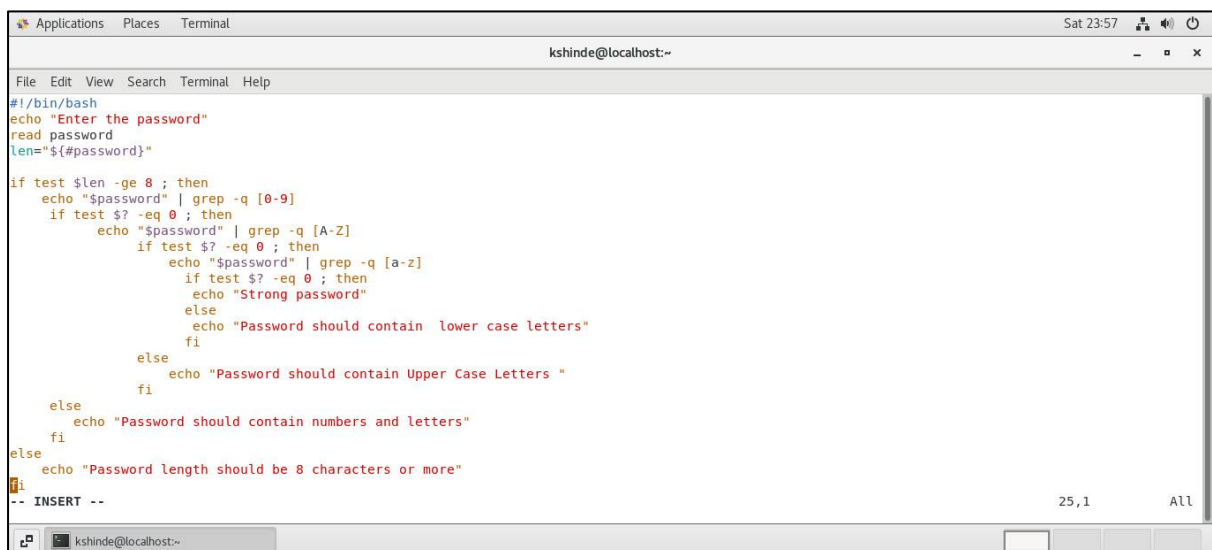
Assignment 3 : Write a script to validate how secure someone's password is



A terminal window titled 'Applications Places Terminal' with the user 'kshinde@localhost'. The terminal shows the execution of a script named 'passwordcheck.sh'. The script prompts the user to 'Enter the password'. Three test cases are shown: 1. Password 'kishore' is rejected with the message 'Password length should be 8 characters or more'. 2. Password 'kishores' is rejected with the message 'Password should contain numbers and letters'. 3. Password 'kishores1' is rejected with the message 'Password should contain Upper Case Letters'. The script then prompts for another password, and 'KISHORES1' is accepted with the message 'Password should contain lower case letters'. Finally, 'Kishores123' is accepted with the message 'Strong password'.

```
[kshinde@localhost ~]$ bash passwordcheck.sh
Enter the password
kishore
Password length should be 8 characters or more
[kshinde@localhost ~]$ bash passwordcheck.sh
Enter the password
kishores
Password should contain numbers and letters
[kshinde@localhost ~]$ bash passwordcheck.sh
Enter the password
kishores1
Password should contain Upper Case Letters
[kshinde@localhost ~]$ bash passwordcheck.sh
Enter the password
KISHORES1
Password should contain lower case letters
[kshinde@localhost ~]$ bash passwordcheck.sh
Enter the password
Kishores123
Strong password
[kshinde@localhost ~]$
```

Code :



A terminal window titled 'Applications Places Terminal' with the user 'kshinde@localhost'. The terminal displays the source code of the 'passwordcheck.sh' script. The script uses 'read' to get the password and 'len' to store its length. It then uses a series of 'if' and 'test' statements to check if the password is at least 8 characters long, contains digits, uppercase letters, and lowercase letters. If all conditions are met, it prints 'Strong password'. Otherwise, it prints specific error messages for each failed condition. The script ends with a 'fi' statement and a '25,1 All' status indicator.

```
#!/bin/bash
echo "Enter the password"
read password
len=${#password}

if test $len -ge 8 ; then
    echo "$password" | grep -q [0-9]
    if test $? -eq 0 ; then
        echo "$password" | grep -q [A-Z]
        if test $? -eq 0 ; then
            echo "$password" | grep -q [a-z]
            if test $? -eq 0 ; then
                echo "Strong password"
            else
                echo "Password should contain lower case letters"
            fi
        else
            echo "Password should contain Upper Case Letters "
        fi
    else
        echo "Password should contain numbers and letters"
    fi
else
    echo "Password length should be 8 characters or more"
fi
-- INSERT --
```

----- Assignment Day 8 Completed -----