Output:

Mon Oct 07 09:42:49 PM UTC 2024

1 date
2 history

man date
man history

Name
date - print or set the system date and time

Synopsis
date [options] ------ [+format]

root

root

Expt. No. 01-a

Date 07/10/24

Page No. 01

# PART-A

01) Learn the use of basic UNIX commands.

@ To access information using date, history, man, who, whoami, uptime, finger, cal.

Commands

1. Date: date.

2. History: history

3. Man:
man date
man history.

4. Who:
who

5. Whoami:
whoami

Teacher's Signature

22:09:01 up 26 min 0 users, load average: 0.11, 0.20, 0.20

| Login | Name | tty | Idle | Login Time |
|---|---|---|---|---|
| root | root | tty7 | 0tc | 11 21:16 |
| root | root | pts/0 | 0tc | 11 21:21 |

October 2024

| Su | Mo | Tu | We | Th | Fr | Sa |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 | | |

Expt. No. 01-a

Date

Page No. 02

6. Uptime:

uptime.

7. finger: (It will not work in shell-script)

finger.

8. cal:

cal.

Teacher's Signature

Output:

cat fruit
Apple
Mango
Grapes
Banana
Watermelon

---

vi file.txt [edits the file]

---

more fruit (displays the content of a file page-by-page)

---

cat > names
hi
hello
good
bye
morning
jan
feb
error
file ↗

bad
sad
smile

Head

hi
hello
good
bye
morning
jan
feb
error
file
bad

Expt. No. 01-b

Date

Page No. 03

(b) Display contents of the files using cat, head, tail & grep, cmp, wc, vi, more.

Commands.

1. cat:

cat > fruit
Apple
Mango
Grapes
Banana
Watermelon

2. vi

vi [filename]

3. more

more [filename]

4. head (display 1st 10 lines)
head [filename]

head names

Teacher's Signature

good
bye
morning
jan
feb
error
file
bad
sad
smile

tail

jan

– (shows nothing)

fruit fruit1 differ: byte 7, line 2

shows nothing when same value in diff file.

Expt. No. 01-b Page No. 04 Date

5. tail. (display last 10 lines)
tail filename

tail names.

6. grep. – searches for patterns in a file
grep [pattern] [filename]

grep jan names

grep xyz names

7. cmp – compares.

| cat > fruit | cat > fruit1 | cat > fruit2 |
|---|---|---|
| Apple | Apple | Apple |
| Mango | Grapes | Mango |
| Litchi | Banana | Litchi |
| Kiwi | Kiwi | Kiwi |
| Papaya | Watermelon | Papaya |

cmp fruit fruit1

cmp fruit fruit2

Teacher's Signature

5 5 31 fourst
lines words chars

5 fourst

31 fourst

5 fourst

Expt. No. 01-b. Date

Page No. 05

8. wc. - word count
wc [filename]

wc fourst

wc -l (no. of lines)
wc -l fourst

wc -c (no. of characters)
wc -c fourst

wc -w (no. of words
wc -w fourst

Teacher's Signature

total 20

drwxr-xr-x 3 root root 163 Aug 21 20:1 bench.py

-rw-r--r-- 1 root root 167 Oct 4 21:21 fruit

-rw-r--r-- 1 root root 167 Oct 4 21:27 fruit2

Apple
Mango
Kiwi
Kiwi
Papaya

dog
cat
rat
cow
sheep

Expt. No. 1-C

Date

Page No. 06

(c) To manage files using cat, cp, ls, mv, rm, chmod, find.

01. ls

ls

ls -l.

02. cp

cp [source] [destination]

cp fruit fruit2
cat fruit2

03. cat - used to create new file

cat > filename

cat > animals
dog
cat
rat
cow
sheep

cat animals.

Teacher's Signature

Apple
Mango
Litchi
Kiwi
papaya

fruit file deleted

```
-rw-rw-r--  1  root root  39  Oct 11  21:24  fruit1
-wx--x-w-   1  root root  86  2024-10-4  21:24  fruit2
```

fruit2

Expt. No. 01-(E) Page No. 07 Date

04. mv

mv [source] [destination]

mv fruit fruit2
cat fruit2.

05. rm - deletes / removes the files
rm [filename]

rm fruit.

06. chmod

chmod [permissions] [file]

chmod g+w fruit2

chmod 312 fruit2

07. Find

find [directory]
(or)
find [filename]

find fruit2

Teacher's Signature

| PID | TTY | TIME | CMD |
|---|---|---|---|
| 2215 | pts/0 | 00:00:00 | bash |
| 3974 | pts/0 | 00:00:00 | ps |

command 'ppid' from dpkg (2.4.9-1+2ubuntu3)

/dev/pts/0

| | |
|---|---|
| real | 0m0.000s |
| usr | 0m0.000s |
| sys | 0m0.000s |

| | |
|---|---|
| user | 0m.78s |
| sys | 0m.47s |

PID 47 killed
No such process

Terminates a script

Expt. No. 01-(d) Date

Page No. 08

(d) Process utilities using ps, pid, ppid, tty, time, kill, exit.

Commands

a) ps.
ps

2) pid - process id
pid

3) ppid - present process id
ppid

4) tty - tele type writer
tty.

5) time
time

6) kill
kill [PID]

kill ish (To forcefully terminate the process)
kill 47
kill 344 (No such process

7) Exit
exit

Teacher's Signature

/root

[root @ local host ~]#

[root @ local host rems]#

[root @ local host ~]#

removed rems

moved root to rems

Expt. No. 01 - (E) Date

Page No. 09

(E) Directory handling utilities using cd, mkdir, rmdir, mv, pwd.

Commands

1. cd - change directory.
   cd [directory]

2. pwd
   pwd

3. mkdir - creates a new directory
   mkdir [directory name]

   mkdir rems

   cd rems

   cd ~ (To comeback to original directory)

4. rmdir
   rmdir [directory name]
   rmdir rems.

5. mv - to move directories
   mv [source] [destination]

   mv rems root

Teacher's Signature

Output: sh new.sh
a.out
fruit

Expt. No. 02 Date 07/10/2024
Page No. 10

02) Write a shellscript that displays list of all file in the current directory to which the user has read, write and execute permissions.

```
# cat > fruit                    # vi new.sh
Apple
Mango
```

Program:

```
for file in *;
do
if [[ -f $file && -r $file && -w $file &&
      -x $file ]]; then
echo $file
fi
done
```

Teacher's Signature

Output: sh occurance.sh
Invalid Argument

sh occurance.sh fruit1 fruit2

Apple
Occurance in fruit2 2
---
Mango
Occurance in fruit2 1
---
Orange
Occurance in fruit2 1
---
Apple
Occurance in fruit2 2
---

Expt. No. 03 Date 10/10/2024
Page No. 11

03) Write a shell script that accepts a list of all the file names as its arguments, count and reports the occurance of each word that is present in the first argument file on other argument files.

cat > fruit1
Mango
Apple
Kiwi
Mango
Apple

cat > fruit2
Apple
Mango
Orange
Apple

vi occurance.sh

Program:
```
if [ "$#" -lt 2 ]; then
echo "Invalid Argument"
exit 1
fi
shift
for word in $(cat $1); do
echo $word
for file in "$@"; do
count=$(grep -o -w $word $file | wc -l)
echo "Occurance in" $file $count
done
echo "------"
done
```

Teacher's Signature

Output: sh convert.sh
Invalid Argument

sh convert.sh fruit1
MANGO
APPLE
KIWI
MANGO
APPLE

Expt. No. 04 Date 10/10/2024
Page No. 12

4) Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase provided by they exist in the current directory

cat > fruit1 vi convert.sh.
Mango
Apple
Kiwi
Mango
Apple

Program:
```
if [ "$#" -lt 1 ]; then
echo "Invalid Arguments"
exit 1
fi
for word in $(cat $1); do
echo $word | tr '[a-z]' '[A-Z]'
done
```

Teacher's Signature

Output: sh space.sh
Invalid Arguments
sh space.sh text
Lines with exact two characters
to
be
Lines with more than one space
bca mca

Date 10/10/24
Expt. No. 05
Page No. 13

5) Write grep commands to the following
(a) To select the lines from a file that has exactly 2 characters
(b) To select the lines from a file that has more than one blank spaces.

cat > txt
to
be
then
nims college
bca mca

vi space.sh

Program:
```
if [ "$#" -lt 1 ]; then
echo "Invalid Augumenti"
exit 1
fi
echo "Lines with exact two characters"
grep '^..$' $1
echo "--------"
echo "Lines with more than one space"
grep '  ' $1
```

Teacher's Signature

Output:

```
sh compare.sh
Invalid Arguments

sh compare.sh fruit1 fruit2
Files are same fruit2 is removed
```

Output 2:

```
sh compare.sh fruit1 fruit3
Files are not same.
```

Date 10/10/24

Expt. No. 06

Page No. 14

06) Write a shellscript that accepts two file names as arguments, compare the contents. If they are same, then delete the second file

```
cat > fruit1
Apple
Mango
Grapes

cat > fruit2
Apple
Mango
Grapes

cat > fruit3
Orange
Apple
Mango
```

vi compare.sh

Program:

```
if [ "$#" -lt 2 ]; then
echo "Invalid Arguments"
exit 1
fi
if cmp -s $1 $2; then
rm $2
echo "Files are same" $2 "is removed"
else
echo "Files are not same"
fi
```

Teacher's Signature

Expt. No. 07 Date 10/10/24

Page No. 15

07) Write a shellscript

(a) To count no. of lines in a file that do not contain vowels

(b) To count the no. of characters, words, lines in a given file.

```
cat > pattern                    vi vowel.sh
rems
sky
education
pen

Program:
if [ "$#" -lt 1 ]; then
echo "Invalid Arguments"
exit 1
fi
nvowel = $(grep -vi '[aeiou]' $1 | wc -l)
echo "Number of lines without vowels: "$nvowel
ccount = $(wc -c < $1)
wcount = $(wc -w < $1)
lcount = $(wc -l < $1)
echo "Total number of characters:" $ccount
echo "Total number of words:" $wcount
echo "Total number of lines:" $lcount
```

Teacher's Signature

Output:

```
sh vowel.sh
Invalid Arguments

sh vowel.sh pattern
Number of lines without vowels : 2
Total number of characters : 23
Total number of words : 4
Total number of lines : 4
```

[root@ localhost ~] #

[root@ local host. nems] #

[root @ locat host ~] #

Date 10/10/24

Expt. No. 08

Page No. 16

08) Write a shellscript to list all the files in a given directory

--> ls -l
--> mkdir nems
--> cd nems
cat > reg1
chilli
potato

cat > animal
dog
cat
Monkey
Donkey

cat > add.c
#include<stdio.h>
int main()
{
int a,b;
printf("Enter the value of a and b");
scanf("%d%d", &a, &b);
printf("Sum of a and b is =%d", a+b);
return 0;
}

--> ls -l
--> cd ~
--> ls -l
vi direct.sh

Teacher's Signature

Output:

```
sh direct.sh
Invalid Augument

sh direct.sh nems
nems/add.c
nems/animal
nems/vegl
```

Expt. No. 08 Date

Page No. 17

Program.

```
if [ "$#" -lt 1 ]; then
echo "Invalid Augument"
exit 1
fi
if [ -d $1 ]; then
for file in $1/*; do
if [ -f $file ]; then
echo $file
fi
done
else
echo "Error"
fi
```

Teacher's Signature

Output:

sh users.sh

List the users who are logged in

root

Expt. No. 09 Date 10/10/24

Page No. 18

09) Write a shellscript to display list of users currently logged in.

vi users.sh

Program:

```
who "List the users who are logged in"
whoami | sort -u
```

Teacher's Signature

Expt. No. 10 Page No. 19

10) Write a shellscript to read three text files in the current directory and merge them into a single file and returns a file descriptor for the new file.

cat > 1.txt Nagarjuna
cat > 2.txt College of
cat > 3.txt Management Studies

vi merge.sh

Program:

```
if [ "$#" -lt 3 ]; then
echo "Invalid Arguments"
exit 1
fi
cat $1 $2 $3 > new.txt
echo "Files are merged"
echo "------"
echo "The Content of Merged file is"
cat new.txt
```

Output:

```
sh merge.sh
Invalid Arguments
sh merge.sh 1.txt 2.txt 3.txt
Files are Merged
-----
The Content of Merged file is
Nagarjuna
College of
Management Studies.
```

Teacher's Signature