

## Assignment 3

### Linux (ChromeOS)

1. Write a shell program to implement basic arithmetic operations using function.

```
niranjankumbhar@penguin:~$ chmod a+x p.sh
niranjankumbhar@penguin:~$ ./p.sh
Sum= 30
niranjankumbhar@penguin:~$ cat p.sh
#!/bin/bash

function add()
{
    sum=$(( $1 + $2 ))
    echo "Sum= $sum"
}

a=10
b=20

add $a $b
niranjankumbhar@penguin:~$
```

2. Run the command to take backup of file and share your output.

```
niranjankumbhar@penguin:~$ mkdir A
niranjankumbhar@penguin:~$ touch A.txt
niranjankumbhar@penguin:~$ cp --backup A.txt
cp: missing destination file operand after 'A.txt'
Try 'cp --help' for more information.
niranjankumbhar@penguin:~$ cp --backup A.txt ~/TVM
niranjankumbhar@penguin:~$
```

## Assignment 3

### Linux (ChromeOS)

My files > Linux files > TVM					Q	⌵	AZ	⋮
Name ↓	Size	Type	Date modified					
TVM2	–	Folder	Today 13:34					
tvm.txt	0 bytes	Plain text	Today 13:30					
TVM.sh	41 bytes	SH file	Today 13:43					
A.txt	0 bytes	Plain text	Today 13:53					

3. How to take hardlink and softlink of a file in linux? Explain its differences.

```
niranjankumbhar@penguin:~$ ln brk.sh brk1.sh
niranjankumbhar@penguin:~$ ls
30.sh      brk1.sh    cont.sh    err.txt    out.txt
31.sh      brk.sh     demo.sh    git2023    shell1.sh
5new.txt   bullet.sh  developer  monday     shell.sh
5.txt      ceitgit    documents  nk.c       x.txt
niranjankumbhar@penguin:~$
```

```
niranjankumbhar@penguin:~$ ln -s brk.sh brk2.sh
niranjankumbhar@penguin:~$ ls
30.sh      brk1.sh    ceitgit    documents  nk.c       x.txt
31.sh      brk2.sh    cont.sh    err.txt    out.txt
5new.txt   brk.sh     demo.sh    git2023    shell1.sh
5.txt      bullet.sh  developer  monday     shell.sh
niranjankumbhar@penguin:~$ ls -l
total 52
-rwxr-xr-x 1 niranjankumbhar niranjankumbhar 485 Sep 17 13:52 30.sh
-rwxr-xr-x 1 niranjankumbhar niranjankumbhar 105 Sep 18 19:24 31.sh
-rw-r--r-- 1 niranjankumbhar niranjankumbhar 18 Sep 17 12:48 5new.txt
-rw-r--r-- 1 niranjankumbhar niranjankumbhar 18 Sep 17 12:42 5.txt
-rwxr-xr-x 2 niranjankumbhar niranjankumbhar 148 Sep 19 11:48 brk1.sh
lrwxrwxrwx 1 niranjankumbhar niranjankumbhar 6 Sep 19 12:22 brk2.sh -> brk.sh
-rwxr-xr-x 2 niranjankumbhar niranjankumbhar 148 Sep 19 11:48 brk.sh
```

### Explanations

- Hard links point to the data on a storage device, while soft links point to another filename or path.
- Hard links are for files only, while soft links can be for files or directories.
- Hard links cannot cross partitions, while soft links can.
- Hard links share the same inode number as the original file, while soft links have a different inode number.
- Hard links remain valid even if the original file is deleted, while soft links become broken.

## Assignment 3

### Linux (ChromeOS)

4. Run commands (ps,fg, bg, find, du, df, head, tail, less, more) in your terminal and share the results

**Ps** display selection of the active processes

```
niranjankumbhar@penguin:~$ ps
  PID TTY          TIME CMD
  542 pts/1        00:00:00 bash
 1133 pts/1        00:00:00 ps
niranjankumbhar@penguin:~$
```

**Fg** foreground running process

```
niranjankumbhar@penguin:~$ fg
head
```

**bg** background running process

```
niranjankumbhar@penguin:~$ bg
[1]+  head &
```

## Assignment 3

### Linux (ChromeOS)

Find (find file) & du (estimate file space usage)

```
niranjankumbhar@penguin:~$ find brk.sh
brk.sh
niranjankumbhar@penguin:~$ du
0      ../config/procps
8      ../config
4      ../local/state/wireplumber
4      ../local/state
0      ../local/share/nano
0      ../local/share
4      ../local
0      ../git/branches
60     ../git/hooks
4      ../git/info
0      ../git/refs/heads
0      ../git/refs/tags
0      ../git/refs
0      ../git/objects/pack
0      ../git/objects/info
0      ../git/objects
76     ../git
0      ../ceitgit/demo4/.git/branches
60     ../ceitgit/demo4/.git/hooks
4      ../ceitgit/demo4/.git/info
12     ../ceitgit/demo4/.git/refs/heads
0      ../ceitgit/demo4/.git/refs/tags
```

## Assignment 3

### Linux (ChromeOS)

**Df** (display amount of disk space availability)

```
niranjankumbhar@penguin:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/vdc        11534336 1536464   9583952  14% /
none            492         4        488    1% /dev
/dev/vdc        11534336 1536464   9583952  14% /dev/wl0
tmpfs           100         0         100    0% /dev/lxd
run             1420792     24   1420768    1% /dev/.cros_mi
lestone
9p              1989332     744   1988588    1% /mnt/chromeos
tmpfs           1420792     0   1420792    0% /mnt/external
/dev/vdb         58324    58220         0 100% /opt/google/c
ros-containers
fonts           3055272 2420656   634616   80% /usr/share/fo
nts/chromeos
tmpfs           100         0         100    0% /dev/.lxd-mou
nts
devtmpfs        1420536     0   1420536    0% /dev/tty
tmpfs           1420792     0   1420792    0% /dev/shm
tmpfs           568320    116   568204    1% /run
tmpfs           5120       0    5120    0% /run/lock
tmpfs           4096       0    4096    0% /sys/fs/cgrou
p
tmpfs           284156     28   284128    1% /run/user/100
0
niranjankumbhar@penguin:~$
```

**Head** (print 1<sup>st</sup> 10 lines as std output)

```
niranjankumbhar@penguin:~$ head brk.sh
for i in `seq 1 5`
do
    for j in `seq 1 5`
    do
        if(( $j==2 ))
        then
            break 2
        fi
        echo "value of j is $j"
    done
done
niranjankumbhar@penguin:~$
```

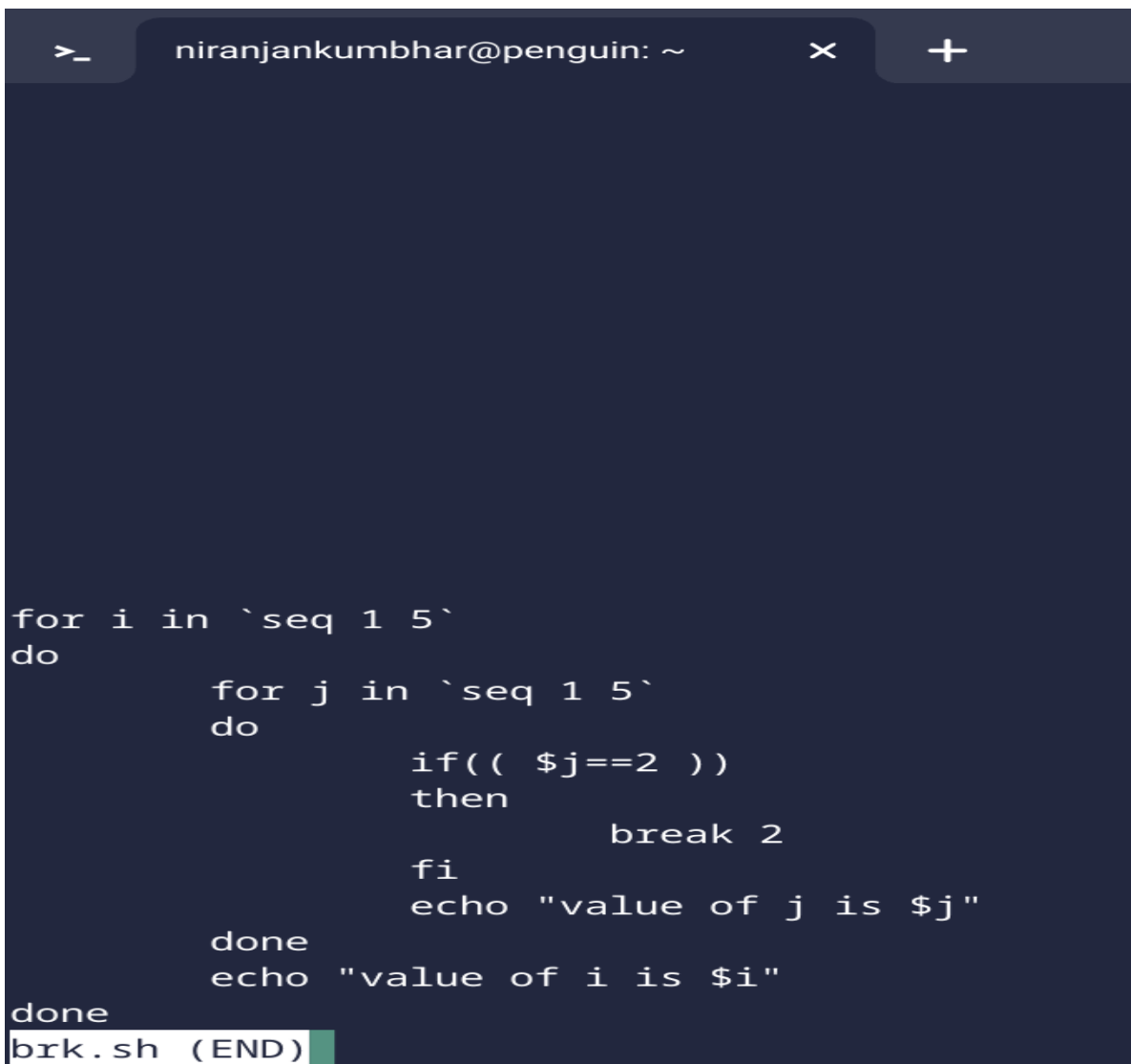
## Assignment 3

### Linux (ChromeOS)

**tail** (print last 10 lines as std output)

```
niranjankumbhar@penguin:~$ tail brk.sh
for j in `seq 1 5`
do
    if(( $j==2 ))
    then
        break 2
    fi
    echo "value of j is $j"
done
echo "value of i is $i"
done
```

**Less**



```
>_ niranjankumbhar@penguin: ~ x +

for i in `seq 1 5`
do
    for j in `seq 1 5`
    do
        if(( $j==2 ))
        then
            break 2
        fi
        echo "value of j is $j"
    done
    echo "value of i is $i"
done
brk.sh (END)
```

## Assignment 3

### Linux (ChromeOS)

More

```
niranjankumbhar@penguin:~$ more brk.sh
for i in `seq 1 5`
do
    for j in `seq 1 5`
    do
        if(( $j==2 ))
        then
            break 2
        fi
        echo "value of j is $j"
    done
    echo "value of i is $i"
done
```

5. With a shell program show the difference of break and continue statements.

<pre>niranjankumbhar@penguin:~\$ vim brk.sh niranjankumbhar@penguin:~\$ chmod a+x brk.sh niranjankumbhar@penguin:~\$ ./brk.sh value of j is 1 niranjankumbhar@penguin:~\$ man brk.sh No manual entry for brk.sh niranjankumbhar@penguin:~\$ cat brk.sh for i in `seq 1 5` do     for j in `seq 1 5`     do         if(( \$j==2 ))         then             break 2         fi         echo "value of j is \$j"     done     echo "value of i is \$i" done niranjankumbhar@penguin:~\$</pre>	<pre>niranjankumbhar@penguin:~\$ cp demo.sh cont.sh niranjankumbhar@penguin:~\$ ./cont.sh value of j is 1 value of j is 1 value of j is 1 value of j is 1 value of j is 1 niranjankumbhar@penguin:~\$ cat cont.sh for i in `seq 1 5` do     for j in `seq 1 5`     do         if (( \$j==2 ))         then             continue 2         fi         echo "value of j is \$j"     done     echo "value of i is \$i" done niranjankumbhar@penguin:~\$</pre>
---	--

## Assignment 3

### Linux (ChromeOS)

Sr. No.	break	continue
1	It terminates the execution of the loop for all the remaining iterations.	It skips the execution of the loop for only the current iteration.
2	It allows early termination of the loop.	It allows early execution of the next iteration.
3	It stops the execution of loops.	It stops the execution of the loop only for the current iteration.
4	The code after the loop which was terminated is continued.	The code in the loop continues its execution skipping the current iteration.

\*\*\*