Operating system

- OS is an **interface** between the user and the hardware components
- OS provides **GUI (Graphical User Interface) or CLI (Command Line Interface)** for the user to perform tasks
- Some **OS only provide CLI** while **others provide both GUI and CLI**
- GUI consists of controls or widgets to interact with the computer using graphical elements such as windows, icons, menus. While in CLI, the user should enter **commands to perform the tasks**
- Overall, **GUI is more user-friendly**, but the **execution speed is higher in CLI**

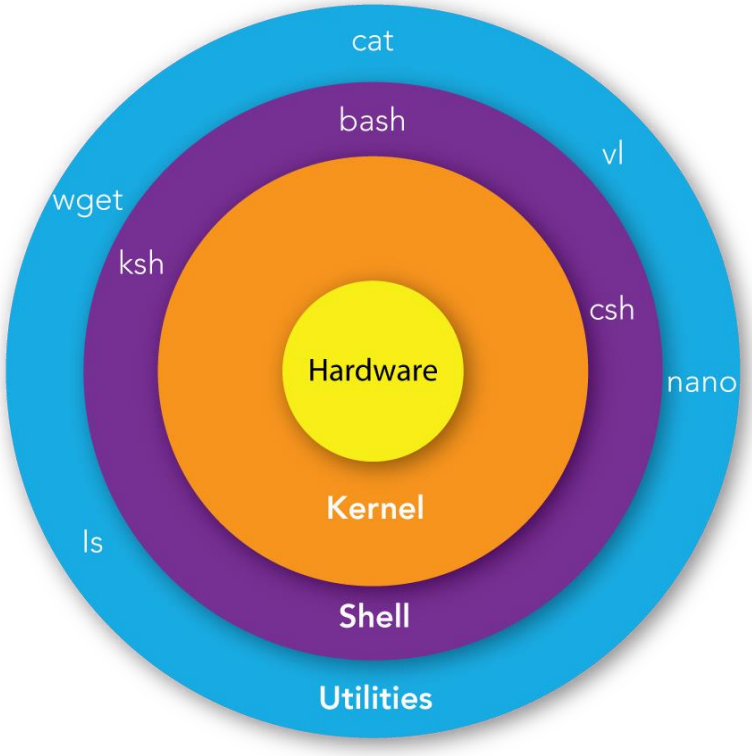
Unix	MacOS, Solaris, AIX (IBM), BSD OS (Berkeley Software Design - Free)
Linux	Debian, Susse, Fedora, RedHat (Paid support), Ubuntu & CentOS (Almalinux & Rockylinux)
Windows	Microsoft

Shell

A shell is the interface that allows the users to communicate with the kernel

Different types of shell

- Bourne shell (sh)
- C shell (csh)
- TC shell (tcsh)
- Korn shell (ksh)
- Bourne Again SHell (bash)

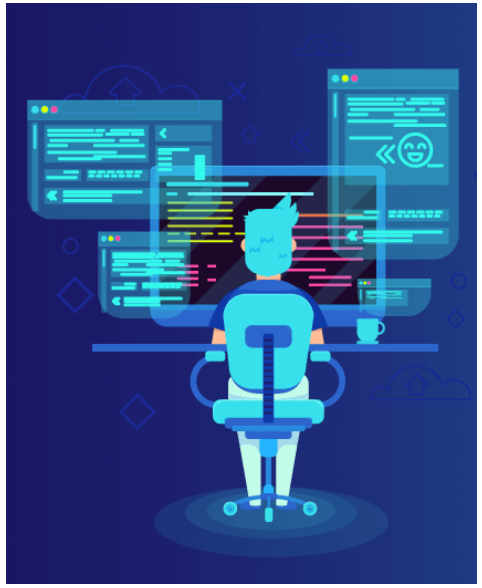


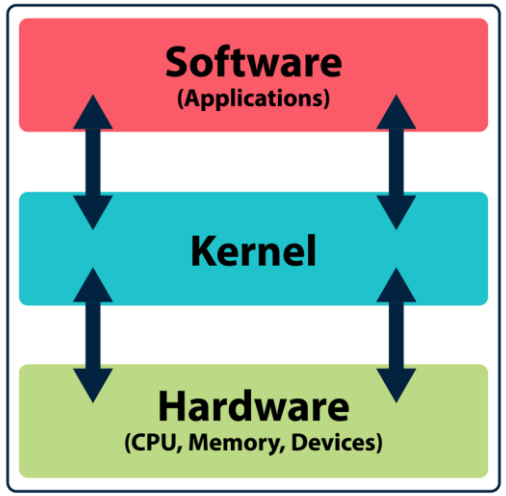
A terminal used by shell to get input-output

Dr. S. Mohan Kumar
Research Associate

Command

- Command is an instruction to the computer which interprets it to perform a specific task or a directive to the command-line interface such as Shell
- Linux & Unix commands are similar because Linux is based on Unix
- Linux kernel code was completely written from scratch (Linus Torvalds & team). Designed in a way, acts like Unix but it does not have the original Unix code
- Kernel is the main part of the OS and responsible for translating the command into something that can be understood by the computer



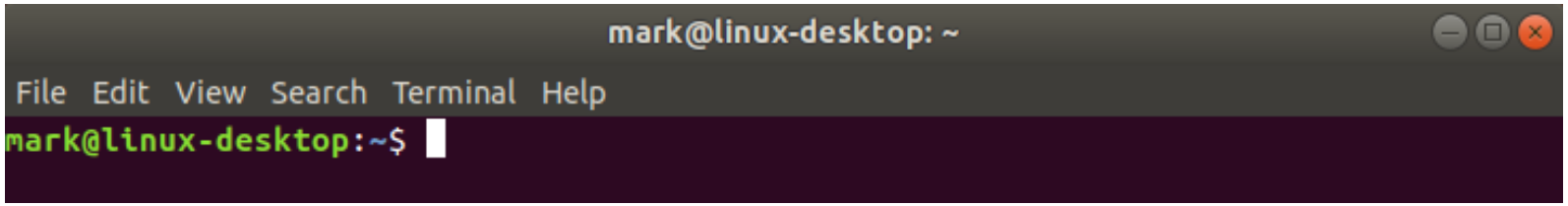


Linux Command Line - Terminal

- Upon launching the CLI (Terminal), there will be a dollar sign **"$"**

  ("%" for the C-Shell)

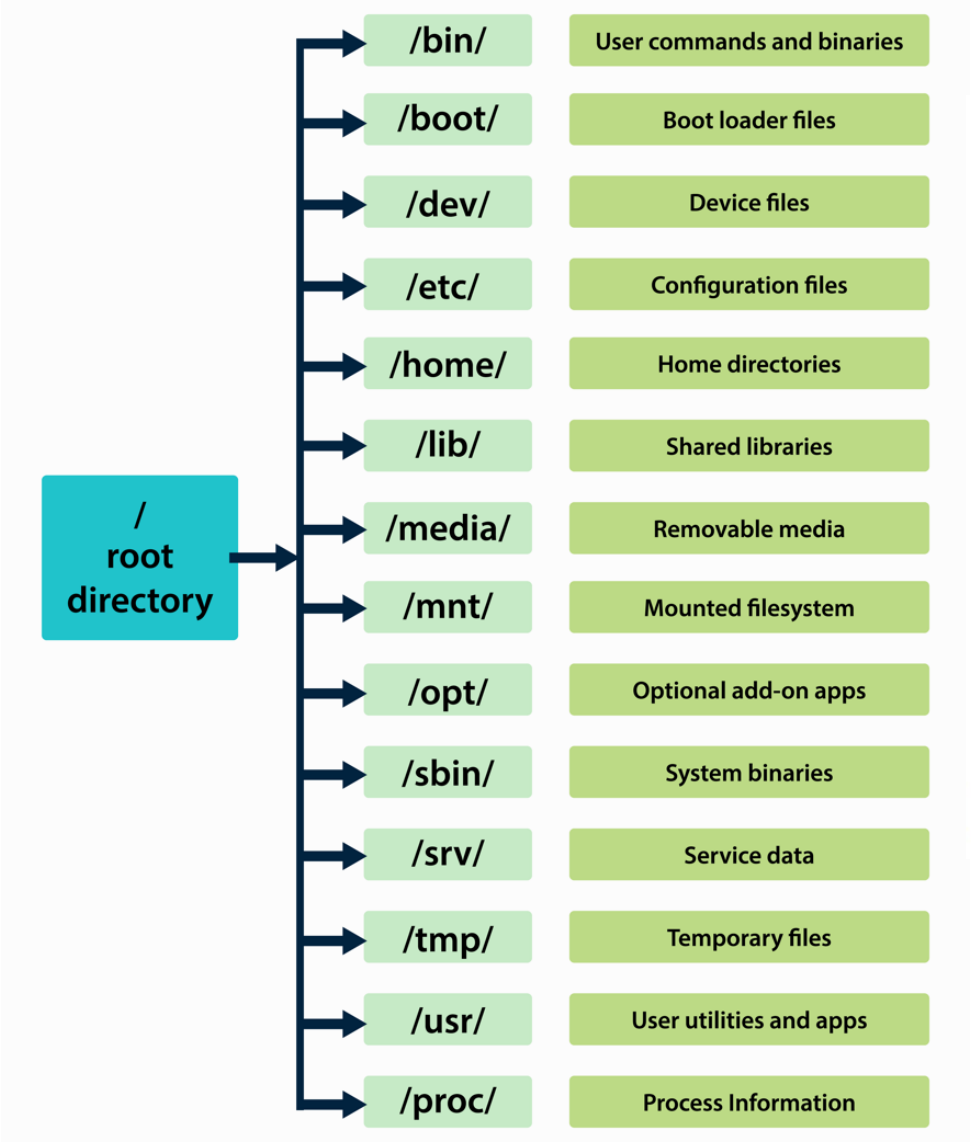
- You can type the command next to the "$" sign

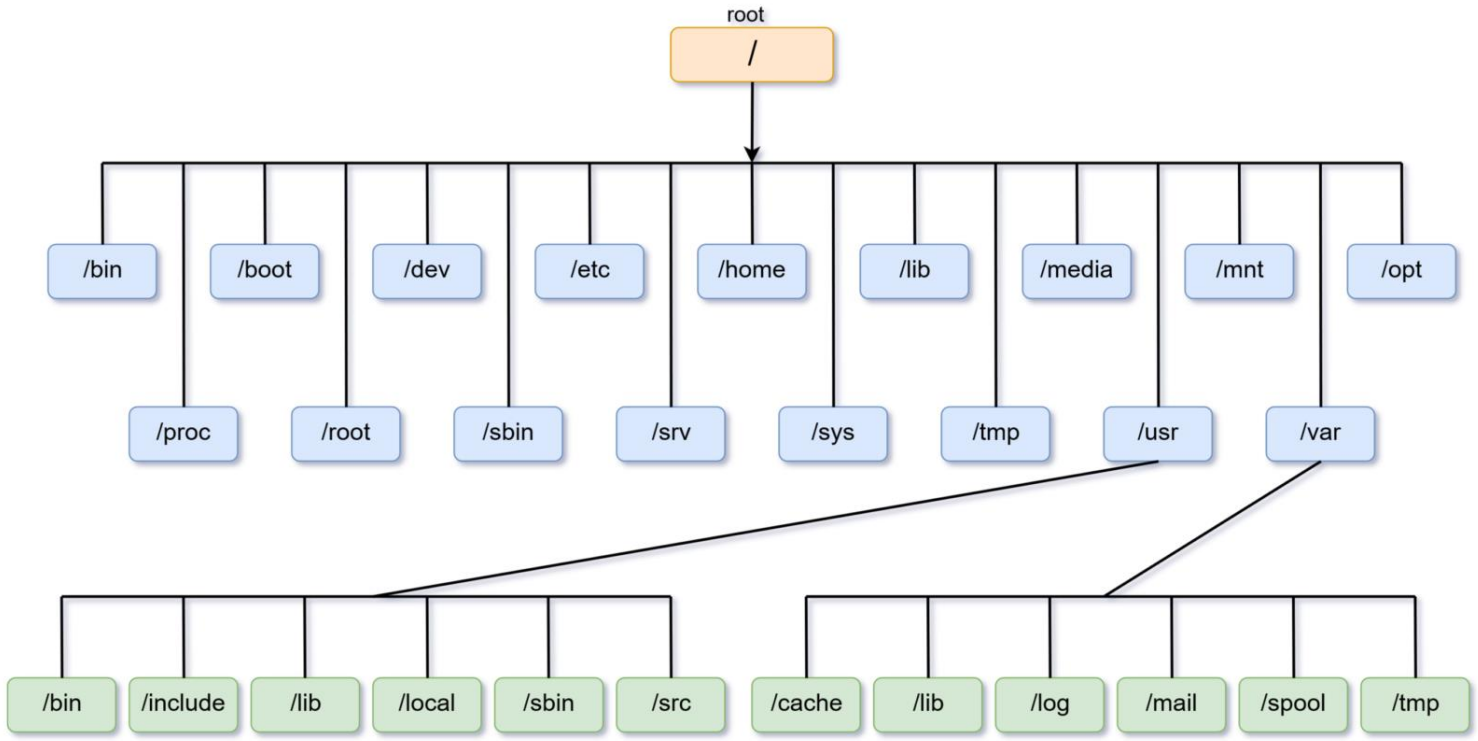


- Many commands have their own set of options, also called **flags or options**
- Flags are added with the commands to enhance their basic functionality

~$ ls -l

Linux file structure & directory





Dr. S. Mohan Kumar
Research Associate

pwd (**p**resent **w**orking **d**irectory)

- Options for command is start with an **"-"** followed by a single word in small or caps form and may be combination of letters
- Check and find options by using **"man"** command like **"$man pwd"**
- Options for $pwd -L or $pwd –P (includes links)

cd (changing directory)

$cd Desktop	- (or full path)
$cd /home/user/Desktop	
$cd ..	- backwards
$cd ../Downloads	- back & next
$cd /	- root directory

- Either change to the next folder by knowing through ls command
- or provide full path if you know already

ls listing files & directories (folders)

```
$ls
```

options are

```
$ls -l
$ls -R
$ls -a
$ls -al
$ls -lh
$ls -lrt
$ls -1
```

Dr. S. Mohan Kumar
Research Associate

File permission

File type — Permission classes

User — Group — Other

drwxrwxrwx

-rwxrwxrwx

Read — Write — Execute

rwx

- for file
d for directory

u for user
g for user group
o for others

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

chmod 777

rwx|rwx|rwx

Owner|Group|Others

Numeric mode: 6 4 4

110100100

Symbolic mode: rw-r--r--

Dr. S. Mohan Kumar
Research Associate

File permission

$ls -l reveals permission status -rwxr-xr-x@

$chmod +x filename or chmod 755 filename

both are same for execution

$chmod 666 filename for write permission

$chmod 777 filename - all permission all grp 7 to 1 is used for changing permissions

- 4-read(r)
- 2-write(w)
- 1-execute(x)

sudo + command

 # password necessary to get root permission

mkdir - creating directory or folder

$mkdir foldername

$mkdir foldername/acrc

$mkdir /home/user/Desktop/foldername (full path)

- Can be created from any location to any location

cat - creation using cat command

```
$cat > filename.txt
```

Then type the content into the filename.txt
Press ctrl+d (quits writing)

```
# File reading
$cat filename.txt
$cat file1 file2 > file3          # appends two files
$touch filename.txt               # only creates file
```

Reading long files

more fname

use space bar read page by page

less fname

use space bar then type "q” to quit

Reading top portion of file

head fname (use longfile.txt for all options)

by default 10 lines at top, option

head -5 fname

to read only 5 lines

Reading bottom portion of file

tail fname (use longfile.txt for all options)

by default 10 lines from bottom,

tail -5 fname

option -5 to read only last 5 lines

tail -f fname

force to see last lines when outputs are driven by process (show wrf run)

tail -n +2 fname ## prints without header

r. S. Mohan Kumar
Research Associate

Moving files to different locations / names

mv fname /home/admin/Downloads

from one place to other or rename to another

mv -i file1 file2 - asks permission to overwrite

mv -f file1 file2 - force to overwrite

mv -v n.txt m.txt - shows what happens

Copying files and folders

```
    cp file1 file2
# copy to  another name
    cp file1 /home/admin/Downloads/
# copy to another place same name
    cp -r folder1 folder2
# recursive copying used for directory / folder
    cp -p file1 file2
#  preserves the mode, ownership etc
    cp -i file1 file2
# confirmation before overwriting
```

Dr. S. Mohan Kumar
Research Associate

Deleting / removing files and folders

```
rm fname
rm -f file
# will not asks for permission
rm -r folder
# recurresive deleting of directories
rm -i fname
# interactive removal
```

Cut, paste & column removal

```
    cut -c 1,2,3 file or cut -c 1-3 or cut -c -3
# all are same  -b for bytes works same way
    colrm 3 6 < three-column.txt
# start end < input
    paste file1 file2
# side by side wider spacing
    paste -d ” " file1 file2
# side by side wider spacing (vertical)
    paste -s file1, file2
# one below the other (horizontal - tranpose)
```

Dr. S. Mohan Kumar
Research Associate

difference & common between two files

diff file1 file2

comm file1 file2

linking file

ln -s creating symbolic link one place to other or from other place to current

ln -s /Users/Jag/Desktop/jag.txt .

note the dot at the end

process management

$ps

to know the process running

$kill pid & kill all

killing a running job or killing all process

top

to display running process & cpu usage

Check status of a service (**works only in Linux)

service sshd status

Check the status of all the services.

service --status-all

Stop a service

service sshd stop

restart the service

service sshd restart

Shutdown system & turn power off immediately

```
shutdown -h now
```

Shutdown the system after 10 minutes.

```
shutdown -h +10
```

Reboot the system using shutdown command

```
shutdown -r now
```

system information

uname -a	- show kernel information
whoami	- Who you are logged in as
w	- Display who is on line
whereis (app)	- where is the application (cmd)
which (app)	- show which application (cmd)
clear	- clears terminal (just pushes up)

system information

history - history of commands used via terminal

ulimit -a - provides all information

ulimit -s unlimited - stack size important when CPU is to be fully utilized

search

locate fname or word - find its location

find fname

find -iname "MyCProgram.c"

Using grep cmd for search & print (egrep fgrep)

```
grep "word" fname
ls | grep learn
```

Print matched line, along with 3 lines

```
grep -A 3 88 file1.txt  - prints after 88
grep -B 3 88 file1.txt  - prints before 88
grep -C 3 88 file1.txt  - prints before & after 88
```

Search for a given string in all files recursively

```
grep -r "ramesh" *
```

disk information

df -h - disk available and space availability

du -h - directory size recursively

time & date

date - current date and time

Formats in date

```
date "+%Y:%m:%d %H:%M:%S"
date "+%H:%M:%S   %d/%m/%y"
```

In MACOS

```
yday=$(expr `date +%Y%m%d` - 86400)
echo $yday
```

date --date="yesterday" in **centos linux**

Dr. S. Mohan Kumar
Research Associate

using calender cal & ncal

cal - current month calendar

cal apr 1956 - for April 1956

cal 1956 - for all months of 1956

cal -3 mar 1956 – March previous & next

cal -j 1956 - Julian days

Diff cal & ncal is only layout

At present 1..9999

compressions

1. tar -cf file.tar file - Create tar archive file.tar
2. tar -xf file.tar - Extracts from file.tar
3. tar -czf file.tar.gz files - Create Gzip compression
4. tar -xzf file.tar.gz - Extract a tar using Gzip
5. tar -cjf file.tar.bz2 files - Create Bzip2 compression
6. tar -xjf file.tar.bz2 - Extract a tar using Bzip2

compressions continued

7. gzip file - Compresses and renames to file.gz
8. gzip -d file.gz - Decompresses file.gz back to file

   gzip -l *.gz - Display compression ratio
9. zip file.zip file - zip compression
10. unzip -l file.zip - un-compression
11. bzip2 test.txt - creates a *.bz2 compressed file
12. bzip2 -d test.txt.bz2 - To un-compress a *.bz2 file

Network information

$hostname

$ping $hostname

$ping yahoo.com or ping 117.232.96.116

$ifconfig - to know the network connections

Download commands

```
$wget URL
$wget cc.iiti.ac.in/docs/linuxcommands.pdf
$curl -O  # mac users
$cat url-list.txt | xargs wget –c  # download list
```

Remote connection

$ssh user@117.232.96.116

- First ask for permanent storage and then after "yes" asks password and enter password not visibly seen

File copy to remote location using scp

$scp file user@117.232.96.116 :/User/Desktop/PDA - asks password

$scp -r # for copying directory use option -r

```
# shell script
cat > start.sh
echo “My first shell script”
mkdir /home/admin/Desktop/scripts
cp /home/admin/Desktop/scripts/acrc1.txt
echo “Finished copying File”
“ctrl d” to quit
 ls -l start.sh        –> pl check permissions
chmod +x  start.sh
./start.sh
```

shortcuts

1. ctrl+c Halts the current command
2. ctrl+z Stops the current command, resume with fg in the foreground or bg in the background
3. ctrl+d Logout the current session, similar to exit
4. ctrl+w Erases one word in the current line
5. ctrl+u Erases the whole line
6. ctrl+r Type to bring up a recent command
7. !! Repeats the last command
8. exit - Logout the current session

Dr. S. Mohan Kumar
Research Associate

text processing

nl file - numbers all the lines

nl -s: file : after the number nl -s:" " : with space

nl -n rz right justified

nl -b nl -ba nl -bt etc

a = Number all lines

t = Number only nonempty lines

n = Number no Lines

p = Number only lines that contain a match for the basic regular expression

nl -v 77 number start at 77

Dr. S. Mohan Kumar
Research Associate

printing sequential numbers

seq 1 100 or seq 100; or seq 5 10;

seq 5 2 10 (middle incr) – for increment print

```
seq -f "mohan%02g" 4
seq -f "mohan%02g" 2 4
seq -f "mohan%02g" 10 10 40
seq -s "  " 10;      seq -s "  " 2 10;
seq -s "  " 5 2 10; - horizontal
seq -w 20 seq -w 99 101  seq -w 1 10 50
# -w 0 padding is done
```

Dr. S. Mohan Kumar
Research Associate

sort for sorting the file

sort -n file - numerical sorting

sort -k 3 file - sorting based on 3rd column

sort -r file - reverse sorting

sort -b file - Ignore blanks at the start of the line.

uniq command

uniq file - only unique numbers or values;

uniq -c file - number of occurrence of one value

Word count commands

wc -l <filename> prints line count

(note: if last line does not have \n, it will not be counted)

wc -l *.txt

wc -c <filename> prints the byte count

wc -m <filename> prints the character count

wc -w <filename> prints the word count

text editors

vi

vim

nano

gedit etc..

Demo each one

Data analytics using sed & awk

```
# To print line containing word or number ie search & print
sed -n '/21.25/p' sample-data/1971-01-cru-sindia.txt
# To count line containing word or number
sed -n '/21.25/p' sample-data/1971-01-cru-sindia.txt | wc -l
# To sort column 5 rain & find maximum & minimum values
fname= sample-data/ 1971-01-cru-sindia.txt
sed -n '/21.25/p’ $fname | sort -k 4
sed -n '/21.25/p' $fname | sort -k 4 | tail -1
sed -n '/21.25/p' $fname | sort -k 4 | head -1
```

```
## printing required lines
sed q sample-data/1971-01-cru-sindia.txt
sed 10q sample-data/1971-01-cru-sindia.txt
sed '$!d' sample-data/1971-01-cru-sindia.txt ||
or use sed -n '$p'
sed -n '8,12p' sample-data/197101-mpicru.txt ||
or use sed '8,12!d'
sed -n '52p' sample-data/197101-mpicru.txt ||
or use sed '52!d'
```

beginning at line **3, print every 7th line**

```
sed -n '3,${p;n;n;n;n;n;n;}' sample-data/1971-01-cru-sindia.txt
```

From to print

```
sed -n '/23.25/,/24.25/p' sample-data/1971-01-cru-sindia.txt
```

Multiple lines print

```
for lno in 1,40p 41,80p 81,120p; do  sed -n "$lno" sample-data/1971-01-cru-sindia.txt; done | wc -l
```

deleting unwanted lines

sed '1,10d' - delete first 10 lines

sed '$d' - delete the last line of a file

sed '/23.25/,/24.25/d’ - delete from to lines

sed '/pattern/d’ - delete lines matches pattern

sed '/^$/d' - delete ALL blank lines from a file

sed '/regexp/d’ - print only lines which do NOT match regexp OR sed -n '/regexp/!p'

python utility for changing xlsx to csv

xlsx2csv Koradachery.xlsx > Koradachery.csv

To install xlsx2csv

pip install xlsx2csv (pip not available install pip)

Installing PIP if not available

sudo yum install python-pip

If no repository try installing repository

sudo yum install epel-release

for loop

```
for i in {1981..2000}
do
echo $i
done
```

OR for i in {1981..2000}; do echo $i; done

simple for loop one variable & two variable

nested loop

RUN simple-arr.sh, simult-arr.sh & multi-arr.sh

Dr. S. Mohan Kumar
Research Associate

bash while loop syntax

```
while [ condition ]
do
   command1
   command2
   command3
done
```

RUN while.sh