

Shell and Shell Scripting Session No.4.1

- **Ps** command is used for checking the process running in the Linux OS.
Every process has some unique ID known as process ID

```
[root@ip-172-31-6-112 ~]# ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	41656	5360	?	Ss	Sep10	0:04	/usr/lib/systemd/systemd - switched-ro
root	2	0.0	0.0	0	0	?	S	Sep10	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	Sep10	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	Sep10	0:00	[rcu_per_gp]
root	6	0.0	0.0	0	0	?	I<	Sep10	0:00	[kworker/0:0H-cv]
root	8	0.0	0.0	0	0	?	I<	Sep10	0:01	[kworker/0:1H-cv]
root	9	0.0	0.0	0	0	?	I<	Sep10	0:00	[mm_percpu_wq]
root	10	0.0	0.0	0	0	?	S	Sep10	0:00	[rcu_task_ruid]

- Ctrl + c to terminate the program which is running in the foreground
- **&** is used for running a command in the background

```
[root@ip-172-31-6-112 ~]# cat &
[1] 11184
[root@ip-172-31-6-112 ~]#
```

- Kill -9 is used for terminating the process running in the background
- Any program running in the background is also known as Daemon
- **Jobs** command shows all the processes running in the background
- **Pgrep** shows the process id of the program

```
[root@ip-172-31-6-112 ~]# pgrep cat
11184
[root@ip-172-31-6-112 ~]# kill -9 `pgrep cat`
```

- **Fg** command is used for bringing the background process into the foreground

```
[root@ip-172-31-6-112 ~]# jobs
[1]+  Running                  ping 8.8.8.8 > /dev/null &
[root@ip-172-31-6-112 ~]# fg 1
ping 8.8.8.8 > /dev/null
^C[root@ip-172-31-6-112 ~]# jobs
[root@ip-172-31-6-112 ~]#
```

- **Ctrl + z** is used for pausing/halting the process in the foreground
- **Trap** command has the capability to keep on listening to the signals

```
[root@ip-172-31-6-112 ~]# lw()
> {
> echo "i m lw"
> }
[root@ip-172-31-6-112 ~]#
[root@ip-172-31-6-112 ~]# lw
i m lw
[root@ip-172-31-6-112 ~]# trap lw 2
[root@ip-172-31-6-112 ~]#
[root@ip-172-31-6-112 ~]# ^Ci m lw
[root@ip-172-31-6-112 ~]# |
```

- **Exit** function is used for killing the shell
- Whenever the command is run either successfully or unsuccessful it always gives the exit code
- Creating exit codes in the script

```
myclear(){
echo -e "\nok bye see u next time...\n"

rm -f /tmp/pass.txt

exit 1
}
```

- **Trap -l** to list all the signals

```
[root@ip-172-31-6-112 ~]# trap -l
1) SIGHUP          2) SIGINT          3) SIGQUIT         4) SIGILL          5) SIGTRAP
6) SIGABRT         7) SIGBUS         8) SIGFPE          9) SIGKILL         10) SIGUSR1
11) SIGSEGV        12) SIGUSR2        13) SIGPIPE        14) SIGALRM        15) SIGTERM
16) SIGSTKFLT      17) SIGCHLD        18) SIGCONT        19) SIGSTOP        20) SIGTSTP
21) SIGTTIN        22) SIGTTOU        23) SIGURG         24) SIGXCPU        25) SIGXFSZ
26) SIGVTALRM      27) SIGPROF        28) SIGWINCH       29) SIGIO           30) SIGPWR
31) SIGSYS         34) SIGRTMIN       35) SIGRTMIN+1     36) SIGRTMIN+2     37) SIGRTMIN+3
38) SIGRTMIN+4     39) SIGRTMIN+5     40) SIGRTMIN+6     41) SIGRTMIN+7     42) SIGRTMIN+8
43) SIGRTMIN+9     44) SIGRTMIN+10    45) SIGRTMIN+11    46) SIGRTMIN+12    47) SIGRTMIN+13
48) SIGRTMIN+14    49) SIGRTMIN+15    50) SIGRTMAX-14    51) SIGRTMAX-13    52) SIGRTMAX-12
53) SIGRTMAX-11    54) SIGRTMAX-10    55) SIGRTMAX-9     56) SIGRTMAX-8     57) SIGRTMAX-7
58) SIGRTMAX-6     59) SIGRTMAX-5     60) SIGRTMAX-4     61) SIGRTMAX-3     62) SIGRTMAX-2
63) SIGRTMAX-1     64) SIGRTMAX
```

- **While** loop in shell scripting

```
[root@ip-172-31-6-112 ~]# page=1
[root@ip-172-31-6-112 ~]# while [ $page -le 10 ]
> do
> echo $page is done
> page=$(( $page + 1 ))
> done
1 is done
2 is done
3 is done
4 is done
5 is done
6 is done
7 is done
8 is done
9 is done
10 is done
[root@ip-172-31-6-112 ~]# |
```

- **Sleep** command is used for creating the delay in command

```
[root@ip-172-31-6-112 ~]# sleep 5
[root@ip-172-31-6-112 ~]# while date
> do
> sleep 1
> done
Sun Sep 11 09:39:13 UTC 2022
Sun Sep 11 09:39:14 UTC 2022
Sun Sep 11 09:39:15 UTC 2022
Sun Sep 11 09:39:16 UTC 2022
Sun Sep 11 09:39:17 UTC 2022
Sun Sep 11 09:39:18 UTC 2022
Sun Sep 11 09:39:19 UTC 2022
Sun Sep 11 09:39:20 UTC 2022
```

- **Break** is a keyword used for breaking the while loop

```
[root@ip-172-31-6-112 ~]# while [ $x -le 5 ]
> do
> if [ $x -eq 3 ]
> then
> break
> fi
>
> echo $x
> x=$(( $x + 1 ))
> done
1
2
[root@ip-172-31-6-112 ~]# |
```

- **Getopt** command has the capability to get options from the command line

- Creating short options

```
[root@ip-172-31-6-112 ~]# getopt "a:l:s:" -a -l=5 -s=7
-a -l =5 -s =7 --
[root@ip-172-31-6-112 ~]#
```

- Creating long options

```
[root@ip-172-31-6-112 ~]# getopt -l "name:,age:,city" -o "n:a:c" -- --name=vimal
--name 'vimal' --
[root@ip-172-31-6-112 ~]# getopt -l "name:,age:,city" -o "n:a:c" -- -n=vimal
-n '=vimal' --
```

- Creating options in the script

à Script

```
#!/bin/bash

name="vimal"      # -n
phone=11111       # -p
age=12            # -a

#read -p "enter ur choice : " ch

while getopts "npa" ch
do
    case $ch in
        n)          echo $name;;
        p)          echo $phone;;
        a)          echo $age;;
        *)          echo "idk";;
    esac
done
```

à Output

```
[root@ip-172-31-6-112 ~]# ./myodata.sh -a
12
[root@ip-172-31-6-112 ~]# ./myodata.sh -p
11111
[root@ip-172-31-6-112 ~]# ./myodata.sh -n
vimal
[root@ip-172-31-6-112 ~]# ./myodata.sh -n -a
vimal
12
[root@ip-172-31-6-112 ~]#
```

- Option argument in script
àScript

```
#!/bin/bash
name="vimal"      # -n
#phone=11111      # -p
age=12            # -a

#read -p "enter ur choice : " ch

while getopts "np:a" ch
do
    case $ch in
        n)          echo $name;;
        p)          echo $OPTARG;;
        a)          echo $age;;
        *)          echo "idk";;
    esac
done
```

Output

```
[root@ip-172-31-6-112 ~]# ./myodata.sh -p 222
222
[root@ip-172-31-6-112 ~]# ./myodata.sh -p 33333
33333
[root@ip-172-31-6-112 ~]# ./myodata.sh -p 33333 -a
33333
12
```