PROCESS MANAGEMENT – PS ,TOP AND SCHEDULING PRIORITIES(NICE,RENICE)

What is a Process

An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process. A process is compiled source code that is currently running on the system.

- All processes have a process id or PID
- Every process has a parent process (with a PPID). The child process is often started by the parent process.
- When a process stops running, the process dies, when you want a process to die, you kill it
- Processes that start at system startup and keep running forever are called daemon processes or daemons. These daemons never die
- When a process is killed, but it still shows up on the system, then the process is referred to as zombie. You cannot kill zombies, because they are already dead

Types of Processes in Linux

- Foreground Processes: They run on the screen and need input from the user. For example Office Programs
- Background Processes: They run in the background and usually do not need user input. For example Antivirus.

ps command

1. To list the processes
   a. top
   b. ps -efa
   c. ps -ef

2. To list all running processes
   a. ps -aux |less
   b. ps -aux | less
3. To list a specific user's processes
   a. ps -u username
4. To get the pid of a process
   a. pidof processname/cmdname
   b. pidof sleep

5. To print the process tree
    a. pstree pidofprocess

```
[root@machine1 ~]# pstree 1589

bash---pstree

       +-sleep
```

6. To search for a process
    a. pgrep sshd → gives the process id (similar to pidof cmd)
    b. pgrep -l sshd → gives PID along with name
    c. ps -ef | grep sshd (only that process details is listed)

7. ps -f
    a. ps -f 1
    b. ps -f 3576

```
[root@machine1 ~]# ps -f
UID        PID   PPID  C STIME TTY          TIME CMD
root      1589   1588  0 09:25 pts/0    00:00:00 -bash
root      1971   1589  0 09:26 pts/0    00:00:00 sleep 1000
root      1972   1589  0 09:26 pts/0    00:00:00 ps -f
```

8. To kill a process
    a. kill PID
    b. kill -9 PID
    c. kill -15 PID (force kill)
    d. kill – l (lists all the signals sent to kill cmd)
    e. pkill processname (kills a process by name)
        i. pkill sleep
        ii. pkill bash
    f. killall processname (kill all the processes by name)
        i. killall sleep (all sleep processes will be killed)

9. To customize the output of ps -aux
   a. ps -aux

OUTPUT

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
```

a. ps -eo user,cpu,cmd ----> option -e is used for editing and option -o is used to user define. The o option allows you to specify which columns are displayed when running the ps command.
b. ps eo uid,user,pid,ppid,stat,cpu,cmd
c. ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan
d. ps axo uid,user,pid,vsz,pcpu,cmd

-a -> Shows information about all users.

-x -> Shows information about processes without terminals.

-e -> Displays all processes.

-u -> expands the output to include additional info like cpu usage,memory etc

-f -> stands full-format listing, which provides detailed information about the processes.

Child Processes:

- The processes which gets created by another process it's parent process known as child process.
- In our above example, the sleep process with PID 1971 is a child process of the bash process with PID 1589.

Daemon process:

- The system related background running processes are called Daemon Processes.
- When you see a process running with a ? mark in sixth column (TTY field), that's a daemon process.

```
[root@machine1 ~]# ps -ef

    UID        PID    PPID  C STIME TTY        TIME CMD
    root         1       0  0 09:24 ?       00:00:02 /usr/lib/systemd/systemd --switched-root --
    system --deserialize 18
    root         2       0  0 09:24 ?       00:00:00 [kthreadd]
    root         3       2  0 09:24 ?       00:00:00 [rcu_gp]
    root         4       2  0 09:24 ?       00:00:00 [rcu_par_gp]
```

Zombie processes:

- At times there are processes which are already dead but still shows up in process list are called Zombie processes.
- These processes can be found while doing ps listing, the process with a Z state are zombie processes.
- They don't consume any CPU resources.

The process status can be any of the following:

R: Running - Running process which is executed on cpu
S: Sleeping - Sleeping or waiting process. This process is in waiting stage for a signal.D: Uninterruptible sleep - Uninterruptable. This process is in sleeping process which will not respond to a signal.

T: Traced (stopped state) - Stopped process or suspended process. This process can be continued by another signal to return to running process.

Z: Zombie - Zombie process. This is a child process sending its parent process an exit signal.

TASK_RUNNING(R) - task (process) currently running

TASK_INTERRUPTABLE (S) - process is sleeping but can be woken up (interrupted)

TASK_UNINTERRUPTABLE(D) - process is sleeping but can not be woken up (interrupted)

TASK_ZOMBIE (Z) - process terminated but its status was not collected (it was not waited for)

TASK_STOPPED(T) - process stopped by a debugger or job control

SCHEDULING PRIORITIES

- Every process have scheduling priorities.
- The value which is assigned for scheduling the priorities is called "NICE VALUE"
- Nice value range is --20 to 19
- The default nice value of a process is 0.
- High nice value process means low CPU usage.(19)
- Low nice value means High CPU usage.(-20)
- The root user can increase or decrease the priorities of a process.
- The normal user can only increase the priority of a process.

1. ps -eo user,pid,ni,cmd → customize the output to display the nice value
2. ps -l →displays output along with the nice value of a processes
3. ps -l PID → Particular process details are displayed.
    a. ps -l 3834

```
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY        TIME CMD
0 S  1000    3834    3772  0  80   0 - 55238 hrtime pts/0      0:00 sleep 100
```

Here NI is nice value

To schedule priorities.

1) For running process
    a. renice nicevalue PID
        i. renice -3 15232

2) For new process

    a. nice -n nicevalue command

    i. nice --n -7 cat
    ii. nice -n -20 seq 1000000000000000000000 > /dev/null

top COMMAND

Terminologies

us: Amount of time the CPU spends executing processes for people in “user space.”

sy: Amount of time spent running system “kernel space” processes.

ni: Amount of time spent executing processes with a manually set nice value.

id: Amount of CPU idle time.

wa: Amount of time the CPU spends waiting for I/O to complete.

hi: Amount of time spent servicing hardware interrupts.

si: Amount of time spent servicing software interrupts.

st: Amount of time lost due to running virtual machines (“steal time”).

PID: Process ID.

USER: The owner of the process.

PR: Process priority.

NI: The nice value of the process.

VIRT: Amount of virtual memory used by the process.

RES: Amount of resident memory used by the process.

SHR: Amount of shared memory used by the process.

S: Status of the process. (See the list below for the values this field can take).

%CPU: The share of CPU time used by the process since the last update.

%MEM: The share of physical memory used.

TIME+: Total CPU time used by the task in hundredths of a second.

COMMAND: The command name or command line (name + options).

1. Changing the Summary Contents - l,1,t,m
2. k - Kill a Process
3. z - Color and Highlighting
4. Sorting by Columns

P: The %CPU column.
M: The %MEM column.
N: The PID column.
T: The TIME+ column.

5. c - See the Full Command Line
6. u - See Processes for a Single User
7. n - Set How Many Processes to Display
8. r - Renice a Process
9. d or s to change the delay time
10. f – to select the columns to display (press d to sel/unsel)
11. o -to filter the output

System monitoring

1. uptime
2. w
3. df -h
4. free -h
5. cat /proc/cpuinfo
6. cat /proc/meminfo
7. top
8. lscpu
9. nproc

Summary

COMMAND	DESCRIPTION
bg	To send a process to the background
fg	To run a stopped process in the foreground
top	Details on all Active Processes
ps	Give the status of processes running for a user
ps PID	Gives the status of a particular process
pidof	Gives the Process ID (PID) of a process
kill PID	Kills a process
nice	Starts a process with a given priority
renice	Changes priority of an already running process
df	Gives free hard disk space on your system
free	Gives free RAM on your system