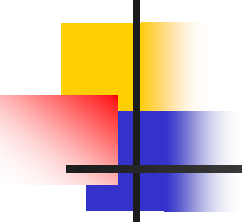




# Process Scheduling

---

- 
- 
- Process is a program that has been loaded from a long-term storage device, usually a hard disk drive, into system RAM and is currently being processed by the CPU on the motherboard
  - Init is the first process started at boot time , always has PID 1.
  - Except init every process has a parent.
  - Pstree Shows the process ancestry for all the process running on the system.



# What Exactly Is a Process?

---

- **Binary executables** : created as a text file using a programming language, such as C or C++. The text file was then run through a compiler to create a binary file that can be processed by the CPU.
- **Internal shell command** : rpm, cd, mkdir, ..
- **Shell scripts** :



# Types of Processes

---

- User Processes: Some processes are created by the end user when he or she executes a command from the shell prompt or through the X Windows graphical interface.
- System processes or daemons: Web server, an FTP server, a file service such as Samba, a print service such as CUPS, a logging service, and so on.
- Ex: cupsd, hald, sshd,...



# The Heredity of Linux Processes

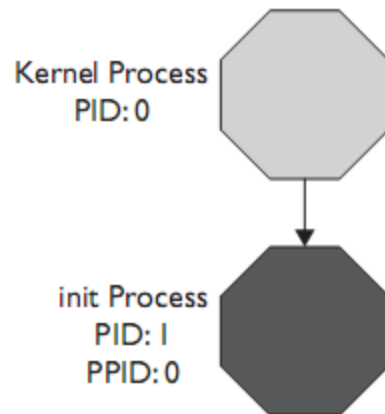
---

- **Process ID (PID)** Number This is a number assigned to each process that uniquely identifies it on the system.
- **Parent Process ID (PPID)** Number This is the PID of the process' parent process.



# Parent Process ID

---





# Process states

---

- **state** values:
  - **TASK\_RUNNING** (executing on CPU or runnable).
  - **TASK\_INTERRUPTIBLE** (waiting on a condition: interrupts, signals and releasing resources may “wake” process).
  - **TASK\_UNINTERRUPTIBLE** (Sleeping process cannot be woken by a signal).
  - **TASK\_STOPPED** (stopped process e.g., by a debugger).
  - **TASK\_ZOMBIE** (terminated before waiting for parent).



# Viewing Running Processes

---

- Using top
- Using ps
- gnome-system-monitor (gui)





# Using top

---

- Top is a powerfull tool
- Dynamic and refreshing tool
- We can see combination of commands
- ■ USER The name of the user that owns the process.
- ■ PR The priority assigned to the process. (We'll discuss process priorities later in this chapter.)
- ■ NI This is the nice value of the process. (We'll talk about what this means later in this chapter.)
- ■ VIRT The amount of virtual memory used by the process.
- ■ RES The amount of physical RAM the process is using (its resident size) in kilobytes.



# Using top...

---

- ■ **%CPU** The percentage of CPU time used by the process.
- ■ **%MEM** The percentage of available physical RAM used by the process.
- ■ **TIME+** The total amount of CPU time the process has consumed since being started.
- ■ **COMMAND** The name of the command that was entered to start the process.



# Top – 1<sup>st</sup> Row

---

- This first line indicates in order:
  - current time (11:37:19)
  - uptime of the machine (up 1 day, 1:25)
  - users sessions logged in (3 users)
  - average load on the system (load average: 0.02, 0.12, 0.07) the 3 values refer to the last minute, five minutes and 15 minutes.

```
top - 11:37:19 up 1 day, 1:25, 3 users, load average: 0.02, 0.12, 0.07
```



# TOP - 2<sup>nd</sup> Row

- The second row gives the following information:
- Row 2 shows the number of process running on server and there state.

Tasks: 146 total, 2 running, 144 sleeping, 0 stopped, 0 zombie



**\*46 total:**      **Total Processes in Active Mode**

**\*2 running:**      **Current running process**      **TecAdmin.net**

**\*114 sleeping:** **Total processes in sleep mode.**

**\*0 stopped:**      **Total processes stopped**

**\*zombie:**      **Total processes in zombie state**



# TOP –Row3

---

- Row three shows the cpu utilization status on server, you can find here how much cpu is free and how much is utilizing by system.

Cpu(s): 1.4%us, 0.2%sy, 0.1%ni, 62.4%id, 35.7%wa, 0.0%hi, 0.0%si, 0.1%st

\*1.4%us: % CPU Used by User Processes

\*0.2%sy: % CPU Used by System Processes

\*0.1%ni: % CPU Processes Used by setting nice value

\*62.4%id: % CPU in idle state

\*35.7%wa: % CPU Waiting for I/O

\*0.0%hi: % CPU Used by Hardware interrupts

\*0.0%si: % CPU Used by Software interrupts

\*0.1%st: Steal time ( Read below definition )

TecAdmin.net



# TOP-Row 4

- Row 4 shows the memory utilization on server, you can find here how much memory is used, the same results you can find using **free** command.

Mem: 1792168ktotal, 1061520kused, 730648k free, 1044k buffers

Total System  
Memory

Current Used  
Memory by  
System

Free  
Memory

Total Memory  
Used by Buffers



# TOP-Row 5

---

- Row 4 shows the swap memory utilization on server, you can find here how much swap is being used, the same results you can find using **free** command.

Swap: 557048k total, 0k used, 557048k free, 67284k cached

Total Swap  
Memory in  
System

Current Used  
Swap Memory by  
System

Free Swap  
Memory

Total Cached  
Memory by  
System



# TOP-Row 6

- In this steps you will see all running process on servers and there additional details about them like below.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5752	root	20	0	176m	25m	12m	S	6	0.4	3:53.01	jsvc

**\*PID(5752):**

**PID of running Process**

**\*USER(root):**

**User under which process is running**

**\*PR(20):**

**Priority Of running Process**

**\*NI(0):**

**Nice Value of running Process**

**\*VIRT(176):**

**Virtual Memory used by Process**

**\*RES(25m):**

**Physical Memory used by Process**

**\*SHR(12m):**

**Shared Memory used by Process**

**\*S(S):**

**Current Status of Running Process**

**\*%CPU(6):**

**% CPU Used by this Processes**

**\*%MEM(0.4):**

**% RAM Used by this Process**

**\*TIME+(3:53:01):**

**Total time of process running for**

**\*COMMAND(jsvc):**

**Name of Process**





# TOP Shortcuts

---

- **Note:** Press below shortcuts at the time of running top command.
- l –To display or to hide load average line
- t –To display or to hide task/cpu line
- 1 –To display or hide all other CPU's
- m –to display or to hide RAM and SWAP details
- s –To change the time interval for updating top results(value is in sec's)
- R –To sort by PID number
- u — Press u then username to get only that user process details
- P –To sort by CPU utilization
- M –To sort by RAM utilization
- c –To display or hide command full path
- r –To renice a process, press r then the PID no then the renice value to renice a process.
- k –To kill a process, press k then PID number then enter to kill a process
- w –To save the modified configuration permanently.
- q –To quit the top command.
- h –for getting help on top command



# ZOMBIE

---

- **Zombie process** or **defunct process** is a process that has completed execution but still has an entry in the process table. This entry is still needed to allow the parent process to read its child's exit status.
- **Zombie process is a process state when the child dies before the parent**
- **Preap**
- **Kill -9**



# Using ps..

---

- `ps -e` : viewing all processes
- `Ps -aux` (to check process)
- Usefull options:
  - a = process by all users
  - x = process from all terminal
  - u = show process owner
  - w = include command arguments
  - f = show process ancestry
- `ps -f` :
  - UID The user ID of the process' owner.
  - PPID The PID of the process' parent process.
  - C The amount of processor time utilized by the process.
  - STIME The time that the process started
  - `ps -l`



# ps -l

---

- S The state of the process. This column uses the following codes:
  - D Uninterruptible sleep.
  - R Running.
  - S Interruptible sleep.
  - T Stopped or traced.
  - Z Zombied.
- PRI The priority of the process.
- NI The nice value of the process. We'll talk about what this means in the next section.
- SZ The size of the process.
- WCHAN The name of the kernel function in which the process is sleeping.
- You will see a dash (–) in this column if the process is currently running.



# Ending a Running Process

---

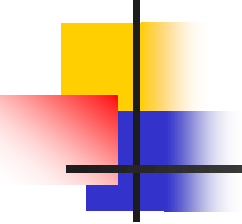
- Kill
  - SIGHUP This is kill signal 1. This signal restarts the process. After a restart, the process will have exactly the same PID that it had before.
  - This is a very useful option for restarting a service for which you've made changes in a configuration file.
  - SIGINT This is kill signal 2. This signal sends a `^C` key sequence to the process.
  - SIGKILL This is kill signal 9. This is a brute-force signal that kills the process. If the process was hung badly, this option will force it to stop. However, the process may not clean up after itself if this signal is used. The resources allocated to the process may remain allocated until the system is restarted.
  - SIGTERM This is kill signal 15. This signal tells the process to terminate immediately. This is the default signal sent by kill if you omit a signal in the command line. This signal allows the process to clean up after itself before exiting.
- Killall
- Exercise 10-1: Working with Linux Processes



# Usage of free command

---

- Total memory
- Used memory
- Free memory
- Shared memory: Indication of how much memory is being shared to 1 process or more.
- Buffers: It's a temporary location to store data for a particular application and this **data is not used** by any other application .
- Cache : It's a memory location to store frequently used data for faster access .this cache used for multiple access.

- 
- 
- #free
  - #free -m
  - #free -g
  - #free -c 5 -s 1 (c=count s=time period)
  - #free -g -c 5 -s 1 (g=gigabyte)
  - #free -l
  - **Low memory:** it's a segment memory that reserved for linux kernel.
  - **High memory:** is simply reserved for user space programs such as application running on a system .

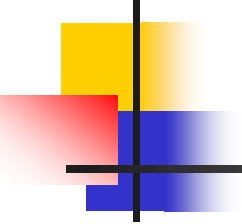


## How to clear the cache memory

---

- `#free -m`
- `sync; echo 3 > /proc/sys/vm/drop_caches`



- 
- 
- To terminate a process, highlight it and press Control-C, select the "Kill" option from the options that are available.



# Clearing Frozen Processes

---

- “kill -signal PID”
- “pkill -signal Process”
- Signal Numbers and Names
  - 1        SIGHUP (Hangup)
  - 2        SIGINT (Interrupt)
  - 9        SIGKILL (Kill)
  - 15       SIGTERM (Terminate)



# Prioritizing Processes

---

- `nice -n nice_levelcommand.`
- `Renice n pid`



# Altering process schedule priority

---

- Nice is the command to change the priority.
- Process are scheduled with default priority of 0.
- Priority value can range from -20(highest priority) to 19 (lowest)
- Syntax
  - `#nice [-n adjustment] command`
  - `#nice top`
  - `#nice -n 15 top`
- Only super user can change the priority.



# Altering process schedule priority

---

- Renice changes the priority of a running process.
- Once a priority value is raised, a non privileged user can't lower it.
- If you want to change the the priority of an entire user group, use the `-g` option.
- To modify the priority of all the process of a particular user use `-u` option.
- `#renice 15 -u username command`
- `#renice -15 -p (pid of user process)`
- `-15` is high priority
- `15` is low priority.