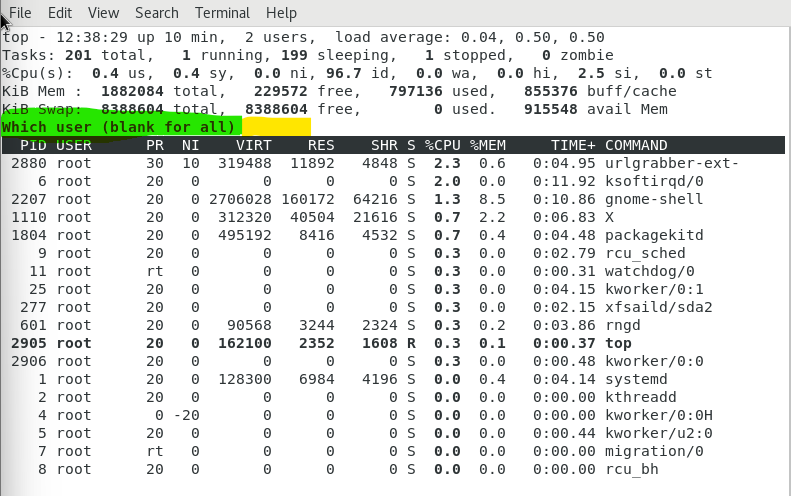
Lecture 09

**Performance Monitoring-TOP-FIND-FSH**

**Process and Daemons**

A process is an instance of a program running on a computer. It consists of the program code, data being processed and system resources such as memory and CPU time.

A daemon is a type of process that runs in the background and performs tasks without user interaction. It is often used to run services or programs that provide system functions, such as the printing spooler or network services.

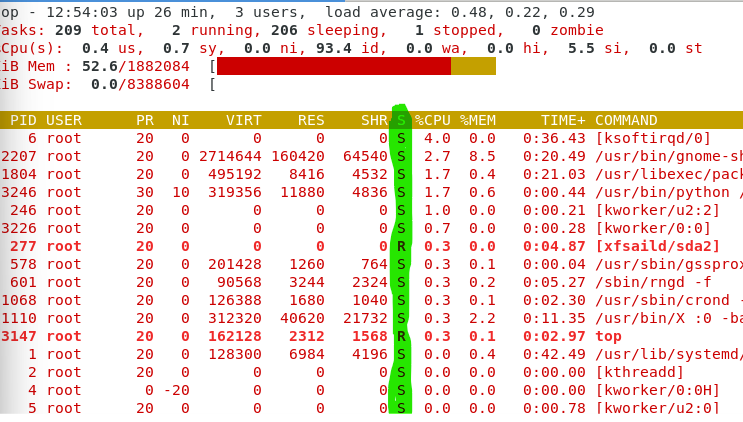
* Top command 🡪 stands for Table of Processes”
* $ sar
* $ ps
* $ vmstat
* $ top -n1 > /opt/top-output.txt 🡪 to take output in a file
* There many flags in “top” according to specific requirement
* To check a particular user related process, press “u” and type user\_name
* 
* Tip:-
* To login from a new user
* $ su - <user\_name>
* Press “c”” to display high usages process, “m” to sort according to memory, “z” to change color to red, “P” to sort according to CPU usage.
* Using multiple flags is called “toggle”

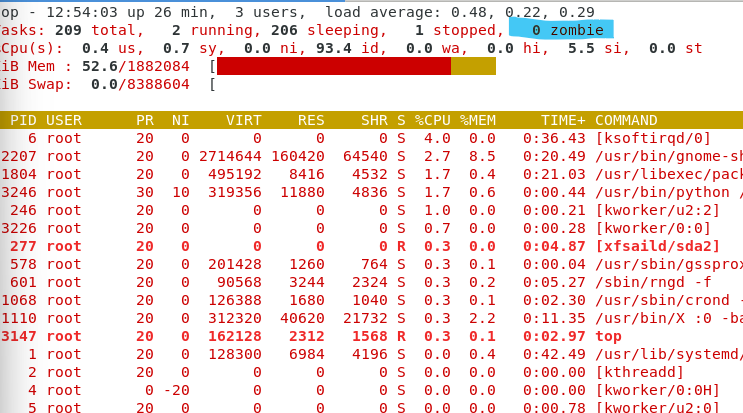
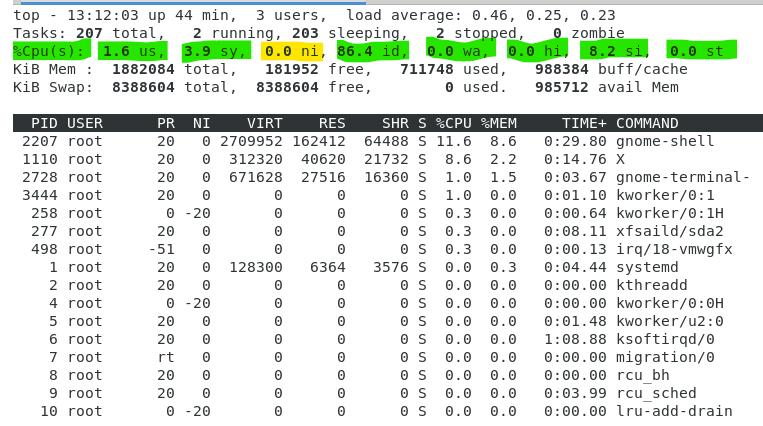
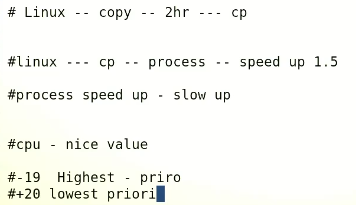
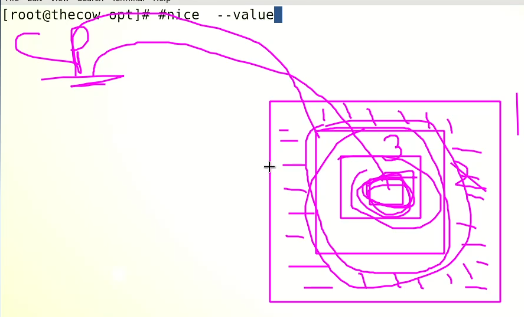
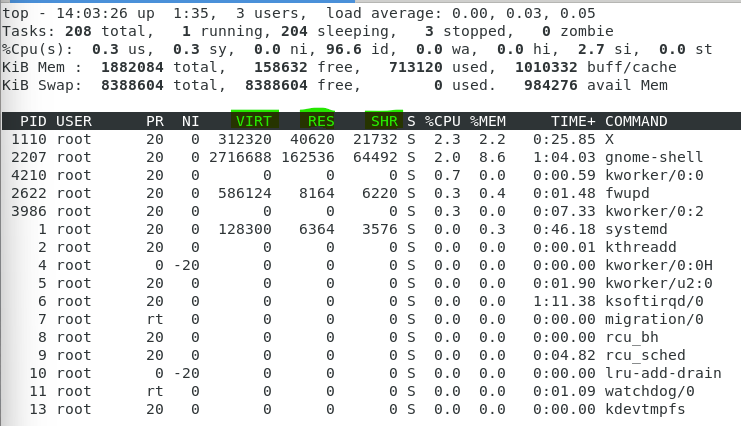
**States of processes of Linux**

In Linux, a process can be in one of several states, including:

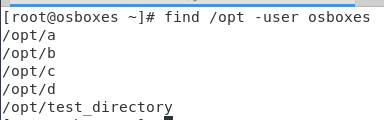
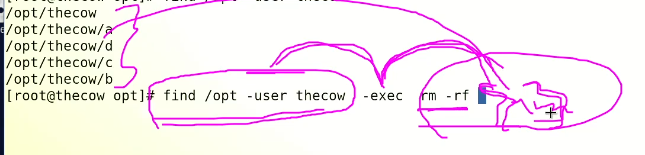
1. Running: The process is currently executing on the CPU.
2. Sleeping: The process is waiting for an event to occur, such as I/O completion or a signal from another process.
3. Stopped: The process has been stopped by a signal from the operating system or a user. It can be restarted by sending a signal to the process.
4. Zombie: The process has completed execution but its resources have not been cleaned up by its parent process.
5. Traced/Debugging: The process is being traced or debugged by another process.
6. Paging: The process is being swapped in or out of physical memory.
7. Dead: The process has terminated and its resources have been cleaned up by the operating system.

**The field of state is highlighted**

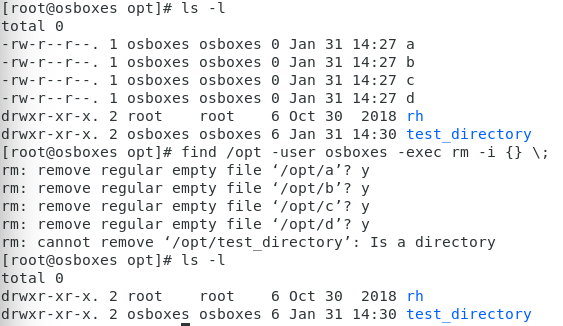
****

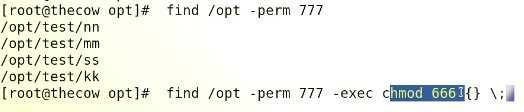
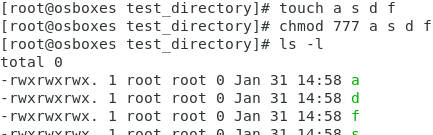
* R 🡪 running
* S 🡪 sleeping (present in RAM but not processing in CPU)
* Zombie or dfunc 🡪 Uninterruptable (dead in RAM but occupy memory in RAM) dead process waiting to be killed by “parent”
* Solution to kill Zombie process 🡪 reboot is a good option after asking for “downtime” if the machine is slow.
* 
* $ ps -ax also displays state of processes.
* 
* In CPU utilization filed,
  + us 🡪 user utilization
  + sy 🡪 system utilization
  + ni 🡪 nice value (important)
  + id 🡪 idle value
  + wa 🡪 wait chain.
  + hi 🡪 hardware interrupt 🡪 suppose network card is creating problem.
  + si 🡪 software interrupt
  + st 🡪 software tracing
* **ni 🡪 nice value**
* In Linux, the "nice" value is used to specify the priority of a process. A process with a higher nice value will have a lower priority and will receive fewer system resources compared to a process with a lower nice value. The range of nice values is from -20 to 19, with -20 being the highest priority and 19 being the lowest. The default nice value is 0. The nice value can be changed using the "nice" or "renice" command.
* How to change the “ni” value in top command,
* 
* Press “r” to put PID of the process that is to be set for priority. 🡪 then set the ni value fro -19 to +20
* Avoid assigning -19 because its for Kernel
* By $ nice command
* 
* The concept of CPU cycles,
* سی پی یو میں سائکلز ہوتے ہیں جیسا کہ تصویر میں دیکھایا گیا ہے،
* سب سے اندر چھوٹا سائیکل ہوتا ہے اور باہر سب سے بڑا سائکل ہوتا ہے۔
* جب ہم نائس ویلیو کمانڈ سے کسی پراسس کی ویلیو تبدیل کرتے ہیں تو وہ پراسس اپنی ویلیو کے حساس سے سب سے چھوٹے سائیکل میں ڈال دیا جاتا ہے اور اس طرح وہ ٹاسک یا پراسس ہائی پریارٹی کی بنیاد پہ جلدی مکمل ہوتا ہے۔
* CPU cycles refer to the basic unit of processing performed by a central processing unit (CPU). One cycle is the time it takes for the CPU to perform a single instruction, such as executing a basic arithmetic operation or accessing a memory location. The number of cycles per second is referred to as the CPU's clock speed, and is typically measured in megahertz (MHz) or gigahertz (GHz). A CPU with a higher clock speed is capable of performing more cycles per second and can therefore handle more demanding tasks. However, clock speed is not the only factor that affects a CPU's performance, as other factors such as the number of cores and the architecture of the CPU also play important roles.
* 
* PR 🡪 Priority
* NI 🡪 Nice Value
* VIRT 🡪 virtual memory
* RES 🡪 resident memory 🡪 memory of the process (self-memory)
* SHR 🡪 shared memory.
* S 🡪 state of process
  + Zombie process 🡪 because of wrongly designed applications
* %CPU 🡪 CPU utilization
* %MEM 🡪 memory utilization
* TIME 🡪 process running time.
* COMMAND 🡪 basically name of the process.
* <https://phoenixnap.com/kb/top-command-in-linux#:~:text=The%20top%20(table%20of%20processes,the%20top%20command%20in%20Linux>.

“fine” command

* Very powerful command
* $ find /<path> -user <user\_name> 🡪 To fine information related to a particular user
* 
* To search a particular file
* $ find / -name “\*.txt”
* To execute tasks 🡪 multiple commands combination
* $ find /opt -user osboxes -exec -rf rm {} \;🡪 to remove the files of a specific user.
* 

To remove all files at once related to a specific user

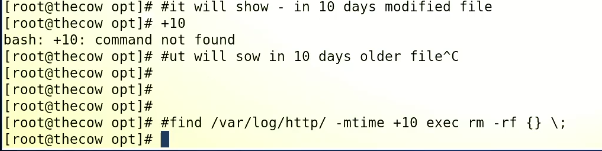


* To find particular permission with “find” command.
* $ find / opt -perm 777 🡪 all files with full permission in opt directory.
* 
* Files created and given full permissions,
* 
* Now files searched with find command that have “777” or full permission.
* Text

  Description automatically generated
* Now permissions changed with find command for all files
* 
* Text

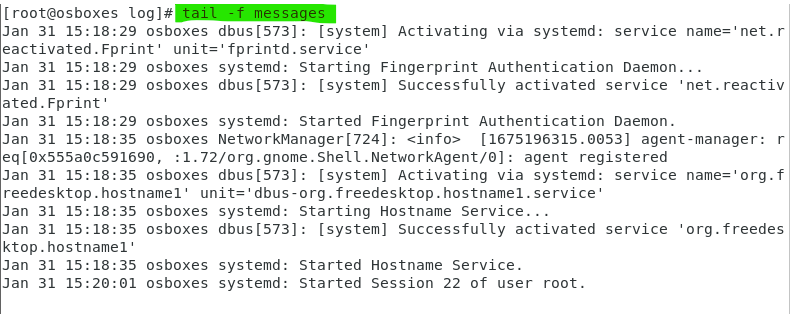
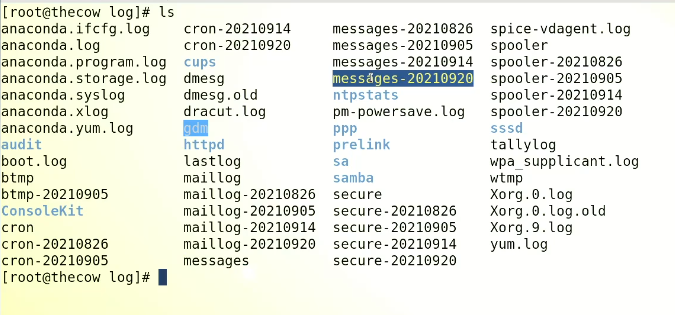
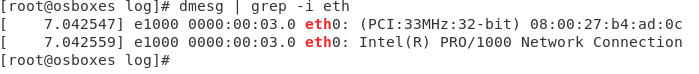
  Description automatically generated

Here is a list of some commonly used flags with the "find" command:

* **-name**: Search for files with a specific pattern in the name.
* **-type**: Specify the type of file to search for (e.g. f for files, d for directories).
* **-iname**: Same as **-name**, but case-insensitive.
* **-mtime**: Search for files modified within a certain number of days (e.g. **-mtime -7** for files modified in the last 7 days).
* **-size**: Search for files with a specific size (e.g. **-size +100M** for files larger than 100 MB).
* **-exec**: Execute a command for each file found by **find**.
* **-printf**: Print information about each file found by **find**.
* **-prune**: Skip the current directory and its subdirectories.
* **-ok**: Same as **-exec**, but prompts for confirmation before executing the command.
* To find files created 10 days before,
* $ find / -mtime -10
* 
* <https://linuxhint.com/use-the-find-command-in-linux-to-search-files/#:~:text=Syntax%20of%20%E2%80%9Cfind%E2%80%9D%20Command%20in%20Linux&text=Three%20attributes%20go%20with%20the,to%20perform%20with%20the%20file>.
* <https://www.tecmint.com/35-practical-examples-of-linux-find-command/>

**File system hierarchy**

****

* /var 🡪 contains system logs
* The /var is the directory in Linux which varies in size that’s why it is called var or variable.
* **Size kept on increasing**
* **$ cad /var/log 🡪 log files in this directory**
* A professional Admin always reads logs
* Log is daily activity report of the system 🡪 event viewer in Windows
* MESSAGES FILE carries common log files
* **head -20 🡪 starting 20 rows**
* **tail -20 last 20 line**
* **- f real time logs**
* ****
* **$ tailf messages 🡪 same result**
* ****
* **File rotate term means backup** is **generated**
* $ dmesg 🡪 diagnostic messages 🡪 OS detected hardware logs 🡪 like device manager in windows
* To find “ethernet drver”
* $ dmesg | grep -I eth
* 
* Tip:- all detected hardware by OS is in “dmesg” 🡪 but not motherboard
* $ dmidecode 🡪 all hardware information
* $ biosdecode 🡪 BIOS information