Lecture 19

**19.grep-find-locate-fstab-process-daemons**

**grep**

* $ ip r l 🡪 to show IP of the current machine.
* Production servers run main services 🡪 live IP can be accessed from out side.
* Nonproduction servers
* Development Server Dev Server 🡪 programming (developers)
* UAT 🡪 for testing
* Proof of Concept 🡪 foe demo purpose
* <https://www.gratasoftware.com/what-is-each-server-for-development-test-uat-or-staging-demo-and-production/>

UAT, Dev, POC, and Prod servers are different types of servers that are used in software development and deployment.

* UAT (User Acceptance Testing) servers are used to test software applications and services before they are deployed to production. These servers are used to ensure that the software functions correctly and meets the requirements of the end users.
* Dev (Development) servers are used by developers to create and test new software applications and services. These servers are typically not connected to the internet and are used for development and testing purposes only.
* POC (Proof of Concept) servers are used to demonstrate the feasibility of a new software application or service. These servers are used to test new technologies and features before they are deployed to production.
* Prod (Production) servers are used to host live, operational software applications and services. These servers are typically connected to the internet and are accessible to end users. Production servers are used to handle the day-to-day operations of an organization, such as serving web pages, processing transactions, and managing databases.

It is important to note that depending on the organization, the terms and the number of servers may vary.

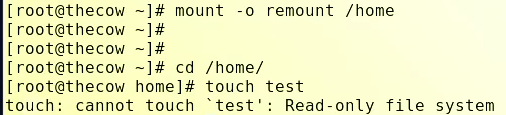
* **Task**
* **Bring to Ctrl + D (maintenance mode)**
* By wrong entries in “fstab”
  + Don’t delete anything (partition will shift in this case)
  + Sometimes the 5th and 6th fields in fstab should be “0” to bring maintenance mode
  + Because it skipps these partitions if there is “0” in these fields during OS loading process.”
* Permissions (4th field in fstab)
* The fourth field in the /etc/fstab file is used to specify the file system mount options. To edit the permissions for a file system in the /etc/fstab file, you can add the **defaults** or **rw** option to the fourth field.
* The **defaults** option sets the file system to default options, which includes read-write access. The **rw** option sets the file system to read-write access.
* Here is an example of editing the /etc/fstab file to set the permissions of a file system to read-write:
* # /etc/fstab /dev/sda1 /mnt/data ext4 defaults,rw 0 0
* It's important to note that editing the /etc/fstab file requires root privileges, you should take a backup of the file before editing, and use the command **mount -a** after editing the file to apply the changes. Also, it's a good idea to test the changes in a non-production environment before applying it in production.
* Types of permissions

The /etc/fstab file uses several options to specify file system mount options, including permissions. Here are some common options that can be used to set permissions in the /etc/fstab file:

* **defaults**: This option sets the file system to default options, which includes read-write access.
* **rw**: This option sets the file system to read-write access.
* **ro**: This option sets the file system to read-only access.
* **user**: This option allows non-root users to mount and unmount the file system.
* **noexec**: This option prevents the execution of binary files on the file system.
* **nosuid**: This option prevents set-user-ID and set-group-ID bits from taking effect.
* **nodev**: This option prevents the use of special device files on the file system.
* **nofail** : This option will not cause the file system to be marked as failed.

It's important to note that the options may vary based on the file system type, and the options that you can use with one file system may not be applicable to another file system type. You should always check the documentation of the filesystem you are using and consult the man page of mount command to have all possible options. It's also important to note that some of these options may have security implications, so you should use them with caution.



* Use “,” otherwise it will consider is as next field i.e 5th field
* “noauto” no mount if system reboots
* “noexec” no script can run or programs
* “ro” read only mode
* “auto” reverse of “”
* “nouser” only root can mount
* After changes run this command
* **$ mount -o remount /<mount\_point>**
* It should be noted that $ mount -a will no do the job this time and it will not reflect the changes made in fstab, **it only mounts unmounted partitions**
* Default field contains following changes,
  + auto
  + rw
  + ro
  + async
  + suid
  + exec
  + nouser
* editing a permission after “default” will take effect,
* 
* 

**Process and Daemons**

A **process** is an instance of a program that is being executed by a computer's operating system. It contains the program code and its current activity.

A **daemon** is a type of process that runs in the background, typically without any user interaction. Daemons are often used to perform tasks that need to be done constantly in the background, such as monitoring for new incoming connections on a network or scheduling tasks to be performed at a later time.

* Everything in RAM is process
* Everything running is called process
* httpd 🡪 apach
* mysql 🡪 database
* docker 🡪 container service
* kubernates 🡪 cluster orchestration service
* **open files 🡪 program or files loaded in RAM are called “open files”**
* to know openfiles
* **$ lsof**
* **$ lsof | wc -l** 🡪 optional command to list no. of open files and programs.
* 
* These files & programs are loaded in RAM
* $ ls -l | wc -c 🡪 checks characters

**grep**

* $ grep 🡪 to search “string wise”
* $ grep <any\_string> /etc/passwd 🡪 displays “<any\_string>” string from “passwd” file
* 
* Stands for: **Global regular expression print**
* **To print in color** add - -color flag
* 
* $ grep -A1 osboxex /etc/passwd 🡪 **includes 1 line below and the target string**
* **A2 for two line**
* **B2 for 1 line above and 1 line below**
* **$ ls /etc/\* | grep -I host 🡪** host file grep
* **$ ls /etc/\* | grep -I host | wc -l 🡪 also counts lines**

**Find**

* In Linux to search specif files of directories/folders,
* There are
* Two commands.
* “find” 🡪 very powerful command. Can delete large no of files, can change permissions of large no of files etc.
* Find command works slow because it goes through each directory to find the desired result
* 15 uses of “find” commands,

**Find all files with the name "file.txt" in the current directory and its subdirectories:**

1. **find . -name "file.txt"**

Find all files with the extension .txt in the current directory and its subdirectories:

1. **find . -name "\*.txt"**

Find all files modified within the last 24 hours in the current directory and its subdirectories:

1. **find . -mtime 0**

Find all files larger than 100MB in the current directory and its subdirectories:

1. **find . -size +100M**

Find all files that are owned by a specific user:

1. **find . -user username**

Find all files that are owned by a specific group:

1. **find . -group groupname**

Find all files with the name "file.txt" and delete them:

1. **find . -name "file.txt" -delete**

Find all files with the extension .txt and print the path of the file and its last modification time:

1. **find . -name "\*.txt" -printf "%p %TY-%Tm-%Td %TT\n"**

Find all files with the name "file.txt" and execute a command on each file:

1. **find . -name "file.txt" -exec cat {} \;**

Find all files with the extension .txt and rename them to .bak:

1. **find . -name "\*.txt" -exec rename .txt .bak {} \;**

Find all files with the name "file.txt" and move them to a specific directory:

1. **find . -name "file.txt" -exec mv {} /path/to/directory/ \;**

Find all files that are empty and delete them:

1. **find . -empty -delete**

Find all files with the extension .txt that are larger than 100MB and print their path:

1. **find . -name "\*.txt" -size +100M -print**

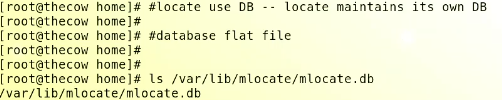
Find all files that are owned by the user 'john' and group 'admin' and change their ownership to 'root':

1. **find . -user john -group admin -exec chown root {} \;**

Find all files that are older than 3 months and compress them:

1. **find . -mtime +90 -exec gzip {} \;**

**Note:** These are some of the examples of find command in Linux, the options and usage of find command are very vast and versatile, you can try different options and use it as per your requirement.

* $ locate 🡪 faster then “find”
* 
* Newly created files are not updates in it DB
* The DB updates after reboot