Assessment -4

Advanced Linux

Trainee Name : Aditya Upadhyay
Newers ID : 3978
Mentor Name : Tarun Saxena
College : UPES

1. What is the size of MBR and what does it contains.

512 bytes
The **Master Boot Record** (**MBR**) is the information in the first **sector** of any hard disk or diskette that identifies how and where an operating system is located so that it can be **boot** (loaded) into the computer's main storage or random access memory.

2. In which file you can write commands which you want to run whenever Linux system starts/restarts?

i.

Put the command in your crontab file. The crontab file in Linux is a daemon that performs user-edited tasks at specific times and events. To edit the file, open a terminal and type "sudo crontab -e" to open your crontab file in the default text editor. At the first available line, type "@reboot xxxx", where "xxxx" is the command you wish to run. Save the file and exit.

ii.

Put a script containing the command in your /etc directory. Create a script such as "startup.sh" using your favorite text editor. Save the file in your /etc/init.d/ directory. Change the permissions of the script (to make it executable) by typing "chmod +x /etc/init.d/mystartup.sh".

iii.

Edit the /rc.local script using your text editor. On Fedora systems, this script is located in /etc/rc.d/rc.local, and in Ubuntu, it is located in /etc/rc.local. Once you add the commands you wish to run -- making sure you do so as root -- save the file and exit. The commands will run after the next startup.

3. Reboot the system using runlevel.

4. Restart cron service.

```
aditya@aditya-0:$ systemctl status cron
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-02-11 17:42:26 IST; 32s ago
     Docs: man:cron(8)
 Main PID: 28782 (cron)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/cron.service
           └─28782 /usr/sbin/cron -f

Feb 11 17:42:26 aditya systemd[1]: Started Regular background program processing daemon.
Feb 11 17:42:26 aditya cron[28782]: (CRON) INFO (pidfile fd = 3)
Feb 11 17:42:26 aditya cron[28782]: (CRON) INFO (Skipping @reboot jobs -- not system startup)
aditya@aditya-0:$ systemctl restart cron
aditya@aditya-0:$
```

5. Create an ext4 filesystem

=> $mkfs –t ext4 /dev/xvda

6. Mount the created filesystem on /partition directory.

=> $ mkdir /partition
$ mount /dev/xvda /partition

7. Difference between LVM and RAID.

=> A RAID device is a physical grouping of disk devices in order to create a logical presentation of one device to an Operating System for redundancy or performance or a combination of the two whereas, LVM is a logical layer that can be manipulated in order to create and, or expand a logical presentation of a disk device to an Operating System. LVM can be used to create partition whose size can be changed even without restarting.

8. Create a LVM(Slide 13)

=> -Select Physical Storage for LVM
$pvcreate /dev/sda1 /dev/sda2
-Create the Volume Group
$vgcreate vol_grp1 /dev/sda1 /dev/sda2
-Create Logical Volumes
$lvcreate -l 20 -n logical_vol1 vol_grp1

9. Create a RAID1 device(Slide 19)

=> **Installation**: apt-get install mdadm rsync initramfs-tools

Create partitions : using fdisk on say /dev/sdb and /dev/sdc

Verify the changes : mdadm -E /dev/sd[b-c]

Create RAID1 Device

mdadm --create /dev/md0 --level=mirror --raid-devices=2 /dev/sd[b-c]1

10. Create a swapfile of 500Mb(slide20)

=>creates a file of a preallocated size instantly, without actually having to write dummy contents

$ fallocate -l 500M /swapfile

$ mkswap /swapfile

$swapon /swapfile

11. Set setuid and setgid on two different files.

```
aditya@aditya-0:$ chmod u+s abc
aditya@aditya-0:$ ll
total 8
drwxr-xr-x  2 aditya aditya 4096 Feb 13 15:10 ./
drwxr-xr-x 20 aditya aditya 4096 Feb 13 15:10 ../
-rwSr--r--  1 aditya aditya    0 Feb 13 15:10 abc
-rw-r--r--  1 aditya aditya    0 Feb 13 15:10 def
aditya@aditya-0:$ chmod u+x abc
aditya@aditya-0:$ ll
total 8
drwxr-xr-x  2 aditya aditya 4096 Feb 13 15:10 ./
drwxr-xr-x 20 aditya aditya 4096 Feb 13 15:10 ../
-rwsr--r--  1 aditya aditya    0 Feb 13 15:10 abc*
-rw-r--r--  1 aditya aditya    0 Feb 13 15:10 def
aditya@aditya-0:$
```

```
aditya@aditya-0:$ chmod g+x def
aditya@aditya-0:$ chmod g+s def
aditya@aditya-0:$ ll
total 8
drwxr-xr-x  2 aditya aditya 4096 Feb 13 15:10 ./
drwxr-xr-x 20 aditya aditya 4096 Feb 13 15:10 ../
-rwsr--r--  1 aditya aditya    0 Feb 13 15:10 abc*
-rwxr-sr--  1 aditya aditya    0 Feb 13 15:10 def*
aditya@aditya-0:$
```

12. What is the use of Sticky bit?

=> A Sticky bit is a permission bit that is set on a file or a directory that lets only the owner of the file/directory or the root user delete or rename the file.

13. Create a user and add it to one secondary group.

```
aditya@aditya-0:$ sudo adduser test
[sudo] password for aditya:
Adding user `test' ...
Adding new group `test' (1001) ...
Adding new user `test' (1001) with group `test' ...
Creating home directory `/home/test' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for test
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
aditya@aditya-0:$ id test
uid=1001(test) gid=1001(test) groups=1001(test)
aditya@aditya-0:$ usermod -G aditya test
usermod: Permission denied.
usermod: cannot lock /etc/passwd; try again later.
aditya@aditya-1:$ sudo !!
sudo usermod -G aditya test
aditya@aditya-0:$ id test
uid=1001(test) gid=1001(test) groups=1001(test),1000(aditya)
aditya@aditya-0:$
```

14. Lock this user.

```
aditya@aditya-0:$ sudo passwd -l test
passwd: password expiry information changed.
aditya@aditya-0:$ su test
Password:
su: Authentication failure
aditya@aditya-1:$
```

15. Give this user full access (without password).

```
  GNU nano 2.9.3                                                /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
test    ALL=(ALL) NOPASSWD:ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

```
test@aditya:/home/aditya/testing$ sudo apt install cowsay
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  filters cowsay-off
The following NEW packages will be installed:
  cowsay
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 17.7 kB of archives.
After this operation, 89.1 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 cowsay all 3.03+dfsg2-4 [17.7 kB]
Fetched 17.7 kB in 0s (55.5 kB/s)
Selecting previously unselected package cowsay.
(Reading database ... 170227 files and directories currently installed.)
Preparing to unpack .../cowsay_3.03+dfsg2-4_all.deb ...
Unpacking cowsay (3.03+dfsg2-4) ...
Setting up cowsay (3.03+dfsg2-4) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
test@aditya:/home/aditya/testing$
```

16. Delete the create user after taking backup of it home directory.

```
aditya@aditya-0:$ sudo deluser --remove-home --backup-to /tmp/ test
Looking for files to backup/remove ...
Backing up files to be removed to /tmp/ ...
backup_name = /tmp//test.tar
/bin/tar: Removing leading `/' from member names
Removing files ...
Removing user `test' ...
Warning: group `test' has no more members.
Done.
aditya@aditya-0:$
```

17. Create a file with some content. Change all lower case letters to upper case letters and save output to another file using redirections.

```
aditya@aditya-0:$ echo "aditya" >> testfile
aditya@aditya-0:$ tr '[a-z]' '[A-Z]' < testfile > output
aditya@aditya-0:$ cat output
ADITYA
aditya@aditya-0:$
```

18. Set nice value of a process to -1.

```
aditya@aditya-0:$ sudo renice -1 14555
14555 (process ID) old priority 0, new priority -1
```

19. Get a list of all files used by “telnet”.

```
aditya@aditya-0:$ dpkg-query --listfiles telnet
/.
/usr
/usr/bin
/usr/bin/telnet.netkit
/usr/share
/usr/share/doc
/usr/share/doc/telnet
/usr/share/doc/telnet/BUGS
/usr/share/doc/telnet/README.gz
/usr/share/doc/telnet/README.telnet
/usr/share/doc/telnet/README.telnet.old.gz
/usr/share/doc/telnet/changelog.Debian.gz
/usr/share/doc/telnet/copyright
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/telnet
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/telnet.netkit.1.gz
/usr/share/menu
/usr/share/menu/telnet
aditya@aditya-0:$
```

20. Check if port 22 is listening using netstat and telnet command.

```
aditya@aditya-0:$ telnet localhost 22
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
^C^]
telnet> close
Connection closed.
aditya@aditya-0:$ sudo netstat -tulnp | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      22498/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      22498/sshd
aditya@aditya-0:$
```

21. Create a cron job which runs once a week at 23:45.

=> $crontab -e
45 23 * * 1

22. Difference between dig and traceroute

Traceroute gives hop count.