OS Lab06 Solution

Task1 :

1. What do you mean by a daemon, give command to display daemons running on your system

A *daemon* (pronounced "dee-mon") is a background process that runs on a Unix-like system, typically waiting to perform a specific task or service. Daemons often start at boot and run continuously without user interaction, handling tasks such as system logging, network connections, scheduled jobs, and various services. They usually have names that end with a "d," like **`sshd`** for the SSH service or **`httpd`** for web servers.

Commands:

To display running daemons on your system, we can use the **`ps`** command along with **`grep`** to filter for common daemon processes, or we can list all services and identify those currently active:

```
habiba@DESKTOP-MC3IIQB:~$ sudo systemctl list-units --type=service --state=running
[sudo] password for habiba:
  UNIT                          LOAD   ACTIVE SUB     DESCRIPTION
  console-getty.service         loaded active running Console Getty
  containerd.service            loaded active running containerd container runtime
  cron.service                  loaded active running Regular background program processing daemon
  dbus.service                  loaded active running D-Bus System Message Bus
  docker.service                loaded active running Docker Application Container Engine
  getty@tty1.service            loaded active running Getty on tty1
  rsyslog.service               loaded active running System Logging Service
  snapd.service                 loaded active running Snap Daemon
  systemd-journald.service      loaded active running Journal Service
  systemd-logind.service        loaded active running User Login Management
  systemd-resolved.service      loaded active running Network Name Resolution
  systemd-timedated.service     loaded active running Time & Date Service
  systemd-timesyncd.service     loaded active running Network Time Synchronization
  systemd-udevd.service         loaded active running Rule-based Manager for Device Events and Files
  unattended-upgrades.service loaded active running Unattended Upgrades Shutdown
  user@1000.service             loaded active running User Manager for UID 1000
  wsl-pro.service               loaded active running Bridge to Ubuntu Pro agent on Windows

Legend: LOAD   → Reflects whether the unit definition was properly loaded.
        ACTIVE → The high-level unit activation state, i.e. generalization of SUB.
        SUB    → The low-level unit activation state, values depend on unit type.

17 loaded units listed.
habiba@DESKTOP-MC3IIQB:~$ ps -ef | grep -E 'd$'
root          51       1  0 22:42 ?        00:00:00 /usr/lib/systemd/systemd-journald
root          93       1  0 22:42 ?        00:00:00 /usr/lib/systemd/systemd-udevd
systemd+     106       1  0 22:42 ?        00:00:00 /usr/lib/systemd/systemd-resolved
systemd+     107       1  0 22:42 ?        00:00:00 /usr/lib/systemd/systemd-timesyncd
```

2. Difference between at and batch command

at command:

- Used to schedule a job to run at a specific time and date in the future.
- Syntax: **`at [time]`**, where you specify the exact time (e.g., **`at 3pm`** or **`at 02:00`** for 2 AM).
- Jobs are executed only once at the scheduled time.
- Useful for tasks that need to run at a precise moment, like reminders or automated tasks at a set time.

`batch` Command:

- Schedules jobs to run when system load is low or below a specified threshold.
- Executes jobs sequentially based on system load rather than a specific time.
- Useful for tasks that are resource-intensive or don't need to run immediately (e.g., backups or heavy calculations).

3. Difference between cron and anacron command

`cron` Command:

- Used to schedule jobs to run at fixed intervals (e.g., daily, weekly, hourly).
- Requires the system to be running at the specified times (e.g., a job scheduled for midnight won't run if the system is off then).
- Ideal for servers or systems that run continuously.

`anacron` Command:

- Also schedules recurring jobs but is designed for systems that may not run continuously (e.g., laptops, desktops).
- Runs jobs as soon as possible if a scheduled job was missed due to the system being off.
- Suitable for tasks that don't need precise timing but must run periodically, regardless of exact timing.

4. Write a cron entry that will execute a command on 1st of every month at 3:45 pm

To execute a command on the 1st of every month at 3:45 pm, the cron entry should look like this:

45 15 1 * * <command>

Here's a breakdown of this cron entry:

- 45 - minute (45th minute)
- **15** - hour (3 pm in 24-hour format)
- **1** - day of the month (1st)
- ***** - any month

- `*` - any day of the week

Replace **`<command>`** with the actual command we want to execute.

Task2:

1. What is the difference between a binary package and a source package?

Binary Package:

- A **binary package** contains precompiled, ready-to-install files that can be directly executed by the system.
- It includes the compiled binary executable, libraries, and other required files.
- Binary packages are typically faster to install since they don't require compilation on the user's machine.
- They are platform-specific (e.g., `.deb` for Debian-based systems, `.rpm` for Red Hat-based systems).

Source Package:

- A **source package** contains the program's source code, which must be compiled on the user's machine before it can be installed and run.
- It gives users more control, allowing customization options at compile-time (such as specifying configuration flags).
- Installation from a source package takes longer since the code has to be compiled, which also requires development tools (like `gcc` or `make`).
- Source packages are more platform-independent and are commonly distributed as **`.tar.gz`** or **`.tar.bz2`** archives.

2. What is the role of a package manager, what all package managers are available to you for installing Debian packages?

A **package manager** automates the process of installing, updating, and removing software on a system. It manages dependencies, ensures software compatibility, handles version control, and secures packages through verified repositories, simplifying system administration.

Package Managers for Debian Packages:

1. **APT (Advanced Package Tool)**: The primary package manager on Debian systems, it automatically handles dependencies. Commands: **`apt install <package>`**, **`apt update`**, etc.
2. **DPKG (Debian Package)**: A low-level tool to install and manage `.deb` files without handling dependencies automatically. Useful for local installations. Command: **`dpkg -i <package>.deb`**.
3. **APT-GET**: The traditional CLI tool for installing and managing packages, particularly useful for scripting. Command: **`apt-get install <package>`**.
4. **Synaptic**: A GUI front-end for APT, ideal for users who prefer graphical interfaces to manage software.
5. **Aptitude**: An advanced tool with both CLI and text-based GUI, offering additional features like package recommendations. Command: **`aptitude install <package>`**.

3. Install binary package cmatrix on your machine using apt-get command.

sudo apt-get update

sudo apt-get install cmatrix

4. Download source package hello-2.10.tar.gz using apt-get as well as using wget command. Install it on your machine, see its manual page, use it and finally uninstall it. Note down all your observations.

Method 1: Using apt-get source

sudo apt-get update

sudo apt-get install dpkg-dev

apt-get source hello

Method 2: Using wget

wget http://ftp.gnu.org/gnu/hello/hello-2.10.tar.gz

Uninstall:

rm -rf hello-2.10 hello-2.10.tar.gz

Task3:

1. What is the difference between su and sudo command

su (Substitute User): Switches to another user account, typically the root user, for a full session. It requires the target user's password.

- **Usage: su - <username>**

sudo (Superuser Do): Executes a single command with elevated privileges without switching users. It requires the current user's password and is more secure for limited privilege escalation.

- **Usage: sudo <command>**

2. What is the difference between switching user using su command and su – command

su <username>: Switches to the specified user but maintains the current shell environment, meaning paths, variables, and settings from the original user remain.

su - <username>: Switches to the specified user and starts a **new login shell**, fully loading the target user's environment (paths, variables, etc.), as if they had logged in directly.

3. Login as root, and create a new user named kakamanna

su -

useradd kakamanna

passwd kakamanna

4. View the contents of /etc/passwd, /etc/shadow, and /etc/group and try understanding the new entries in those files

1. /etc/passwd

The **/etc/passwd** file contains basic information about user accounts.

2. /etc/shadow

The **/etc/shadow** file stores secure user password information.

3. /etc/group

The **/etc/group** file defines the groups to which users belong

To view the contents of these files, we can use the following commands:

cat /etc/passwd

cat /etc/shadow

cat /etc/group

5. Try logging in as kakamanna, what happened.

```
habiba@DESKTOP-MC3IIQB:~$ sudo su -
root@DESKTOP-MC3IIQB:~# su - kakamanna
$
```

6. As root, assign password to kakamanna and try again logging in as kakamanna. View the contents of /etc/shadow file, what difference have you noticed

```
habiba@DESKTOP-MC3IIQB:~$ sudo su -
root@DESKTOP-MC3IIQB:~# passwd kakamanna
New password:
Retype new password:
passwd: password updated successfully
root@DESKTOP-MC3IIQB:~# su - kakamanna
$ exit
root@DESKTOP-MC3IIQB:~# cat /etc/shadow
root:*:19993:0:99999:7:::
daemon:*:19993:0:99999:7:::
bin:*:19993:0:99999:7:::
sys:*:19993:0:99999:7:::
sync:*:19993:0:99999:7:::
games:*:19993:0:99999:7:::
man:*:19993:0:99999:7:::
lp:*:19993:0:99999:7:::
mail:*:19993:0:99999:7:::
news:*:19993:0:99999:7:::
uucp:*:19993:0:99999:7:::
proxy:*:19993:0:99999:7:::
www-data:*:19993:0:99999:7:::
backup:*:19993:0:99999:7:::
list:*:19993:0:99999:7:::
irc:*:19993:0:99999:7:::
_apt:*:19993:0:99999:7:::
nobody:*:19993:0:99999:7:::
systemd-network:!*:19993:::::
systemd-timesync:!*:19993:::::
dhcpcd:!:19993:::::
messagebus:!:19993:::::
syslog:!:19993:::::
```

7. Login as kakamanna, and see the contents of his home directory. What all hidden files you see, from where they come from, what are their contents, for what all purpose they are used.

```
habiba@DESKTOP-MC3IIQB:~$ su - kakamanna
Password:
$ ls -la ~
total 12
drwxr-xr-x 3 kakamanna kakamanna 4096 Oct 27 00:39 .
drwxr-xr-x 4 root      root      4096 Oct 27 00:38 ..
drwxr-xr-x 2 kakamanna kakamanna 4096 Oct 27 00:39 .landscape
-rw-rw-r-- 1 kakamanna kakamanna    0 Oct 27 00:39 .motd_shown
$ cat .bashrc
cat: .bashrc: No such file or directory
$ ls -la ~/.landscape
total 8
drwxr-xr-x 2 kakamanna kakamanna 4096 Oct 27 00:39 .
drwxr-xr-x 3 kakamanna kakamanna 4096 Oct 27 00:39 ..
-rw-r--r-- 1 kakamanna kakamanna    0 Oct 27 00:39 sysinfo.log
$
```

Task04:

1. Change his personal information of kakamanna using chfn command. Do it as root and then do it as kakamanna and note the difference. What all files have been changed

```
habiba@DESKTOP-MC3IIQB:~$ sudo su -
root@DESKTOP-MC3IIQB:~# chfn kakamanna
Changing the user information for kakamanna
Enter the new value, or press ENTER for the default
        Full Name []: Kaka Manna
        Room Number []:  101
        Work Phone []: 1234567890
        Home Phone []: 0987654321
        Other []:
root@DESKTOP-MC3IIQB:~# grep kakamanna /etc/passwd
kakamanna:x:1001:1001:Kaka Manna,101,1234567890,0987654321:/home/kakamanna:/bin/sh
```

2. Login as root and lock kakamanna. Try logging in as kakamanna, what happened. View the contents of /etc/passwd file, what difference you observed.

```
root@DESKTOP-MC3IIQB:~# passwd -l kakamanna
passwd: password changed.
root@DESKTOP-MC3IIQB:~# cat /etc/shadow | grep kakamanna
kakamanna:!$y$j9T$shvTub.WfTcVuk10RO.aM.$3rkU25pf.Ae946ntOMGBdwwBfxBxKPNpSZ6Tr1qFuA3:20022:0:99999:7:::
root@DESKTOP-MC3IIQB:~#
```

3. Login as root and unlock kakamanna. Repeat above procedure, and note your observations

```
root@DESKTOP-MC3IIQB:~# passwd -u kakamanna
passwd: password changed.
root@DESKTOP-MC3IIQB:~# cat /etc/shadow | grep kakamanna
kakamanna:$y$j9T$shvTub.WfTcVuk10RO.aM.$3rkU25pf.Ae946ntOMGBdwwBfxBxKPNpSZ6Tr1qFuA3:20022:0:99999:7:::
root@DESKTOP-MC3IIQB:~#
```

Task05:

4. Login as root, and delete user kakamanna. See the contents of files /etc/passwd, /etc/shadow, and /etc/group. Note your observations. Also see if the home directory of the user is deleted or not?

```
root@DESKTOP-MC3IIQB:~# userdel kakamanna
root@DESKTOP-MC3IIQB:~# cat /etc/passwd | grep kakamanna
root@DESKTOP-MC3IIQB:~# cat /etc/shadow | grep kakamanna
root@DESKTOP-MC3IIQB:~# cat /etc/group | grep kakamanna
root@DESKTOP-MC3IIQB:~# ls -ld /home/kakamanna
drwxr-xr-x 3 1001 1001 4096 Oct 27 00:39 /home/kakamanna
root@DESKTOP-MC3IIQB:~# rm -rf /home/kakamanna
root@DESKTOP-MC3IIQB:~#
```

5. View the contents of the file /etc/default/useradd, and try to understand its impact on user creation and his password expiration.

```
root@DESKTOP-MC3IIQB:~# cat /etc/default/useradd
# Default values for useradd(8)
#
# The SHELL variable specifies the default login shell on your
# system.
# Similar to DSHELL in adduser. However, we use "sh" here because
# useradd is a low level utility and should be as general
# as possible
SHELL=/bin/sh
#
# The default group for users
# 100=users on Debian systems
# Same as USERS_GID in adduser
# This argument is used when the -n flag is specified.
# The default behavior (when -n and -g are not specified) is to create a
# primary user group with the same name as the user being added to the
# system.
# GROUP=100
#
# The default home directory. Same as DHOME for adduser
# HOME=/home
#
# The number of days after a password expires until the account
# is permanently disabled
# INACTIVE=-1
#
# The default expire date
# EXPIRE=
#
# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
```

6. View the contents of the file /etc/login.defs, and make a note of its contents and its usage.

```
root@DESKTOP-MC3IIQB:~# cat /etc/login.defs
#
# /etc/login.defs - Configuration control definitions for the login package.
#
# Three items must be defined:  MAIL_DIR, ENV_SUPATH, and ENV_PATH.
# If unspecified, some arbitrary (and possibly incorrect) value will
# be assumed.  All other items are optional - if not specified then
# the described action or option will be inhibited.
#
# Comment lines (lines beginning with "#") and blank lines are ignored.
#
# Modified for Linux.  --marekm

# REQUIRED for useradd/userdel/usermod
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define MAIL_DIR and MAIL_FILE,
#   MAIL_DIR takes precedence.
#
#   Essentially:
#      - MAIL_DIR defines the location of users mail spool files
#        (for mbox use) by appending the username to MAIL_DIR as defined
#        below.
#      - MAIL_FILE defines the location of the users mail spool files as the
#        fully-qualified filename obtained by prepending the user home
#        directory before $MAIL_FILE
#
# NOTE: This is no more used for setting up users MAIL environment variable
#       which is, starting from shadow 4.0.12-1 in Debian, entirely the
#       job of the pam_mail PAM modules
#       See default PAM configuration files provided for
#       login, su, etc.
```

Task06:

8. Login as root, and create three users and assign them passwords. Use su- command to switch user and login as these three users one after another, create files within their respective home directories. Try entering the home directories of other users and see what happens. keep a note of your observations.

```
root@DESKTOP-MC3IIQB:~# useradd -m user1
useradd -m user2
useradd -m user3
root@DESKTOP-MC3IIQB:~# passwd user1
passwd user2
passwd user3
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
New password:
Retype new password:
passwd: password updated successfully
New password:
Retype new password:
passwd: password updated successfully
root@DESKTOP-MC3IIQB:~# su - user1
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Oct 27 01:54:43 PKT 2024

  System load:  0.0                Processes:             37
  Usage of /:   0.5% of 1006.85GB  Users logged in:       1
  Memory usage: 6%                 IPv4 address for eth0: 172.30.117.55
  Swap usage:   0%
```

9. Delete all these three users and observe the contents of the related configuration files again

```
habiba@DESKTOP-MC3IIQB:~$ sudo su -
root@DESKTOP-MC3IIQB:~# userdel -r user1
userdel -r user2
userdel -r user3
userdel: user1 mail spool (/var/mail/user1) not found
userdel: user2 mail spool (/var/mail/user2) not found
userdel: user3 mail spool (/var/mail/user3) not found
root@DESKTOP-MC3IIQB:~# cat /etc/passwd | grep user1
cat /etc/passwd | grep user2
cat /etc/passwd | grep user3
root@DESKTOP-MC3IIQB:~# cat /etc/shadow | grep user1
cat /etc/shadow | grep user2
cat /etc/shadow | grep user3
root@DESKTOP-MC3IIQB:~# cat /etc/group | grep user1
cat /etc/group | grep user2
cat /etc/group | grep user3
root@DESKTOP-MC3IIQB:~# ls -ld /home/user1
ls -ld /home/user2
ls -ld /home/user3
ls: cannot access '/home/user1': No such file or directory
ls: cannot access '/home/user2': No such file or directory
ls: cannot access '/home/user3': No such file or directory
root@DESKTOP-MC3IIQB:~#
```

10. Explore the newusers command and see if you have to create 100 users at a time how this command can help you.

The newusers command in Linux enables batch creation of user accounts, making it ideal for generating multiple users (e.g., 100) efficiently.

Key Features:

- **Batch Processing:** Allows defining multiple user accounts in a single input file, creating all users in one command.
- **Input File Structure:** Requires details like username, password, UID, GID, home directory, and shell for each user, ensuring standardized settings.
- **Error Handling:** Reports issues (e.g., duplicate usernames) without stopping the entire process, simplifying troubleshooting.
- **Consistency:** Ensures uniform configuration across all created accounts.

Benefits:

- **Efficiency:** Streamlines user creation for large groups, useful in educational and corporate settings.
- **Simplicity:** Reduces administrative tasks and makes user account management easier.

Task07:

11. Login as root, and create three groups with the name of faculty, staff and students.

```
root@DESKTOP-MC3IIQB:~# groupadd faculty
root@DESKTOP-MC3IIQB:~# groupadd staff
groupadd: group 'staff' already exists
root@DESKTOP-MC3IIQB:~# groupadd students
root@DESKTOP-MC3IIQB:~# cat /etc/group | grep -E 'faculty|staff|students'
staff:x:50:
faculty:x:1001:
students:x:1002:
root@DESKTOP-MC3IIQB:~#
```

12. Create three users and made them member of faculty group

```
root@DESKTOP-MC3IIQB:~# useradd -m user4
root@DESKTOP-MC3IIQB:~# useradd -m user5
root@DESKTOP-MC3IIQB:~# useradd -m user6
root@DESKTOP-MC3IIQB:~# passwd user4
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
root@DESKTOP-MC3IIQB:~# passwd user4
New password:
Retype new password:
passwd: password updated successfully
root@DESKTOP-MC3IIQB:~# passwd user5
New password:
Retype new password:
passwd: password updated successfully
root@DESKTOP-MC3IIQB:~# passwd user6
New password:
Retype new password:
passwd: password updated successfully
root@DESKTOP-MC3IIQB:~# usermod -aG faculty user4
root@DESKTOP-MC3IIQB:~# usermod -aG faculty user5
root@DESKTOP-MC3IIQB:~# usermod -aG faculty user5
root@DESKTOP-MC3IIQB:~# ^C
root@DESKTOP-MC3IIQB:~# getent group faculty
faculty:x:1001:user4,user5
root@DESKTOP-MC3IIQB:~#
```

13. Create three users and made them members of staff group

```
root@DESKTOP-MC3IIQB:~# usermod -aG staff user4
usermod -aG staff user5
usermod -aG staff user6
root@DESKTOP-MC3IIQB:~# getent group staff
staff:x:50:user4,user5,user6
root@DESKTOP-MC3IIQB:~#
```

14. Create three users and made them members of students group

```
root@DESKTOP-MC3IIQB:~# usermod -aG students user4
usermod -aG students user5
usermod -aG students user6
root@DESKTOP-MC3IIQB:~# getent group students
students:x:1002:user4,user5,user6
root@DESKTOP-MC3IIQB:~#
```

15. Give sudo privileges to the users of faculty group by adding them in the sudo group

```
root@DESKTOP-MC3IIQB:~# getent group faculty
faculty:x:1001:user4,user5
root@DESKTOP-MC3IIQB:~# usermod -aG sudo user4
usermod -aG sudo user5
root@DESKTOP-MC3IIQB:~# getent group sudo
sudo:x:27:habiba,user4,user5
root@DESKTOP-MC3IIQB:~#
```