Linux Distributions:

Linux distributions (distros) are different versions or flavors of the Linux operating system, each tailored for specific use cases or preferences. Distributions combine the Linux kernel (core of the operating system) with different software packages, utilities, and tools. Some distros focus on user-friendliness, while others emphasize performance, security, or flexibility. Common Linux distributions include:

1. **Debian** - Known for its stability and simplicity.
2. **Ubuntu** - User-friendly, based on Debian, and widely used in both desktop and server environments.
3. **Fedora** - Sponsored by Red Hat, it's a cutting-edge distro known for integrating the latest technologies.
4. **CentOS** (now replaced by CentOS Stream) - A community-driven distro based on Red Hat Enterprise Linux (RHEL).
5. **Arch Linux** - A rolling release distro focused on simplicity, flexibility, and control for advanced users.
6. **Red Hat Enterprise Linux (RHEL)** - Commercially supported, geared towards enterprise environments.
7. **openSUSE** - A powerful and flexible distribution aimed at developers and system administrators.
8. **Mint** - A user-friendly distro based on Ubuntu, popular among beginners.

Advantages of using Linux systems as servers:

1. **Stability & Reliability**: Linux is known for its long-term stability, making it a preferred choice for servers.
2. **Security**: With strong user permissions, security tools, and regular updates, Linux is considered more secure than many other OSes.
3. **Cost-effective**: Most Linux distributions are open-source and free, which reduces licensing and operational costs.
4. **Performance**: Linux is lightweight and can run on older hardware, making it highly efficient for server tasks.
5. **Customization**: You can configure and optimize Linux servers for specific requirements without much overhead.
6. **Community Support**: A massive global community and forums for troubleshooting and support.
7. **Wide Range of Server Tools**: Linux comes with a vast array of built-in tools and software packages for server management.

Checking current memory usage and running processes:

To check memory usage and running processes, use the command:

```
top
```

or for a more detailed view, use:

```
htop
```

Check the status of the nginx service:

To check if nginx is running, use:

```
systemctl status nginx
```

View the last 100 lines of a log file:

You can use the `tail` command with the `-n` option:

```
tail -n 100 /var/log/<log_file>
```

Search for the keyword "error" around 4:00 AM in a log file:

Use `grep` combined with `awk` or `sed` to search logs based on a time range:

```
grep "error" /var/log/<log_file> | grep "04:00"
```

Append more content to the end of a file:

To append text to a file, you can use `echo` or the `cat` command:

```
echo "New content" >> filename.txt
```

Quickly view the content of a file:

Use `cat`, `less`, or `more` to view a file's content:

```
cat filename.txt
```

or for easier navigation:

```
less filename.txt
```

Delete a directory with files inside:

To delete a directory along with its contents:

```
rm -r directory_name
```

Use `vi` editor and understand interactive and command mode:

- **Interactive mode**: You can navigate the file, make changes, and insert text.
- **Command mode**: You use keyboard shortcuts (like `:wq` to save and exit, `:q!` to quit without saving).

To enter insert mode, press `i`. To return to command mode, press `Esc`.

Kill a process running on port 8080:

To kill a process on a specific port:

1. Find the process ID:

```
lsof -i :8080
```

2. Kill the process by its ID:

```
kill <PID>
```

Check all running processes on ports:

To check which processes are using which ports:

```
netstat -tuln
```

Change file permissions to Execute and Read for all users:

Use the `chmod` command:

```
chmod a+rx filename
```

Change file permissions to read-only for others and read-write for users and guests:

```
chmod 764 filename
```

Provide all permissions to all users:

To give read, write, and execute permissions to all users:

```
chmod a+rwx filename
```

Run a process in the background:

To run a process in the background, append `&` to the command:

```
command &
```

Find where a particular file resides:

To find the file, use:

```
find / -name "syslog-22011994.log"
```

Find where a particular application is installed (Docker, Python, Java):

To locate where a package is installed:

```
which docker
which python
which java
```

Provide/change system variables of Linux:

To set a system variable, use:

```
export VARIABLE_NAME=value
```

View all system variables in Linux:

Use the `env` or `printenv` command:

```
env
```

What is `sudo`?

`sudo` stands for "superuser do" and allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.

Why do we use `sudo`?

`sudo` is used to execute commands that require elevated privileges (admin/root rights) without logging in as the root user.

Display all current network connections and listening ports:

Use the `netstat` or `ss` command:

```
netstat -tuln
```

or

```
ss -tuln
```

Test network connectivity to an external server:

To test connectivity to an external server, use:

```
ping google.com
```

Compare content of two text files:

To compare the content of `file1.txt` and `file2.txt`, use:

```
diff file1.txt file2.txt
```