**Part 1 – Outputs of the following commands**

**1. pwd (Print Working Directory)**

bash

CopyEdit

$ pwd

/home/user

* Displays the current working directory.

**2. cd (Change Directory)**

bash

CopyEdit

$ cd /home/user/Documents

* Changes the working directory to /home/user/Documents.
* Running cd alone takes you to the home directory.

**3. ls (List Files and Directories)**

bash

CopyEdit

$ ls

file1.txt file2.txt Documents Downloads

* Lists files and directories in the current directory.

**4. mkdir (Make Directory)**

bash

CopyEdit

$ mkdir new\_folder

* Creates a new directory named new\_folder.

**5. rm (Remove Files/Directories)**

bash

CopyEdit

$ rm file1.txt

* Deletes file1.txt (no output if successful).

**6. touch (Create an Empty File)**

bash

CopyEdit

$ touch myfile.txt

* Creates an empty file named myfile.txt.

**7. hostname (Display System Name)**

bash

CopyEdit

$ hostname

my-computer

* Shows the hostname of the system.

**8. cat (Concatenate and Display File Content)**

bash

CopyEdit

$ cat file.txt

Hello, this is a text file.

* Displays the contents of file.txt.

**9. chmod (Change File Permissions)**

bash

CopyEdit

$ chmod 755 script.sh

* Changes the permission of script.sh (no output if successful).

**10. echo (Print Text to Terminal)**

bash

CopyEdit

$ echo "Hello, World!"

Hello, World!

* Prints text to the terminal.

**11. grep (Search for Patterns in Files)**

bash

CopyEdit

$ grep "error" log.txt

Error: File not found

* Searches for the word "error" in log.txt.

**12. fgrep (Fixed String Search in Files)**

bash

CopyEdit

$ fgrep "Hello" file.txt

Hello, this is a text file.

* Similar to grep, but treats search patterns as **fixed strings**, not regular expressions.

**13. mv (Move/Rename Files)**

bash

CopyEdit

$ mv oldfile.txt newfile.txt

* Renames oldfile.txt to newfile.txt.

**14. cp (Copy Files and Directories)**

bash

CopyEdit

$ cp file.txt backup.txt

* Copies file.txt to backup.txt.

**15. more (View Large Files Page by Page)**

bash

CopyEdit

$ more largefile.txt

-- output displayed page by page --

* Displays a file page by page (press Space to scroll).

**16. less (View Files with Navigation)**

bash

CopyEdit

$ less largefile.txt

-- output displayed, can scroll up/down --

* Similar to more, but allows **scrolling up**.

**17. wc (Word Count in a File)**

bash

CopyEdit

$ wc file.txt

10 25 150 file.txt

* Displays the number of **lines, words, and characters** in file.txt.

**18. awk (Text Processing)**

bash

CopyEdit

$ awk '{print $1}' data.txt

John

Alice

Mike

* Extracts and prints the first column from data.txt.

**19. sed (Stream Editor for Text)**

bash

CopyEdit

$ sed 's/Hello/Hi/' file.txt

Hi, this is a text file.

* Replaces "Hello" with "Hi" in file.txt.

**20. tail (Display Last Lines of a File)**

bash

CopyEdit

$ tail log.txt

-- shows last 10 lines of log.txt --

* Displays the last 10 lines of log.txt

**Part 2. Answers to the following Questions: (you need to supply commands)**

1. **How to navigate to a Specific Directory?**

**Command**: cd /path/to/directory

**2.How to See Detailed Information About Files and Directories Using ls?**

The ls command is used to list files and directories. To see **detailed information**, you can use various options.

**1️⃣ Long Listing Format (-l option)**

bash

* **First column**: File permissions (-rw-r--r--)
* **Second column**: Number of hard links (1)
* **Third & Fourth columns**: Owner and group (user user)
* **Fifth column**: File size (1234 bytes)
* **Sixth column**: Last modified date (Mar 23 10:00)
* **Last column**: File/directory name (file.txt or Documents)

**3.How to Create Multiple Directories in Linux Using mkdir Command?**

The mkdir (make directory) command is used to create directories (folders) in Linux. You can create multiple directories at once using different methods.

**1️⃣ Create Multiple Directories in One Command**

bash

CopyEdit

mkdir dir1 dir2 dir3

* Creates **three directories**: dir1, dir2, and dir3 in the current location.

**4.How to Remove Multiple Files at Once Using rm?**

The rm (remove) command in Linux can delete multiple files at once using different methods.

**1️⃣ Remove Multiple Files by Name**

bash

CopyEdit

rm file1.txt file2.txt file3.txt

* Deletes file1.txt, file2.txt, and file3.txt in the current directory.

**2️⃣ Remove Files Using Wildcards (\* and ?)**

**Delete All .txt Files**

bash

CopyEdit

rm \*.txt

* Deletes **all .txt files** in the current directory.

**Delete Files That Start with "log"**

bash

CopyEdit

rm log\*

* Deletes log1.txt, log2.txt, log\_backup.log, etc.

**Delete Files with a Single Character Difference**

bash

CopyEdit

rm file?.txt

* Deletes file1.txt, file2.txt, but **not** file10.txt.

**3️⃣ Remove Files Interactively (-i option)**

bash

CopyEdit

rm -i file1.txt file2.txt

* Asks for **confirmation** before deleting each file.

**4️⃣ Force Remove Files Without Confirmation (-f option)**

bash

CopyEdit

rm -f file1.txt file2.txt

* Deletes files **without asking for confirmation**.

**5️⃣ Remove All Files in a Directory (But Not the Directory Itself)**

bash

CopyEdit

rm -rf /path/to/directory/\*

* Deletes **all files and subdirectories** inside /path/to/directory/.

**6️⃣ Remove Multiple Files Using a Loop**

bash

CopyEdit

for file in file1.txt file2.txt file3.txt; do rm "$file"; done

* Deletes each specified file using a loop.

**5.Can rm Be Used to Delete Directories?**

Yes, the rm command can delete **directories**, but you need to use specific options because rm is primarily designed for deleting files.

**1️⃣ Remove an Empty Directory**

bash

CopyEdit

rmdir directory\_name

* This only works for **empty** directories.

**2️⃣ Remove a Non-Empty Directory (-r option)**

bash

CopyEdit

rm -r directory\_name

* The -r (recursive) option **removes the directory and all its contents** (files + subdirectories).

**3️⃣ Force Remove a Directory Without Confirmation (-rf option)**

bash

CopyEdit

rm -rf directory\_name

* **Deletes everything** inside the directory **without asking for confirmation**.

In Linux, the cp (copy) command is used to copy files and directories.

**1️⃣ Copy a Single File**

bash

CopyEdit

cp file1.txt /destination/path/

* Copies file1.txt to /destination/path/.

**2️⃣ Copy Multiple Files to a Directory**

bash

CopyEdit

cp file1.txt file2.txt file3.txt /destination/path/

* Copies multiple files to the specified directory.

**3️⃣ Copy a File and Rename It**

bash

CopyEdit

cp file1.txt newfile.txt

* Copies file1.txt and saves it as newf
* Asks **confirmation** before overwriting an existing file.

**6️⃣ Copy Files While Preserving Attributes (-p option)**

bash

CopyEdit

cp -p file1.txt /destination/path/

* Preserves **timestamps, ownership, and permissions**.

**7️⃣ Copy Files with Verbose Output (-v for Details)**

bash

CopyEdit

cp -v file1.txt /destination/path/

* Shows detailed output of what’s being copied.

**8️⃣ Copy Only Newer Files (-u for Update Mode)**

bash

CopyEdit

cp -u file1.txt /destination/path/

* Copies only **if the file is newer** than the destination file.

**9️⃣ Copy All Files Matching a Pattern**

bash

CopyEdit

cp \*.txt /destination/path/

* Copies **all .txt files** to the destination.

**7. How to Move Multiple Files in Linux Using the mv Command?**

**The mv (move) command is used to move multiple files from one location to another in Linux.**

**1️⃣ Move Multiple Files to a Directory**

**bash**

**CopyEdit**

**mv file1.txt file2.txt file3.txt /destination/path/**

* **Moves file1.txt, file2.txt, and file3.txt to /destination/path/.**

**2️⃣ Move All Files of a Specific Type**

**bash**

**CopyEdit**

**mv \*.txt /destination/path/**

* **Moves all .txt files to /destination/path/.**

**3️⃣ Move Files That Start with a Specific Prefix**

**bash**

**CopyEdit**

**mv log\* /destination/path/**

* **Moves all files starting with "log" (e.g., log1.txt, log\_backup.log) to /destination/path/.**

**4️⃣ Move Multiple Files and Rename One of Them**

**bash**

**CopyEdit**

**mv file1.txt file2.txt /destination/path/newfile.txt**

* **Moves file1.txt and file2.txt to /destination/path/, renaming file2.txt to newfile.txt.**

**5️⃣ Move an Entire Directory**

**bash**

**CopyEdit**

**mv myfolder /destination/path/**

* **Moves myfolder and all its contents to /destination/path/.**

**6️⃣ Prevent Overwriting Existing Files (-i option)**

**bash**

**CopyEdit**

**mv -i file1.txt file2.txt /destination/path/**

* **Asks confirmation before overwriting existing files.**

**7️⃣ Move Files with Verbose Output (-v optio**

In Linux, the mv (move) command is used to **rename** files and directories.

**8.How to Move Multiple Files in Linux Using the mv Command?**

The mv (move) command is used to **move multiple files** from one location to another in Linux.

**1️⃣ Move Multiple Files to a Directory**

bash

CopyEdit

mv file1.txt file2.txt file3.txt /destination/path/

* Moves file1.txt, file2.txt, and file3.txt to /destination/path/.

**2️⃣ Move All Files of a Specific Type**

bash

CopyEdit

mv \*.txt /destination/path/

* Moves **all .txt files** to /destination/path/.

**3️⃣ Move Files That Start with a Specific Prefix**

bash

CopyEdit

mv log\* /destination/path/

* Moves all files **starting with "log"** (e.g., log1.txt, log\_backup.log) to /destination/path/.

**4️⃣ Move Multiple Files and Rename One of Them**

bash

CopyEdit

mv file1.txt file2.txt /destination/path/newfile.txt

* Moves file1.txt and file2.txt to /destination/path/, renaming file2.txt to newfile.txt.

**5️⃣ Move an Entire Directory**

bash

CopyEdit

mv myfolder /destination/path/

* Moves myfolder and all its contents to /destination/path/.

**6️⃣ Prevent Overwriting Existing Files (-i option)**

bash

CopyEdit

mv -i file1.txt file2.txt /destination/path/

* Asks **confirmation** before overwriting existing files.

**7️⃣ Move Files with Verbose Output (-v option)**

**9.How to Create Multiple Empty Files Using touch Command in Linux?**

The touch command is used to **create empty files** or update the timestamps of existing files.

**1️⃣ Create Multiple Empty Files**

bash

CopyEdit

touch file1.txt file2.txt file3.txt

* Creates file1.txt, file2.txt, and file3.txt.

**2️⃣ Create Multiple Files with a Numeric Pattern**

bash

CopyEdit

touch file{1..5}.txt

* Creates file1.txt, file2.txt, file3.txt, file4.txt, and file5.txt.

**3️⃣ Create Multiple Files with Different Extensions**

**10.How to View the Content of Multiple Files in Linux?**

There are several commands to view the content of multiple files in Linux. Here are the most commonly used methods:

**1️⃣ Using cat (Concatenate and Display Files)**

bash

CopyEdit

cat file1.txt file2.txt

* Displays the contents of **file1.txt** and **file2.txt** one after another.

bash

CopyEdit

cat \*.txt

* Displays the content of **all .txt files** in the directory.

**2️⃣ Using more (View Large Files Page by Page)**

bash

CopyEdit

more file1.txt file2.txt

* Displays the content one page at a time (use Space to go to the next page and q to quit).

**11.How to Create a File and Add Content in Linux Using the cat Command?**

The cat command is mainly used for **viewing files**, but it can also be used to **create** and **add content** to files.

**1️⃣ Create a New File and Add Content**

bash

CopyEdit

cat > filename.txt

* This **creates** a new file and allows you to **type content**.
* After typing, **press Ctrl + D** to save and exit.

**Example:**

bash

CopyEdit

cat > myfile.txt

**Then type:**

kotlin

CopyEdit

Hello, this is a test file.

This is the second line.

**Press Ctrl + D to save and exit.**

**12.How to Append the Contents of One File to Another Using the cat Command?**

To **append** the content of one file to the end of another file, use the >> operator with the cat command.

**1️⃣ Append One File to Another**

bash

CopyEdit

cat source.txt >> destination.txt

* **Appends** the content of source.txt to the end of destination.txt.
* If destination.txt **does not exist**, it will be created.

**Example:**

bash

CopyEdit

cat notes.txt >> summary.txt

* Appends the content of notes.txt to summary.txt.

**2️⃣ Append Multiple Files to Another File**

bash

CopyEdit

cat file1.txt file2.txt >> combined.txt

* Appends file1.txt and file2.txt into combined.txt.

**3️⃣ Verify the Appended Content**

bash

CopyEdit

cat destination.txt

* Displays the updated file content.

**4️⃣ Append Input Directly to a File**

bash

CopyEdit

cat >> destination.txt

* Allows you to **type text manually** and append it to destination.txt.
* **Press Ctrl + D** to save and exit.

**Example:**

bash

CopyEdit

cat >> notes.txt

**Then type:**

csharp

CopyEdit

This is an extra note.

**Press Ctrl + D to save.**

**13.How to Use cat Command When the File Has a Lot of Content?**

If a file is **too large** to fit in the terminal, displaying it all at once using cat may not be ideal. Instead, you can use **paging and filtering commands** to view the content effectively.

**1️⃣ Use less for Scrollable Viewing**

bash

CopyEdit

cat largefile.txt | less

or

bash

CopyEdit

less largefile.txt

* **Allows scrolling** up (↑) and down (↓) through the content.
* **Press q** to exit.

**2️⃣ Use more to View Page by Page**

bash

CopyEdit

cat largefile.txt | more

or

bash

CopyEdit

more largefile.txt

* **Press Space** to move to the next page.
* **Press q** to exit.

**3️⃣ Show Only the First Few Lines (head)**

bash

CopyEdit

cat largefile.txt | head -n 20

or

bash

CopyEdit

head -n 20 largefile.txt

* Displays the **first 20 lines** of the file.

**4️⃣ Show Only the Last Few Lines (tail)**

**14.How to Merge Contents of Multiple Files Using the cat Command in Linux?**

The cat command can be used to **merge multiple files** into one. Here are different ways to do it:

**1️⃣ Merge Multiple Files into a New File**

bash

CopyEdit

cat file1.txt file2.txt > merged.txt

* Combines file1.txt and file2.txt into merged.txt.
* **Overwrites** merged.txt if it already exists.

**2️⃣ Append Multiple Files to an Existing File**

bash

CopyEdit

cat file1.txt file2.txt >> existingfile.txt

* **Appends** the contents of file1.txt and file2.txt to existingfile.txt.

**3️⃣ Merge All Files of a Specific Type**

bash

CopyEdit

cat \*.txt > all\_text\_files\_merged.txt

* Merges **all .txt files** in the directory into all\_text\_files\_merged.txt.

**4️⃣ Merge Files and Display Output on Terminal**

bash

CopyEdit

cat file1.txt file2.txt

* Prints the **combined content** of file1.txt and file2.txt to the terminal.

**5️⃣ Merge Files and Show Line Numbers**

bash

CopyEdit

cat -n file1.txt file2.txt > merged.txt

* Merges files and **adds line numbers** to the output.

**6️⃣ Verify the Merged File**

bash

CopyEdit

cat merged.txt

* Displays the content of merged.txt to confirm merging.

**15.How to Use cat Command to Append to an Existing File in Linux?**

The cat command allows you to **append** content to an existing file using the >> operator.

**1️⃣ Append Another File to an Existing File**

bash

CopyEdit

cat source.txt >> destination.txt

* Appends the content of source.txt to the end of destination.txt.
* If destination.txt **does not exist**, it will be created.

**Example:**

bash

CopyEdit

cat notes.txt >> summary.txt

* Appends notes.txt to summary.txt.

**2️⃣ Append Multiple Files to an Existing File**

bash

CopyEdit

cat file1.txt file2.txt >> destination.txt

* Merges file1.txt and file2.txt into destination.txt without overwriting it.

**3️⃣ Manually Append Text to an Existing File**

bash

CopyEdit

cat >> filename.txt

* Allows you to **type text manually** and append it to filename.txt.
* **Press Ctrl + D** to save and exit.

**Example:**

bash

CopyEdit

cat >> notes.txt

**16.What is “chmod 777 “, “chmod 755” and “chmod +x “or “chmod a+x”?**

**Understanding chmod Command in Linux**

The chmod (change mode) command is used to **change file permissions** in Linux.

Each file and directory has **three types of permissions** for **three types of users**:

* **r (read)** → View file contents.
* **w (write)** → Modify or delete the file.
* **x (execute)** → Run the file as a program.

**User Categories:**

* **Owner (u)** → The creator of the file.
* **Group (g)** → Users in the same group.
* **Others (o)** → Everyone else.
* **All (a)** → Applies to everyone (owner, group, and others).

**1️⃣ chmod 777 filename → Give Full Permissions**

bash

CopyEdit

chmod 777 filename

* 7 → **Read (4) + Write (2) + Execute (1) = 7**
* This means **everyone (owner, group, others) has full permissions**.

**17.How to Find the Number of Lines That Match a Given String/Pattern in Linux?**

To count the number of lines that match a specific **string or pattern**, use the grep command with the -c (count) option.

**1️⃣ Count Lines Matching a Specific String**

bash

CopyEdit

grep -c "pattern" filename.txt

* Searches for "pattern" in filename.txt and returns the **number of matching lines**.

**Example:**

bash

CopyEdit

grep -c "error" logfile.txt

* Counts how many lines contain the word "error" in logfile.txt.

**2️⃣ Count Lines Matching a Case-Insensitive String (-i option)**

bash

CopyEdit

grep -ci "pattern" filename.txt

* **Ignores case differences** when searching.

**Example:**

bash

CopyEdit

grep -ci "warning" logfile.txt

* Counts occurrences of "warning", "Warning", or "WARNING", etc.

**3️⃣ Count Lines Using Regular Expressions**

bash

CopyEdit

grep -cE "pattern1|pattern2" filename.txt

* Counts lines containing **either** pattern1 or pattern2.

**18.How to Display Files That Contain a Given String/Pattern in Linux?**

To **find files** that contain a specific string or pattern, use the grep command with the -l (lowercase L) option.

**1️⃣ Show Files That Contain a Specific String**

bash

CopyEdit

grep -l "pattern" \*.txt

* Searches for "pattern" in all .txt files and **displays only the filenames** that contain it.

**Example:**

bash

CopyEdit

grep -l "error" logs/\*.log

* Lists all .log files in the logs directory that contain "error".

**2️⃣ Search in All Files in a Directory**

bash

CopyEdit

grep -l "pattern" \*

* Displays all filenames in the **current directory** that contain "pattern".

**3️⃣ Search Recursively in Subdirectories (-r option)**

bash

CopyEdit

grep -rl "pattern" /path/to/directory

* Searches for "pattern" in **all files and subdirectories** under /path/to/directory.

**Example:**

bash

CopyEdit

grep -rl "database error" /var/logs/

* Finds all files under /var/logs/ that contain "database error".

**19.How to Show the Line Number of a Matched Line in a File?**

**To display the line number of a file along with the matched line, use the grep command with the -n option.**

**1️⃣ Show Line Numbers for Matches**

**bash**

**CopyEdit**

**grep -n "pattern" filename.txt**

* **Displays the matching lines along with their line numbers.**

**Example:**

**bash**

**CopyEdit**

**grep -n "error" logfile.txt**

**Output:**

**go**

**CopyEdit**

**5:error in database connection**

**12:error code 500**

* **Shows line numbers (5 and 12) where "error" appears in logfile.txt.**

**2️⃣ Search in All Files in a Directory**

**bash**

**CopyEdit**

**grep -n "pattern" \***

* **Displays matches and line numbers from all files in the current directory.**

**3️⃣ Recursive Search in Subdirectories (-r option)**

**bash**

**CopyEdit**

**grep -rn "pattern" /path/to/directory**

* **Searches all files in a directory and its subdirectories.**
* **Displays filenames, line numbers, and matched lines.**

**Example:**

**bash**

**CopyEdit**

**grep -rn "database error" /var/logs/**

**Output:**

**pgsql**

**CopyEdit**

**/var/logs/system.log:45:database error detected**

**/var/logs/app.log:120:critical database error occurred**

* **"database error" appears at line 45 in system.log and line 120 in app.log.**

**20.How to Match Lines That Start with a Specific String Using grep in Linux?**

To **find lines that start with a specific string**, use the **^ (caret)** symbol in grep, which represents the **beginning of a line**.

**1️⃣ Match Lines That Start with a Specific String**

bash

CopyEdit

grep "^pattern" filename.txt

* **^pattern** → Ensures the line **starts** with "pattern".
* Searches filename.txt and prints lines **beginning with "pattern"**.

**Example:**

bash

CopyEdit

grep "^Error" logfile.txt

**Output:**

javascript

CopyEdit

Error: Connection failed

Error: Invalid input detected

* Displays only lines that **start with** "Error".

**2️⃣ Match Lines That Start with a Word (Case-Insensitive)**

bash

CopyEdit

grep -i "^pattern" filename.txt

* **-i** → Makes the search **case-insensitive**.

**Example:**

bash

CopyEdit

grep -i "^warning" system.log

* Matches "Warning", "warning", and "WARNING".

**3️⃣ Match Lines That Start with Multiple Words (-E for Extended Regex)**

bash

CopyEdit

grep -E "^word1|^word2" filename.txt

* Searches for **multiple words** at the start of a line.

**Example:**

bash

CopyEdit

grep -E "^Error|^Warning" logfile.txt

* Matches lines starting with **"Error"** or **"Warning"**.

**4️⃣ Search in Multiple Files**

bash

CopyEdit

grep "^pattern" file1.txt file2.txt

* Finds matching lines in **multiple files**.

**21.Can the sort Command Sort Files in Descending Order by Default?**

No, the sort command **by default sorts in ascending order** (A-Z, 0-9). However, you can **explicitly specify descending order** using the -r (reverse) option.

**Sorting in Descending Order (-r Option)**

bash

CopyEdit

sort -r filename.txt

* -r → Sorts the file in **reverse (descending) order**.

**Example:**

bash

CopyEdit

cat names.txt

nginx

CopyEdit

Alice

Charlie

Bob

David

bash

CopyEdit

sort -r names.txt

**Output:**

nginx

CopyEdit

David

Charlie

Bob

Alice

* Names are sorted in **reverse (descending) order**.

**Sorting Numbers in Descending Order (-nr Option)**

bash

CopyEdit

sort -nr numbers.txt

* -n → Sorts numbers (instead of treating them as text).
* -r → Reverses the order (**descending**).

**Example:**

bash

CopyEdit

cat numbers.txt

CopyEdit

23

5

98

12

bash

CopyEdit

sort -nr numbers.txt

**Output:**

CopyEdit

98

23

12

5

* Numbers are sorted from **largest to smallest**.

**Sorting by a Column in Descending Order (-k and -r)**

If your file has multiple columns, you can sort by a specific column in descending order.

bash

CopyEdit

sort -k2 -nr filename.txt

* -k2 → Sorts by **second column**.
* -nr → Sorts **numerically in descending order**.

**Example (Sorting by Age in Descending Order)**

bash

CopyEdit

cat employees.txt

nginx

CopyEdit

Alice 30

Bob 25

Charlie 40

David 35

bash

CopyEdit

sort -k2 -nr employees.txt

**Output:**

nginx

CopyEdit

Charlie 40

David 35

Alice 30

Bob 25

* Sorted by **age (column 2) in descending order**.

**22.How to Sort a File Based on a Specific Column Using the sort Command?**

The sort command allows you to **sort files by a specific column** using the -k option.

**1️⃣ Sorting by a Specific Column (Text Sorting)**

bash

CopyEdit

sort -kN filename.txt

* -kN → Sorts by **column N** (default is alphabetical sorting).
* Columns are **separated by whitespace (spaces or tabs)**.

**Example (Sort by Second Column - Names)**

bash

CopyEdit

cat employees.txt

CopyEdit

101 Alice HR

102 Charlie Sales

103 Bob IT

104 David Marketing

bash

CopyEdit

sort -k2 employees.txt

**Output:**

CopyEdit

101 Alice HR

103 Bob IT

102 Charlie Sales

104 David Marketing

* Sorted alphabetically by the **second column (names)**.

**2️⃣ Sorting by a Specific Column (Numeric Sorting)**

bash

CopyEdit

sort -kN -n filename.txt

* -n → Sorts **numerically** instead of alphabetically.

**Example (Sort by Age - Column 2)**

bash

CopyEdit

cat employees.txt

nginx

CopyEdit

Alice 30

Charlie 40

Bob 25

David 35

bash

CopyEdit

sort -k2 -n employees.txt

**Output:**

nginx

CopyEdit

Bob 25

Alice 30

David 35

Charlie 40

* Sorted by **age (column 2) in ascending order**.

**3️⃣ Sorting by a Specific Column in Descending Order**

bash

CopyEdit

sort -kN -nr filename.txt

* -r → **Reverses the order (descending)**.

**Example (Sort by Age in Descending Order)**

bash

CopyEdit

sort -k2 -nr employees.txt

**Output:**

nginx

CopyEdit

Charlie 40

David 35

Alice 30

Bob 25

* Sorted by **age in descending order**.

**4️⃣ Sorting by a Column with a Custom Delimiter (-t)**

By default, sort assumes **whitespace** as the separator. Use -t to specify a **custom delimiter** (e.g., , or :).

bash

CopyEdit

sort -t',' -kN filename.txt

* -t',' → Sets **comma (,) as the delimiter**.
* -kN → Sorts by **column N**.

**Example (Sort by Salary in a CSV File)**

bash

CopyEdit

cat employees.csv

CopyEdit

Alice,HR,5000

Charlie,Sales,7000

Bob,IT,6000

David,Marketing,5500

bash

CopyEdit

sort -t',' -k3 -n employees.csv

**Output:**

CopyEdit

Alice,HR,5000

David,Marketing,5500

Bob,IT,6000

Charlie,Sales,7000

* Sorted by **salary (column 3) in ascending order**.