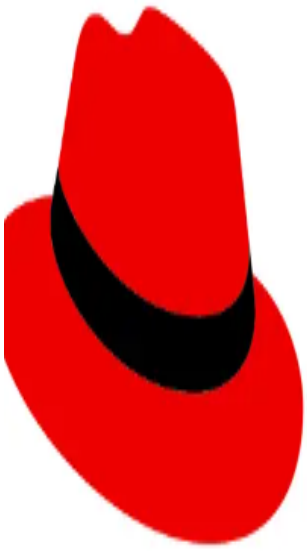


Red Hat Enterprise Linux 9

RHCSA – SA 1 LAB BOOK

Chapter 5 - Creating, Viewing, and Editing Text Files

Redirecting Output to a File or program:

A process structure is constructed with numbered channels (file descriptors)to manage open files.

Processes connect to files to reach data content or devices these files represent.

File descriptors: A process uses numbered channels called as filedescriptors.

There are 3 file descriptors to start with as below:

a) **Standard Input (channel 0):** Reads input from the keyboard.
b) **Standard Output (channel 1):** Sends normal output to the terminal.
c) **Standard Error (Channel 2):** Sends error messages to the terminal.

If program opens separate connections to other files, it may use higher- numbered file descriptors.

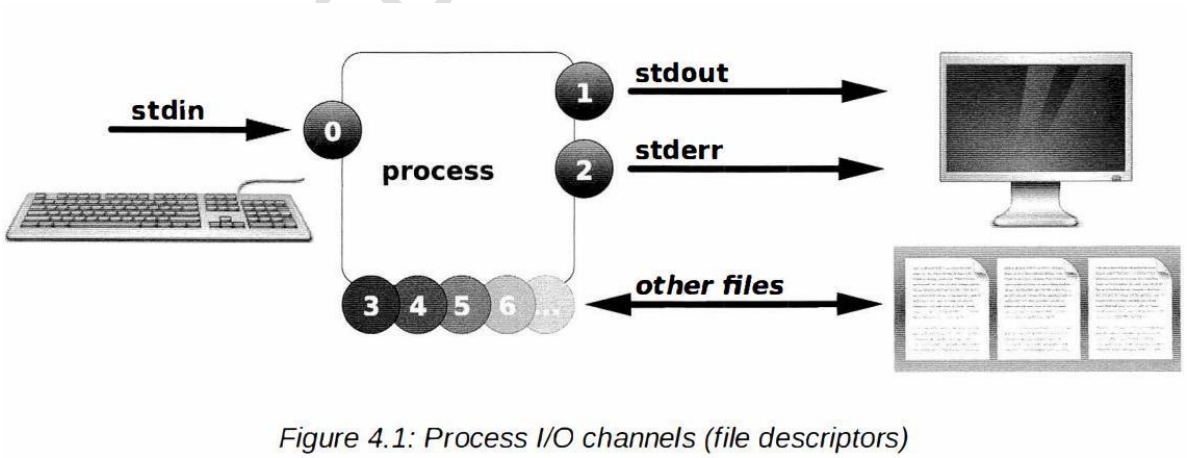


Figure 4.1: Process I/O channels (file descriptors)

These numeric numbers are the channel associated with terminal, they are identified as file descriptor.

Note:

Operator used for re-directions are as below:

A] For Output / Error = (>), (>>)

B] For Input = (<)

A] For Output / Error = (>), (>>) Redirection:

1. **By default ls -l command displays the output on screen.**

   Redirect the output to the file named output

   Command: ls –l > output

   Explanation: If output file is not present then command will create it, if it is already available then command will overwrite it.

2. **To append the output to already created file use (>>)**
   Command: date >> output
   Explanation: Here we have not used file descriptor 1 because it is assumed that 1 is present.

3. **Redirect error message to file error-messages**
   Command: ls –l xyz
   Explanation: If file with name 'xyz' is available then it shows the output but if the file is not available then error message will be displayed on screen.
   Command: ls –l xyz 2> error-messages
   Whereas,
   2 = Is a file descriptor for error**.**
   2>> = To append error messages in same file "2>> '' operator is used.

   Note: Some commands simultaneously gives output and error.

4. **Redirect output to one file and error messages to another file.**
   Command: find / -user student >> output 2>> error-messages

Explanation: Above command finds all the files having ownership set to Harry. The o/p of the command is redirected to 'output' file and error messages will be redirect to 'error' file.

5. **Redirect output to one file and ignore error messages**
   Command: find / -user student >> output 2>> /dev/null
   Explanation: Above command finds all the files having ownership set to Harry. The o/p of the command is redirected to 'output' file and error messages will be deleted.

6. **Redirect output and error messages to same file.**
   Command: find / -user student >> output 2>&1
   Explanation: Both output and error messages are stored in output file.

7. **Display output on screen and redirect error messages to file.**
   Command: find / -user student 2>> error-messages

B] For Input = (<) Redirection:

1. **Command: wc < /etc/passwd**

Explanation:

It returns word count, line count.

This command takes file name as input.

Note: I/O redirection is mostly used to restore database from backup file.

C] Constructing pipelines (|) (Shift+l)

- A pipeline is a sequence of one or more commands by |, the pipe character.
- A pipe connects the standard output of the first command to the standard input of the next command.

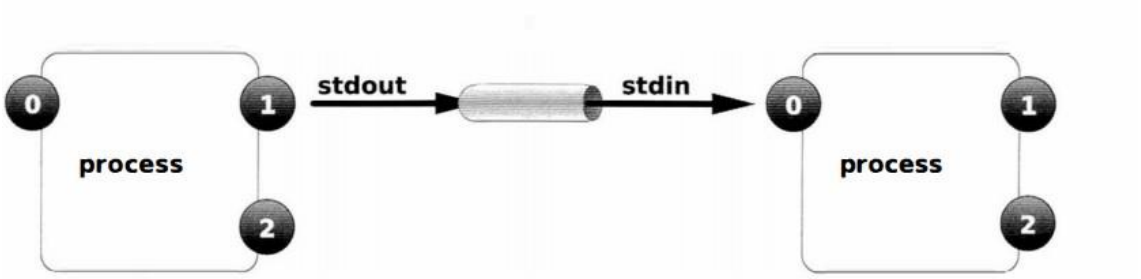


Figure 4.8: Process I/O piping

Example based on pipelines:

1. Sending output of one command as an input to other command

Command: cat /etc/passwd | wc

Command: cat /etc/passwd | grep root

2. Many commands can be combined using pipe. But output of last command will be displayed on screen

Command: cat /etc/passwd | grep root | wc

3. Sends the output to file1.doc and displays it on screen as well.

Command: ls –l | tee file1.doc

D] Tee command:

- In a pipeline, tee will copy its standard input to its standard output and will also redirect its standard output to the files named as arguments to the command.
- If data is imagined to be like water flowing through a pipeline, tee can be visualized as a "T" joint in the pipe which directs output in two directions.

Example based on Tee Command:

1. Sending output to one or more location.

tee command can be only used with pipe

Command: tee

E] VIM Editor:

What is vim editor?

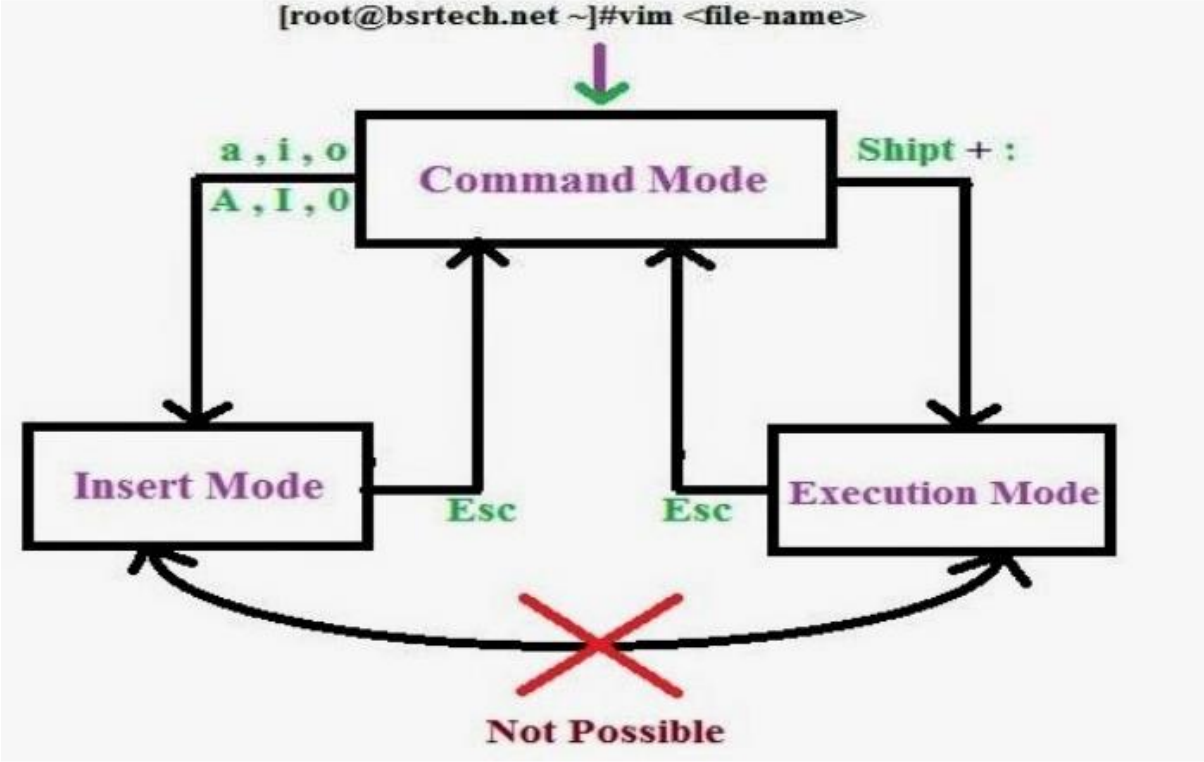
- Vim stands for VI Improved, as vim comes with many features not found in the original vi, while still remaining (mostly) backward- compatible.
- Vim is highly extensible.

Why learn vim editor?

- Every system administrator will have a preference for a text editor. Some will prefer gedit, others like nano, and there even are people who prefer emacs.
- Vim or vi for one simple reason: It's the editor that one can count onto be installed on whatever system is begin worked on.

Different Modes:

1. **Command Mode:** Used for cursor navigations, cut, copy, and paste operations as well as for other text manipulation.
2. **Insert Mode:** We can edit the file.
3. **Execution Mode:** Used to save the file, quite the editor.



1. **Command Mode:**

Options used to move cursor in command mode

1 i: insert before cursor
2 l: moves cursor to beginning of the line and start inserting
3 a: insert after cursor
4 A: moves cursor to the end of line and starts inserting
5 o: opens new line below the cursor
6 O: opens new line above the cursor
7. G: Moves cursor to the last line of the file
8. gg: Moves cursor to the first line of the file

Other command mode operations: Copy, Paste, Delete the content.

Copy: 1 line = yy

a) If you want to copy single line then use the below command

Command: Esc + yy ➔ go to the next line ➔

Esc + Ctrl + P

b) If you want to copy double line then use the below command

Command: Esc + 2yy➔ go to the next line➔

Esc + Ctrl + P

Whereas,

P: above the cursor

p: below the cursor

Delete:

a) If you want to delete a character then use the below command

Command: Esc + x ➔ go to the next line ➔

Esc + Ctrl + P

b) If you want to delete a single line then use the below command

Command: Esc + dd➔ go to the next line➔

Esc + Ctrl + P

c) If you want to delete 2 lines then use the below command

Command: Esc + 2dd➔go to the next line➔

Esc + Ctrl + P

Undo = Use Esc + u

Redo = Use Ctrl + r

2. **Insert Mode:**
   Press the button from keyboard “i” and immediately in the vim editor box you will be able to see the “insert” name which is highlighted.

3. **Execution Mode:** Use the **“Esc”** button first and then perform the below commands in **vi editor.**
   1. : w: write
   2. :wq: write and quite
   3. : wq! : write and quite forcefully
   4. : q : quite without saving
   5. : q! : Quite without saving forcefully
   6. : x: same like wq
   7. : w <filename>: save as
   8. : Set number : will give line number in the file

4. **Visual Mode:**

   Selecting the content:

   - Character wise selection: v
   - Line wise selection: shift + v
   - Block wise selection: ctrl + v
   - To cut the selected text use‘d’, to copy selected text press ‘y’ and to paste press ‘p’.

   Search:

   - /keyword - Forward search
   - ?keyword – Backward search

   Search and replace:

   - Command: ls –l > output.txt

   Search keyword on particular line replace 1st occurrence of word.

   Syntax: :< Line No>s/<word to search>/<word to replace>

   Command: 7s/student/root

Search keyword on particular line and replace all occurrence of word with case insensitivity

Syntax: :< Line No>s/<word to search>/<word to replace>/gi

Command =:7s/student/root/gi

Search keyword in an entire document and replace all occurrence of word with case insensitivity

Syntax =:%s/<word to search>/<word to replace>/g

To remove the selection (Highlighting) in vim editor use

Command =: noh

F] Type of file:

In Linux there is no need to provide extensions to file.

To check the type of file use the below command:

Command: file <name of file> / <name of directory>

Examples based on type of file commnad:

Command 1 :file /etc/passwd

Command 2: file /bin

Command 3: file /etc

Command 4: file /dev/vda

END