I/O redirection

Standard input , standard output

Standard input (stdin) device is the keyboard ,voice assistant , file

Standard output (stdout) device is the monitor

Use of redirection

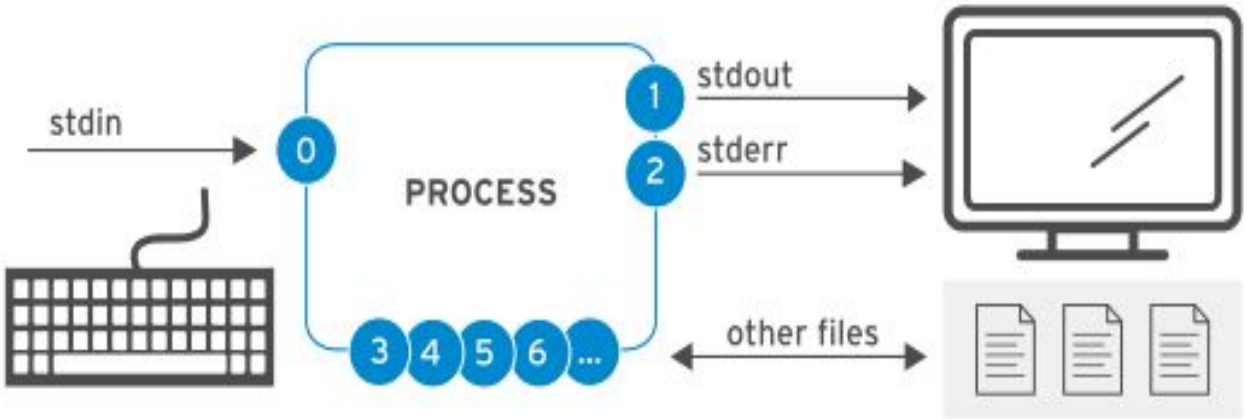
- Output can save from file

- Output can send to mail

- Output can discard

Standard input , standard output, standard error

- Standard input (channel 0) reads input from the Keyboard
- Standard output (channel 1) sends normal output to the terminal
- Standard error (channel 2) sends error messages to the terminal



I/O redirection

- I/O redirection changes how the process gets its input or output
- you can use redirection to discard output or errors, so they are not displayed on the terminal or saved.
- If you want to discard messages, the special file /dev/null quietly discards channel output redirected to it and is always an empty file.

Output redirection Operators

> file	redirect stdout to overwrite a file
>> file	redirect stdout to append to a file
2> file	redirect stderr to overwrite a file
2> /dev/null	discard stderr error messages by redirecting to /dev/null

&> file	redirect stdout and stderr to overwrite the same file redirect stdout and stderr to append to the same file
&>> file	redirect stdout and stderr to append to the same file

Example of Output redirection

Save a time for later reference

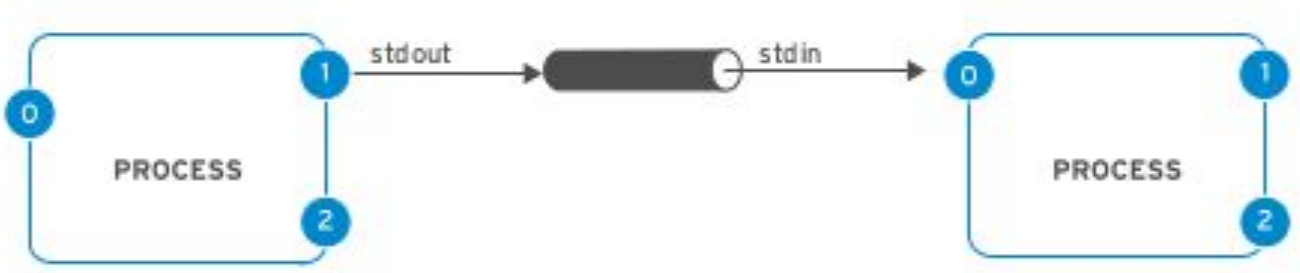
```
[user@host ~]$ date > /tmp/saved-timestamp
```

Concatenate four files into one

```
[user@host ~]$ cat file1 file2 file3 file4 > /tmp/all-four-in-one
```

Constructing pipelines

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



Pipeline with redirection

head will output the first 5 lines of output from ip a , with the final result redirected to a file.

```
[user@host ~]$ ip a | head -5 > /tmp/io.txt
```