STDIN (0) : It is what you type on terminal
e.g. whoami

STDOUT (1) : return value of stdin
e.g. stdout of 'whoami' is root or your 'username'

STDERR (2) :

READ_Link :
http://linuxcommand.org/lts0060.php

Note : cd ~/darshitp/Desktop ---- ~ means home directory

cd /home/$(whoami) -- stdout of whoami is being fed as stdin to cd command

To redirect stdout to 'NOWHERE' :
cat whoami > /dev/null

REDIRECT STANDARD ERROR TO STANDARD OUTPUT OR COMMAND :

2>
Redirect standard error

2>&1
Redirect standard error to standard output

<
Redirect standard inout

|
pipe standard output to another command

>>
append to standard output

2>&1|
Pipe standard output and error to another command

Note : ps -ef will give the list of running processes on the system

Suppose you want to kill firefox process :
You do :
ps ef | grep firefox ----------you will get the process id

kill <process id> ---- use that process id to kill the firefox

Suppose in 'ps -ef' command' you want to just print all the process id :

-ps -ef | awk -F" " '{print$2}' ---F means delimiter. " " specifies 'space' as a delimeter and 'print $2' specifies 2nd field to be printed

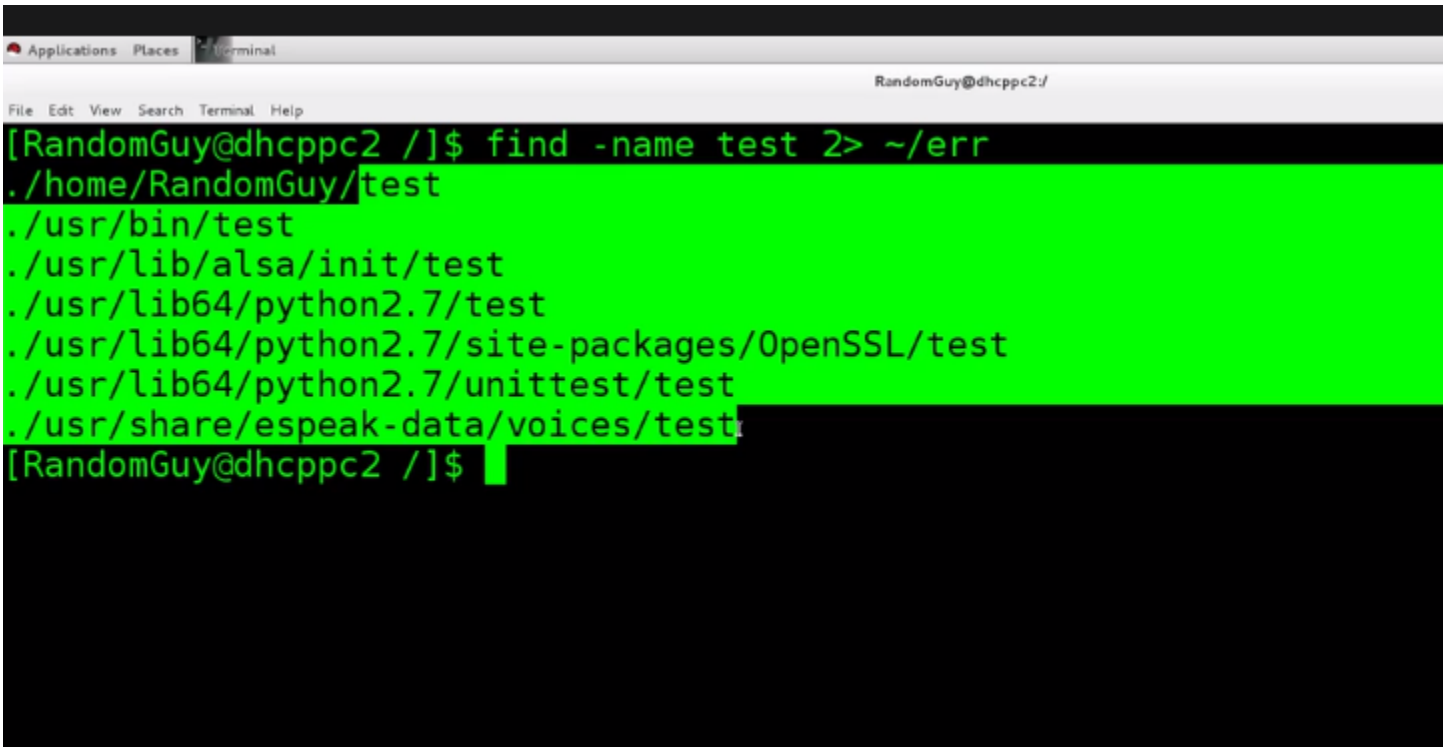
- To reverse those number "103 --> 301" we pass the stdout of this command to rev
ps -ef | awk -F" " '{print$2}' | rev

- To store the result and simultaneously print :
ps -ef | awk -F" " '{print$2}' | rev > nums **&&** cat nums

- Suppose you wanted to find a file in your system :

You use command :
find -name <filename..... stdout will be give tons of error messages as shown below. Only the directory with the correct file will prompt "something useful" rest all will be error messages. So you just want the location of file and ignore unwanted error messages, in such scenario we can stdout err to some file.

find -name mode.txt 2> ~/err



Important thing here is always do 'redirection of standard error last" in any case :
e.g. :
find -name mode.txt > ~/found 2> ~/err