Operating Systems

Lab 03: Files and Directories

Content

Content
Creating Files
Wildcards
Manipulating Files and Directories

Creating Files and Directories

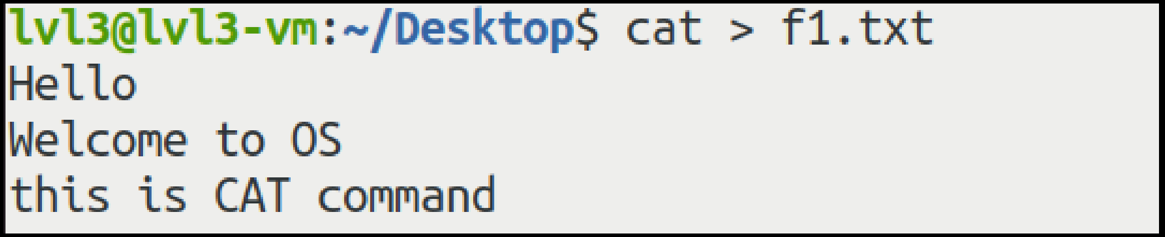
- Linux file system considers everything as a file in Linux
    - text files, images, partitions, compiled programs, directories, or hardware devices.
- If it is not a file, then it must be a process.
- Linux files are case sensitive, so $test.txt$ and $Test.txt$ will be considered as two different files.

Creating Files and Directories

- We can create files using the following methods:
  - using cat command
  - using $touch$ command
  - using redirect '$>$' symbol
  - using $echo$ command
  - using $printf$ command
  - using a different text editor like $vim, nano, vi$
  - from the desktop file manager

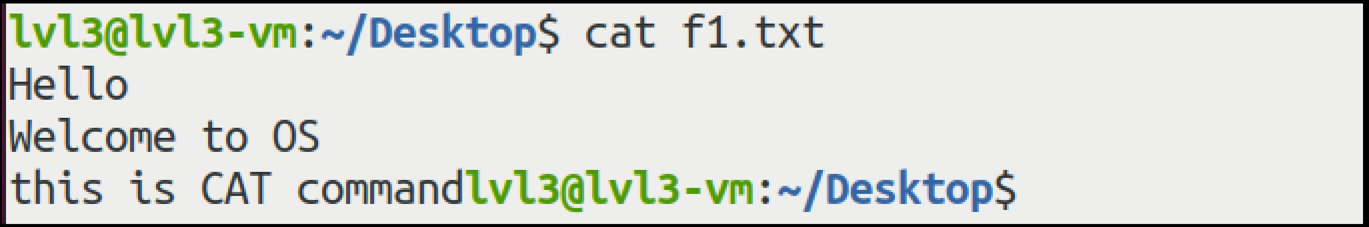
cat Command

- It is used to
  - create a file,
  - display the content of the file,
  - concatenate the contents of multiple files,
- Create a text file in the editor mode



```
lvl3@lvl3-vm:~/Desktop$ cat > f1.txt
Hello
Welcome to OS
this is CAT command
```

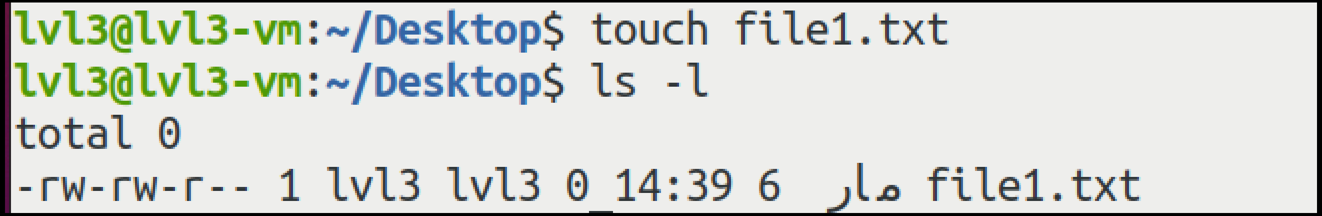
  - Press CTRL+ C to save and exit the file with a line break.
  - Press CTRL+ D to save and exit the file without line break.
- Use the cat command to display the contents of a file
  - Remove the ‘>’ symbol



```
lvl3@lvl3-vm:~/Desktop$ cat f1.txt
Hello
Welcome to OS
this is CAT commandlvl3@lvl3-vm:~/Desktop$
```

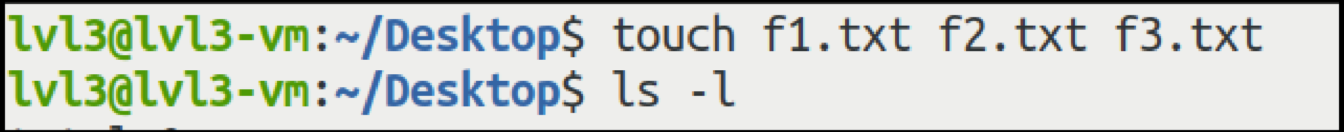
touch Command

- It is used to
  - create a new empty file,



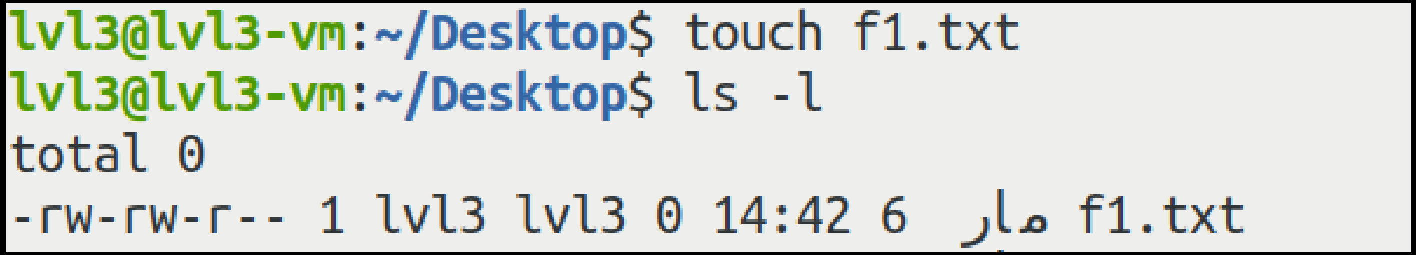
```
lvl3@lvl3-vm:~/Desktop$ touch file1.txt
lvl3@lvl3-vm:~/Desktop$ ls -l
total 0
-rw-rw-r-- 1 lvl3 lvl3 0 14:39 6  مار file1.txt
```

  - create multiple files by executing this command at once.



```
lvl3@lvl3-vm:~/Desktop$ touch f1.txt f2.txt f3.txt
lvl3@lvl3-vm:~/Desktop$ ls -l
```

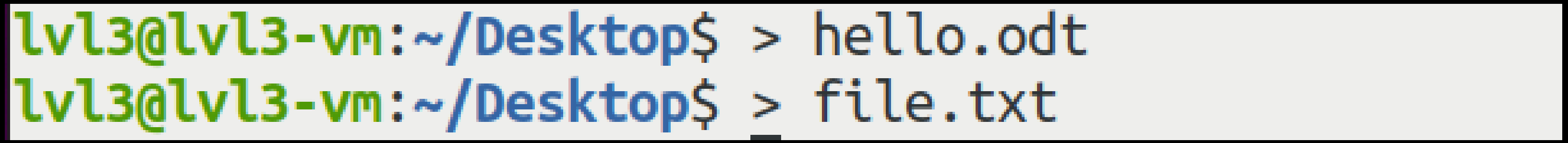
  - update the time stamp on existing files and directories,
    - Write the existing file name



```
lvl3@lvl3-vm:~/Desktop$ touch f1.txt
lvl3@lvl3-vm:~/Desktop$ ls -l
total 0
-rw-rw-r-- 1 lvl3 lvl3 0 14:42 6  مار f1.txt
```

Redirect > Symbol

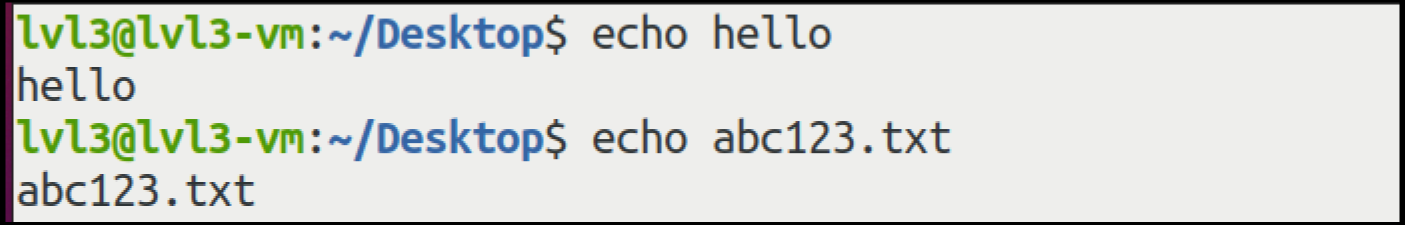
- This symbol is mostly used to redirect the output.
- The same as *touch* command.



```
lvl3@lvl3-vm:~/Desktop$ > hello.odt
lvl3@lvl3-vm:~/Desktop$ > file.txt
```

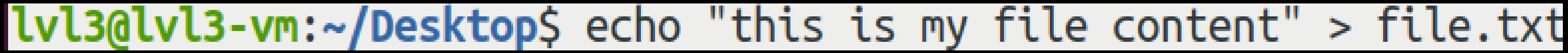
echo Symbol

- *echo* command is used to display line of text/string that are passed as an argument.
  - Used in shell scripts and batch files to output status text to the screen or a file.



```
lvl3@lvl3-vm:~/Desktop$ echo hello
hello
lvl3@lvl3-vm:~/Desktop$ echo abc123.txt
abc123.txt
```

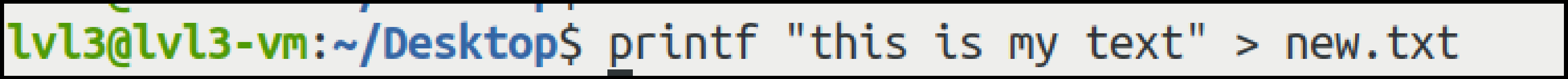
- Create a file using *echo* command.
  - Use > operator



```
lvl3@lvl3-vm:~/Desktop$ echo "this is my file content" > file.txt
```

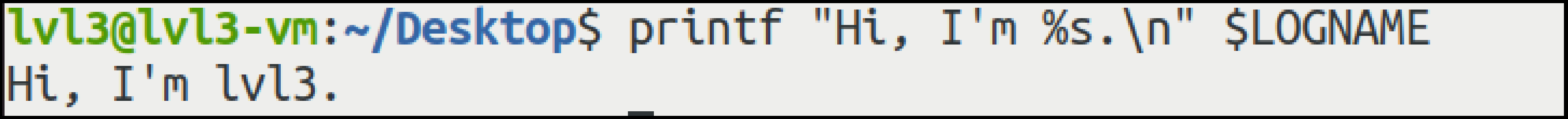
printf Command

- We can also create a file using *printf* command.
  - Like *echo* command, use the > operator



```
lvl3@lvl3-vm:~/Desktop$ printf "this is my text" > new.txt
```

- Use *printf* command to format strings as in C programming language.

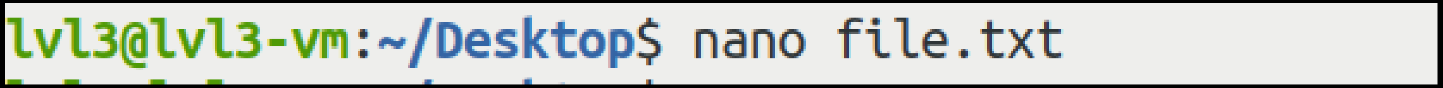


```
lvl3@lvl3-vm:~/Desktop$ printf "Hi, I'm %s.\n" $LOGNAME
Hi, I'm lvl3.
```

  - %s is a format specifier to be replaced by a string variable ($LOGNAME)
  - \n is a format specifier to print new line.
  - $LOGNAME is an environment variable, which is the username of whoever ran the command.

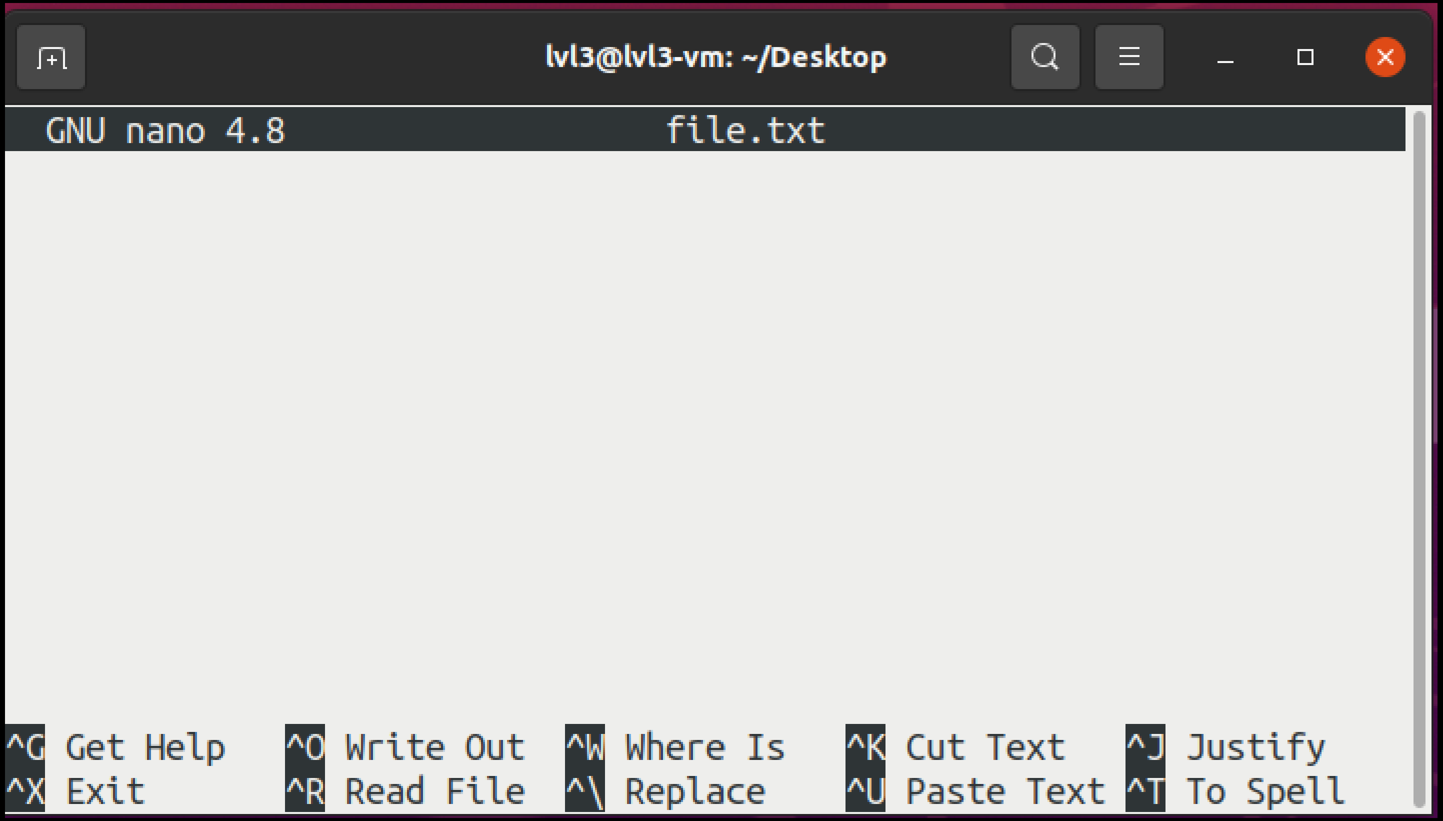
nano Editor

- *nano* is a text editor in Linux.



```
lvl3@lvl3-vm:~/Desktop$ nano file.txt
```

  - ○ CTRL + O saves the file



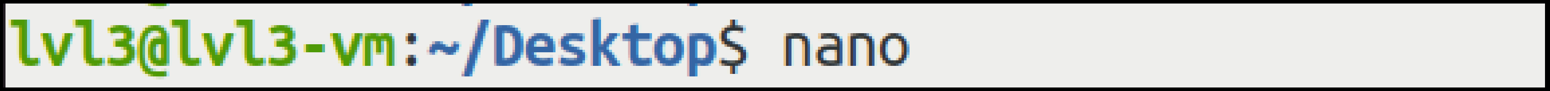
```
  GNU nano 4.8                  file.txt




^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell
```

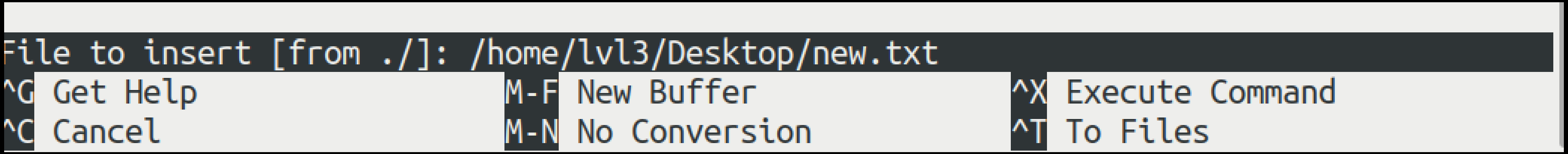
nano Editor

- We can open new file in *nano* editor given the file’s path
  1. Enter *nano* in the shell



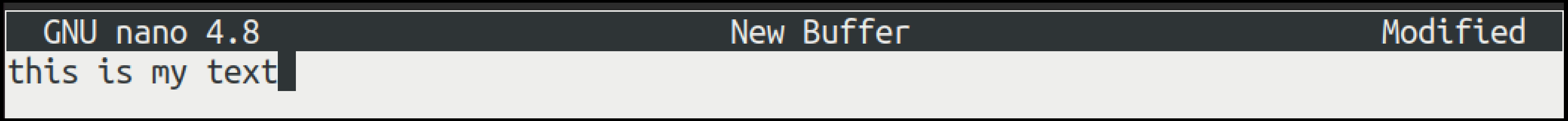
```
lvl3@lvl3-vm:~/Desktop$ nano
```

  2. Press CTRL + R
  3. Enter the path of the file



```
File to insert [from ./]: /home/lvl3/Desktop/new.txt
^G Get Help              M-F New Buffer              ^X Execute Command
^C Cancel                M-N No Conversion           ^T To Files
```

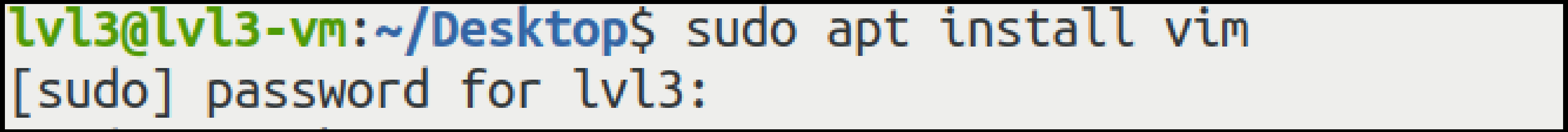
  4. Press Enter



```
  GNU nano 4.8                     New Buffer                          Modified
this is my text
```

vim and vi Editor

- vim is a text editor that is not built-in Linux by default.
- vi is a text editor very similar to vim, installed by default.
- To install vim

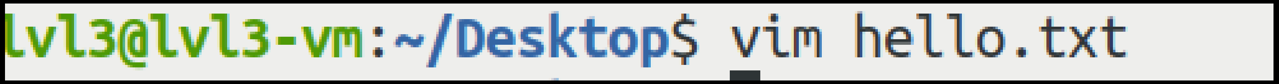


```
lvl3@lvl3-vm:~/Desktop$ sudo apt install vim
[sudo] password for lvl3:
```

  - $sudo$ is a command to perform operations that require root privileges.
  - apt is the Advanced Package Tool that is used to install, update, and remove applications in Linux.
  - $install$ indicates the installation of a package.
  - vim is the application we want to install.

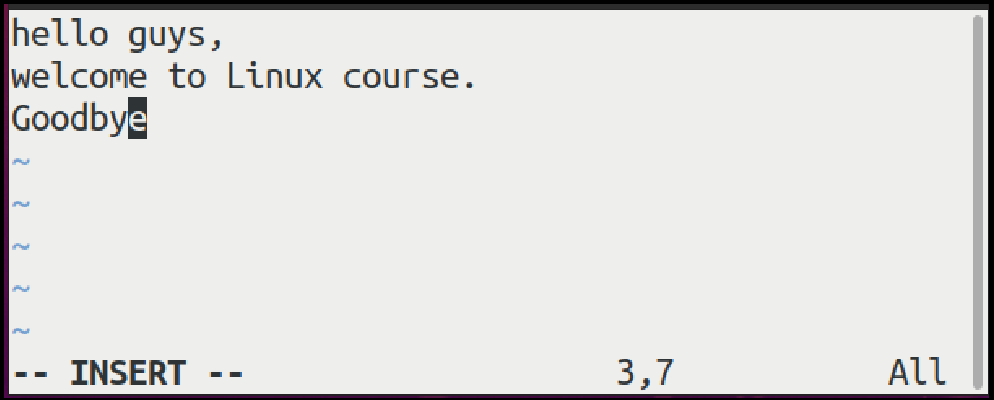
vim and *vi* Editor

- To create a file using *vim*
  1. Enter the command



```
lvl3@lvl3-vm:~/Desktop$ vim hello.txt
```

  2. Press *i* to enter insertion mode



```
hello guys,
welcome to Linux course.
Goodbye
~
~
~
~
~
-- INSERT --                    3,7            All
```

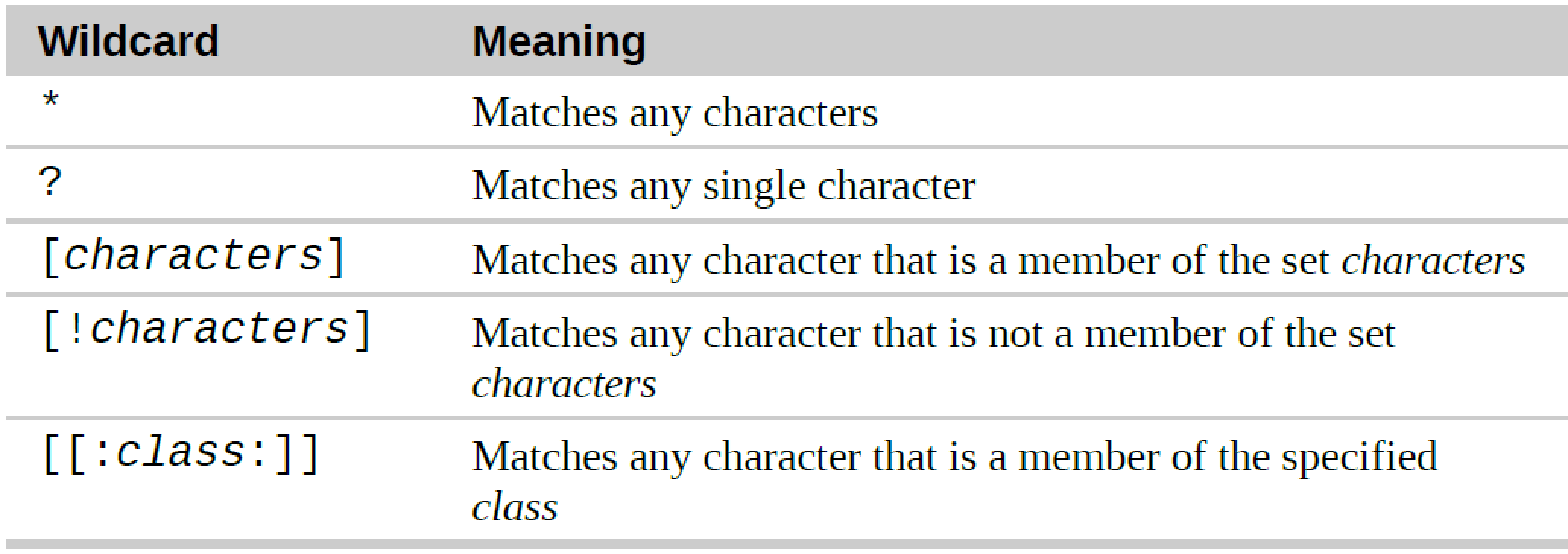
  3. Press ESC to exit insertion mode.
  4. Enter :*wq* to save and exit the file

Content

Content
Creating Files
Wildcards
Manipulating Files and Directories

Wildcards

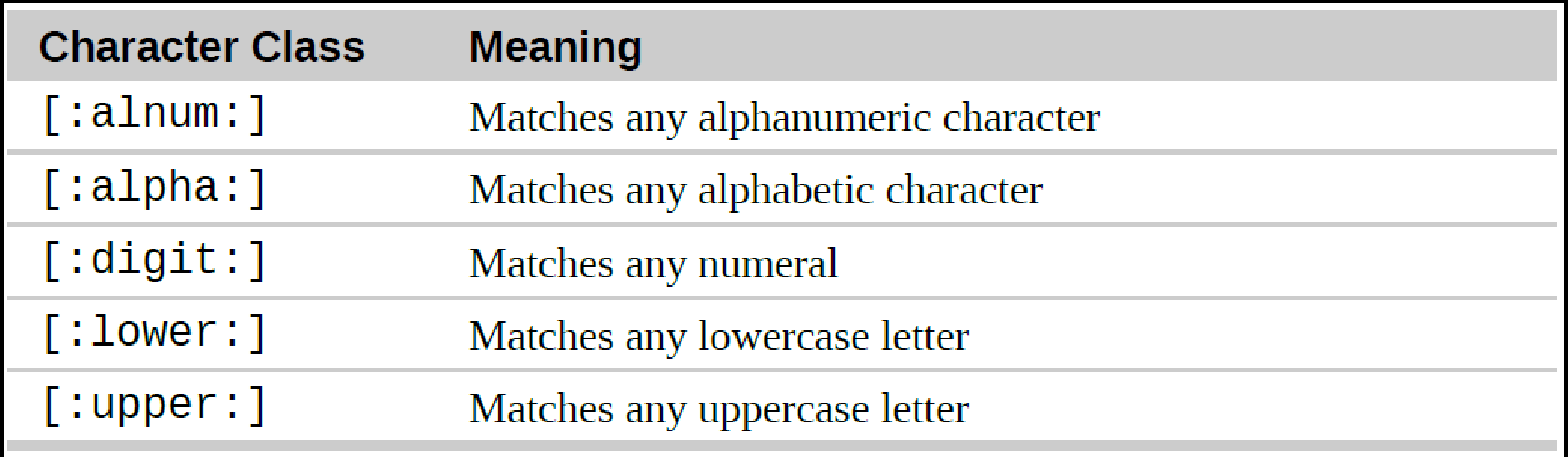
- Wildcards (globbing) are special characters to help us rapidly specify groups of filenames.
  - allows us to select filenames based on patterns of characters.
  - Wildcards can be used with any command that accepts filenames as arguments



Wildcard	Meaning
`*`	Matches any characters
`?`	Matches any single character
`[`*`characters`*`]`	Matches any character that is a member of the set *characters*
`[!`*`characters`*`]`	Matches any character that is not a member of the set *characters*
`[[:`*`class`*`:]]`	Matches any character that is a member of the specified *class*

Wildcards

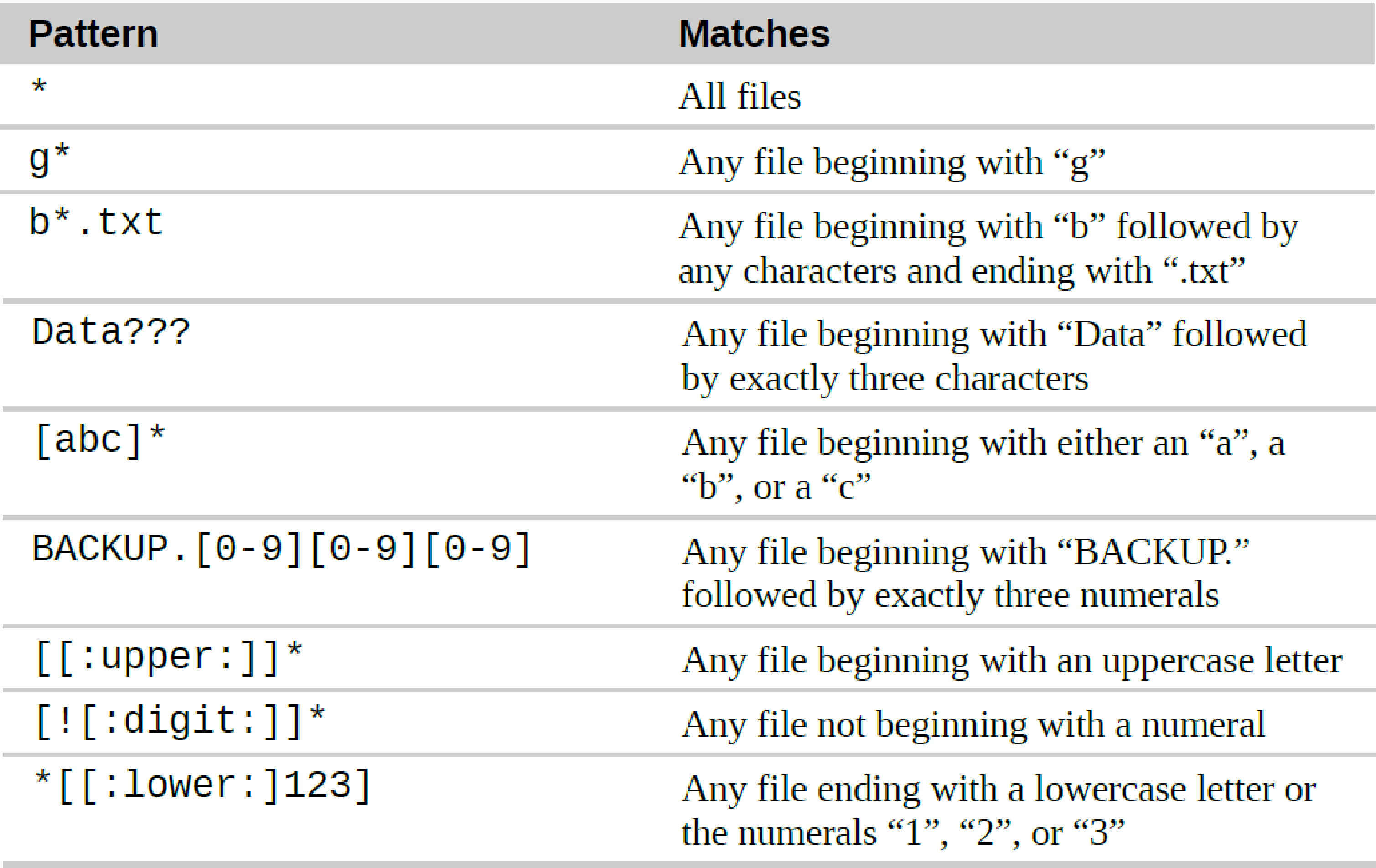
- Character classes.
  - Using wildcards makes it possible to construct sophisticated selection criteria for filenames.



Character Class	Meaning
`[:alnum:]`	Matches any alphanumeric character
`[:alpha:]`	Matches any alphabetic character
`[:digit:]`	Matches any numeral
`[:lower:]`	Matches any lowercase letter
`[:upper:]`	Matches any uppercase letter

Wildcards

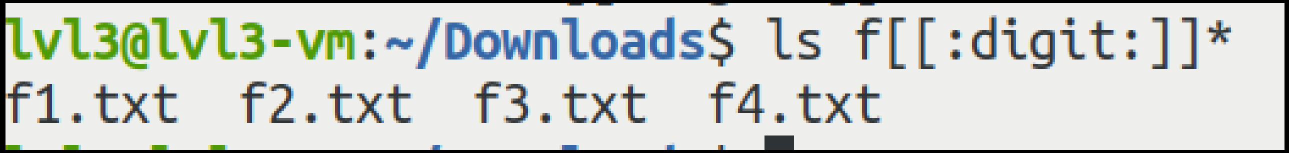
- Examples



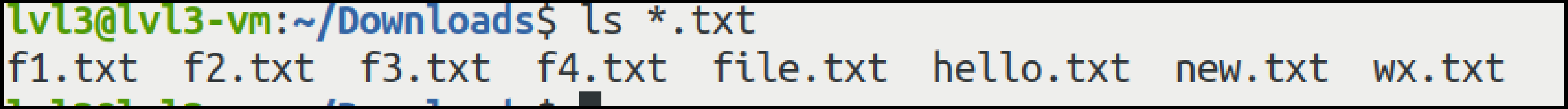
Pattern	Matches
`*`	All files
`g*`	Any file beginning with “g”
`b*.txt`	Any file beginning with “b” followed by any characters and ending with “.txt”
`Data???`	Any file beginning with “Data” followed by exactly three characters
`[abc]*`	Any file beginning with either an “a”, a “b”, or a “c”
`BACKUP.[0-9][0-9][0-9]`	Any file beginning with “BACKUP.” followed by exactly three numerals
`[[:upper:]]*`	Any file beginning with an uppercase letter
`[![:digit:]]*`	Any file not beginning with a numeral
`*[[:lower:]123]`	Any file ending with a lowercase letter or the numerals “1”, “2”, or “3”

Wildcards

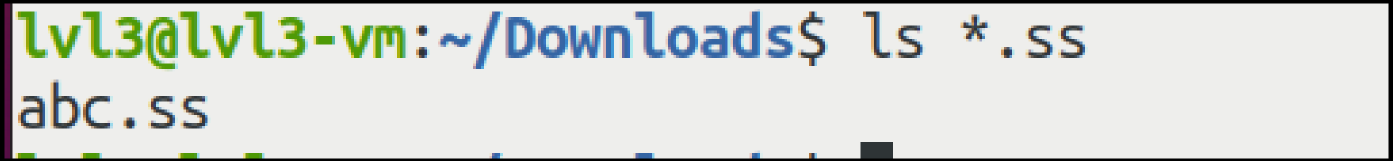
- Examples



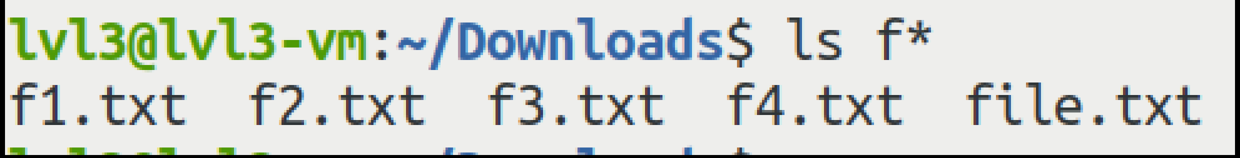
```
lvl3@lvl3-vm:~/Downloads$ ls f[[:digit:]]*
f1.txt  f2.txt  f3.txt  f4.txt
```



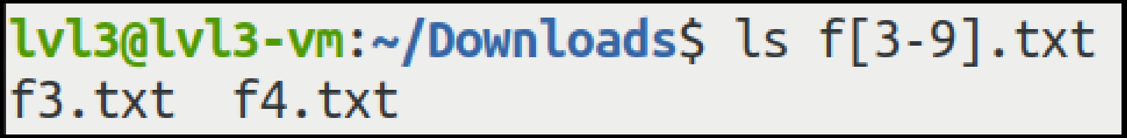
```
lvl3@lvl3-vm:~/Downloads$ ls *.txt
f1.txt  f2.txt  f3.txt  f4.txt  file.txt  hello.txt  new.txt  wx.txt
```



```
lvl3@lvl3-vm:~/Downloads$ ls *.ss
abc.ss
```



```
lvl3@lvl3-vm:~/Downloads$ ls f*
f1.txt  f2.txt  f3.txt  f4.txt  file.txt
```



```
lvl3@lvl3-vm:~/Downloads$ ls f[3-9].txt
f3.txt  f4.txt
```

Content

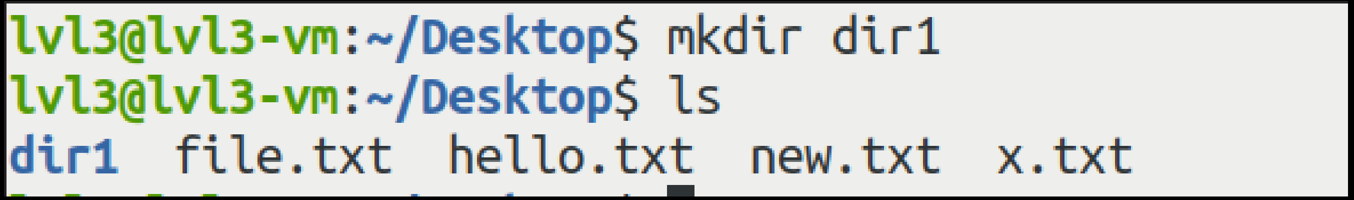
Content
Creating Files
Wildcards
Manipulating Files and Directories

Manipulating Files and Directories

- Commands
  - cp – Copy files and directories
  - mv – Move/rename files and directories
  - $mkdir$ – Create directories
  - rm – Remove files and directories

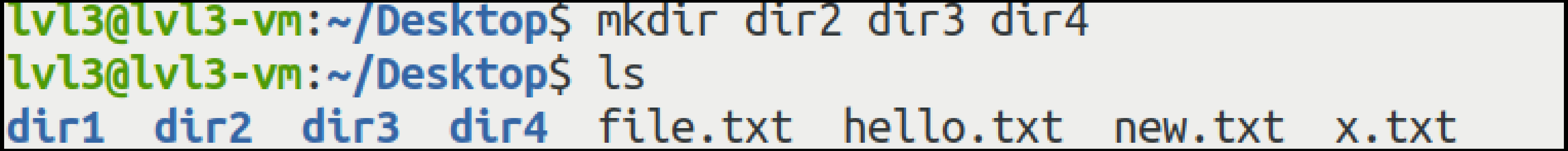
mkdir Command

- The $mkdir$ command is used to create directories.



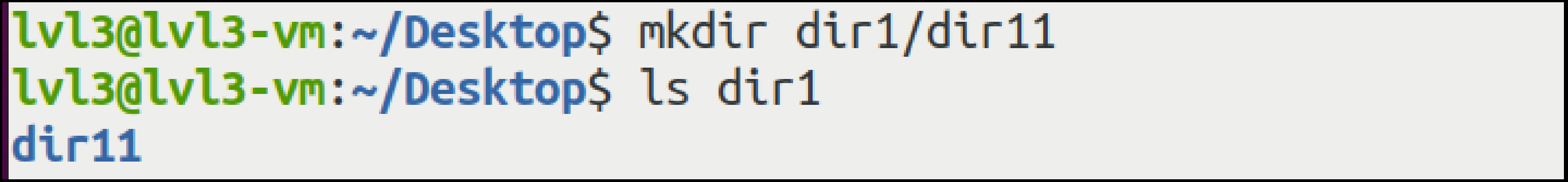
```
lvl3@lvl3-vm:~/Desktop$ mkdir dir1
lvl3@lvl3-vm:~/Desktop$ ls
dir1  file.txt  hello.txt  new.txt  x.txt
```

- We can create multiple directories in one line



```
lvl3@lvl3-vm:~/Desktop$ mkdir dir2 dir3 dir4
lvl3@lvl3-vm:~/Desktop$ ls
dir1  dir2  dir3  dir4  file.txt  hello.txt  new.txt  x.txt
```

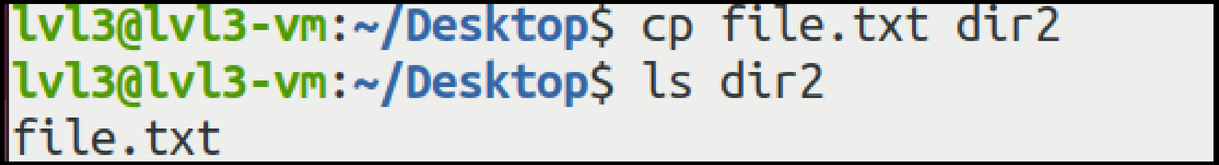
- We can create a directory in another directory



```
lvl3@lvl3-vm:~/Desktop$ mkdir dir1/dir11
lvl3@lvl3-vm:~/Desktop$ ls dir1
dir11
```

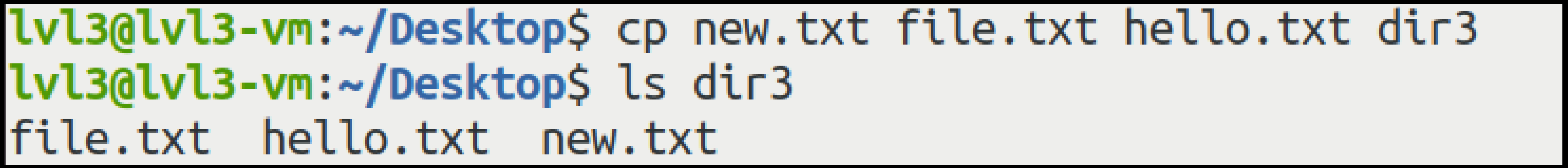
cp Command

- The *cp* command copies files or directories. It can be used two different ways.
  - copies the single file or directory.



```
lvl3@lvl3-vm:~/Desktop$ cp file.txt dir2
lvl3@lvl3-vm:~/Desktop$ ls dir2
file.txt
```

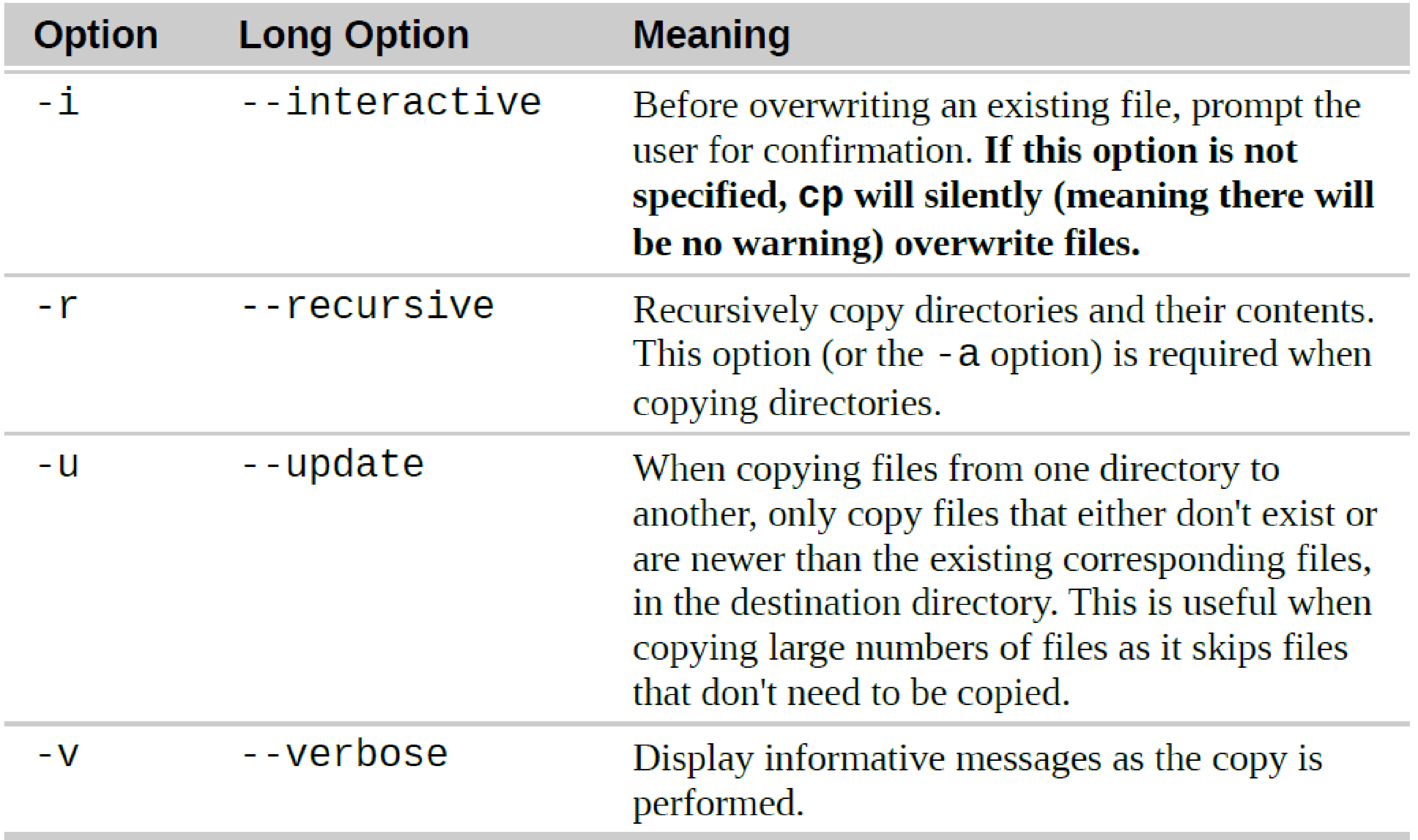
  - copies multiple items (either files or directories) into a directory.



```
lvl3@lvl3-vm:~/Desktop$ cp new.txt file.txt hello.txt dir3
lvl3@lvl3-vm:~/Desktop$ ls dir3
file.txt  hello.txt  new.txt
```

cp Command

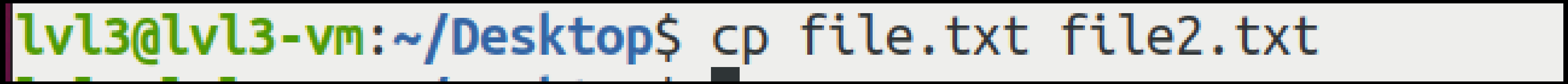
- Options and arguments



Option	Long Option	Meaning
`-i`	`--interactive`	Before overwriting an existing file, prompt the user for confirmation. **If this option is not specified, `cp` will silently (meaning there will be no warning) overwrite files.**
`-r`	`--recursive`	Recursively copy directories and their contents. This option (or the `-a` option) is required when copying directories.
`-u`	`--update`	When copying files from one directory to another, only copy files that either don't exist or are newer than the existing corresponding files, in the destination directory. This is useful when copying large numbers of files as it skips files that don't need to be copied.
`-v`	`--verbose`	Display informative messages as the copy is performed.

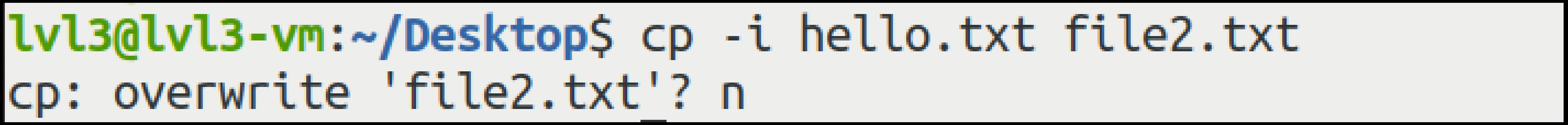
cp Command

- Examples
  - If $file2.txt$ exists, it is overwritten with the contents of $file.txt$.
  - If $file2.txt$ does not exist, it is created.

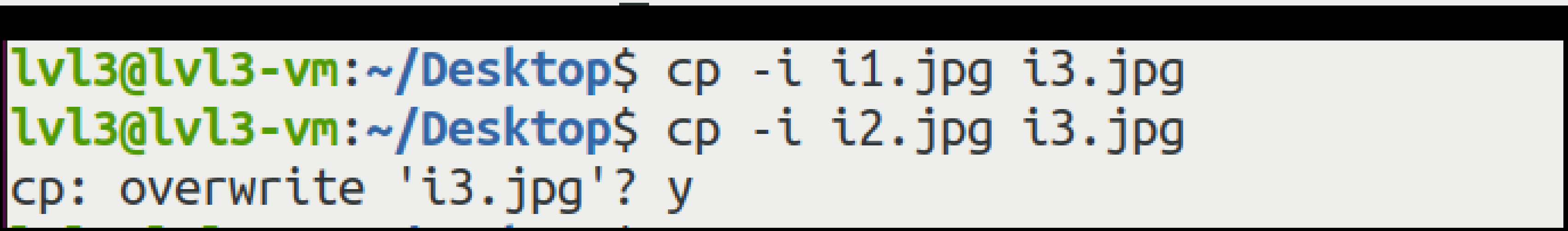


```
lvl3@lvl3-vm:~/Desktop$ cp file.txt file2.txt
```

  - Use $-i$ option to ask before overwrite



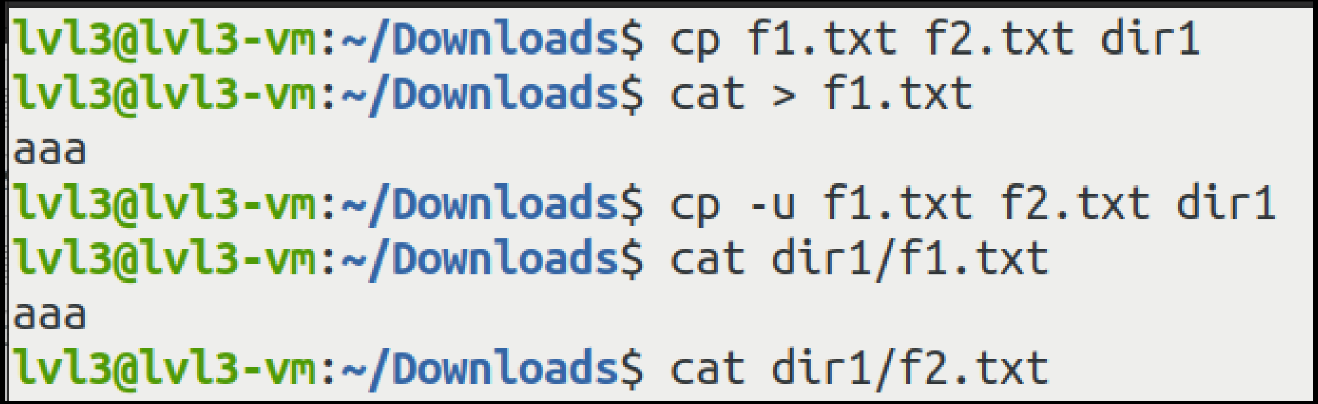
```
lvl3@lvl3-vm:~/Desktop$ cp -i hello.txt file2.txt
cp: overwrite 'file2.txt'? n
```



```
lvl3@lvl3-vm:~/Desktop$ cp -i i1.jpg i3.jpg
lvl3@lvl3-vm:~/Desktop$ cp -i i2.jpg i3.jpg
cp: overwrite 'i3.jpg'? y
```

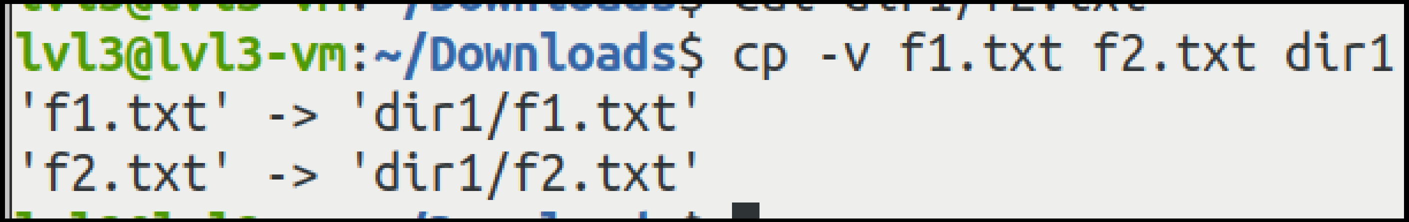
cp Command

- Examples
  - Copy updated files only



```
lvl3@lvl3-vm:~/Downloads$ cp f1.txt f2.txt dir1
lvl3@lvl3-vm:~/Downloads$ cat > f1.txt
aaa
lvl3@lvl3-vm:~/Downloads$ cp -u f1.txt f2.txt dir1
lvl3@lvl3-vm:~/Downloads$ cat dir1/f1.txt
aaa
lvl3@lvl3-vm:~/Downloads$ cat dir1/f2.txt
```

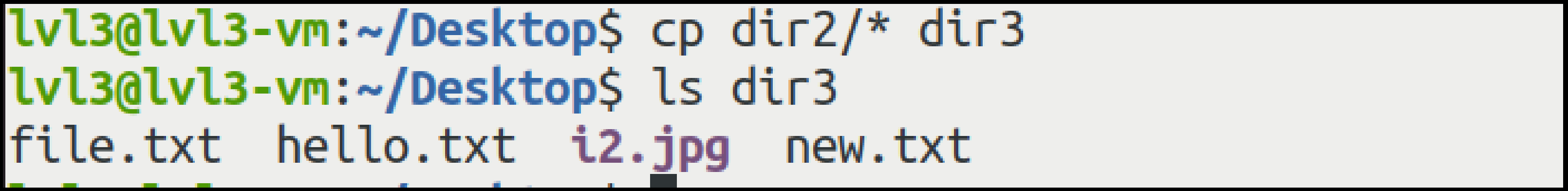
  - Show informative messages during copy



```
lvl3@lvl3-vm:~/Downloads$ cat dir1/f2.txt
lvl3@lvl3-vm:~/Downloads$ cp -v f1.txt f2.txt dir1
'f1.txt' -> 'dir1/f1.txt'
'f2.txt' -> 'dir1/f2.txt'
```

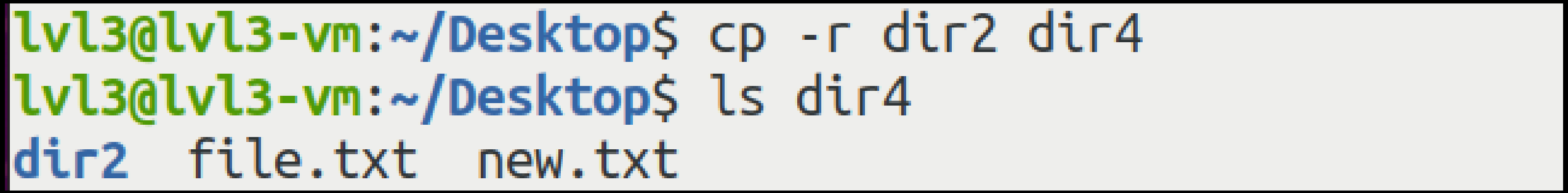
cp Command

- Examples
  - Using a wildcard, copy all the files in $dir2$ into $dir3$. The directory $dir3$ must already exist.



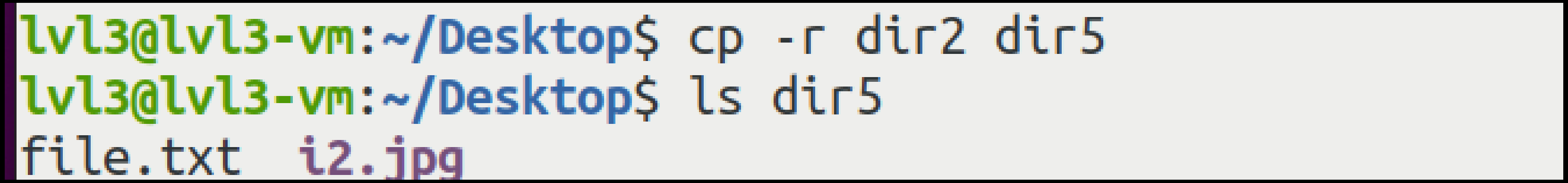
```
lvl3@lvl3-vm:~/Desktop$ cp dir2/* dir3
lvl3@lvl3-vm:~/Desktop$ ls dir3
file.txt  hello.txt  i2.jpg  new.txt
```

  - Use $-r$ option to recursively copy all files and directories



```
lvl3@lvl3-vm:~/Desktop$ cp -r dir2 dir4
lvl3@lvl3-vm:~/Desktop$ ls dir4
dir2  file.txt  new.txt
```

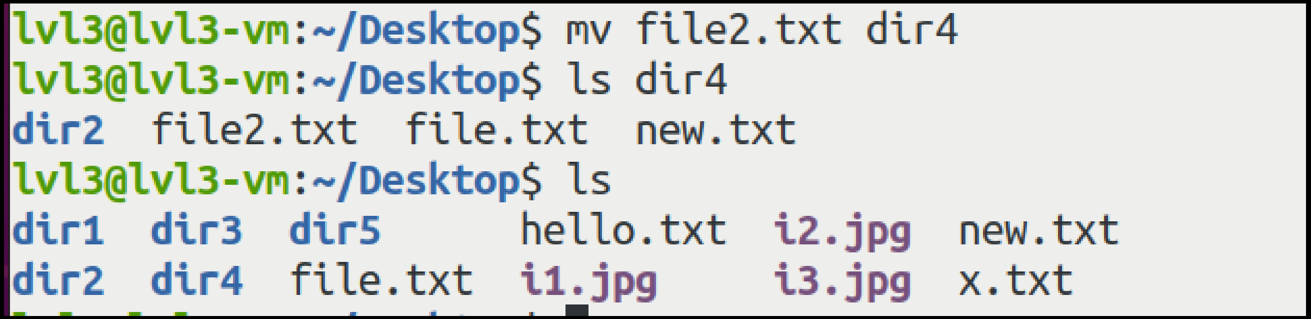
  - When using $-r$, if $dir5$ does not exist, it is created, and the content is copied



```
lvl3@lvl3-vm:~/Desktop$ cp -r dir2 dir5
lvl3@lvl3-vm:~/Desktop$ ls dir5
file.txt  i2.jpg
```

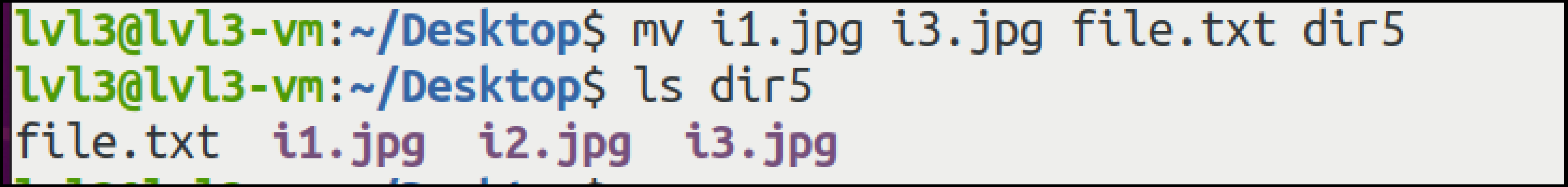
mv Command

- The mv command performs both file moving and file renaming.
  - In either case, the original filename no longer exists after the operation.
- Move (cut) a file into a directory



```
lvl3@lvl3-vm:~/Desktop$ mv file2.txt dir4
lvl3@lvl3-vm:~/Desktop$ ls dir4
dir2  file2.txt  file.txt  new.txt
lvl3@lvl3-vm:~/Desktop$ ls
dir1  dir3  dir5      hello.txt  i2.jpg  new.txt
dir2  dir4  file.txt  i1.jpg     i3.jpg  x.txt
```

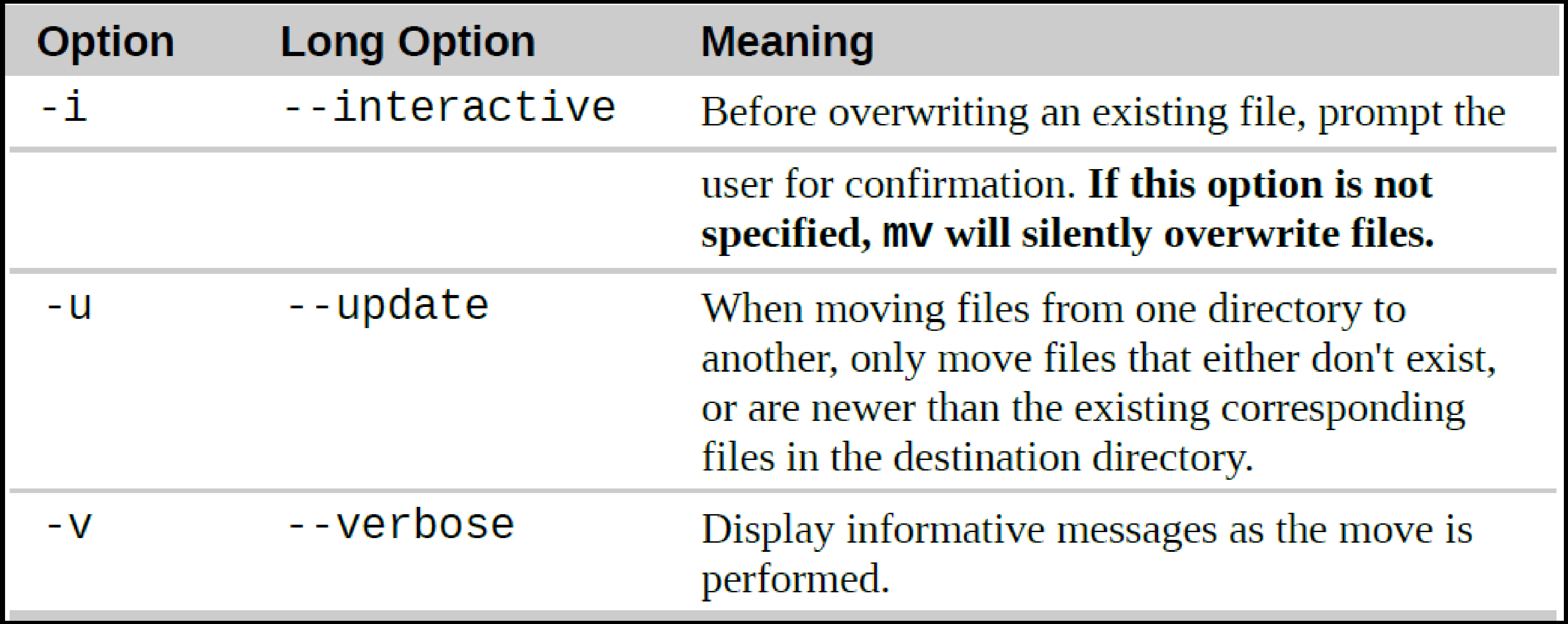
- Move multiple files into another directory



```
lvl3@lvl3-vm:~/Desktop$ mv i1.jpg i3.jpg file.txt dir5
lvl3@lvl3-vm:~/Desktop$ ls dir5
file.txt  i1.jpg  i2.jpg  i3.jpg
```

mv Command

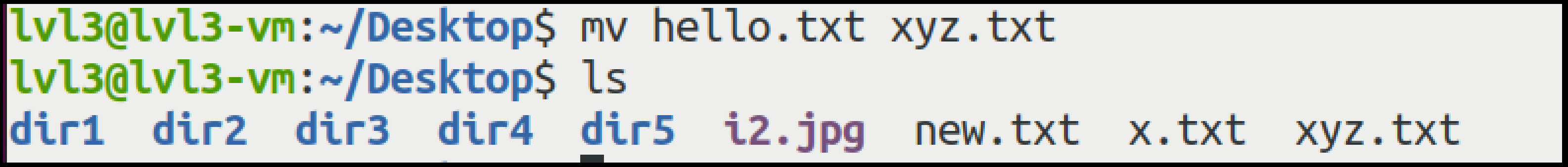
- Options and arguments



Option	Long Option	Meaning
`-i`	`--interactive`	Before overwriting an existing file, prompt the
		user for confirmation. **If this option is not specified, `mv` will silently overwrite files.**
`-u`	`--update`	When moving files from one directory to another, only move files that either don't exist, or are newer than the existing corresponding files in the destination directory.
`-v`	`--verbose`	Display informative messages as the move is performed.

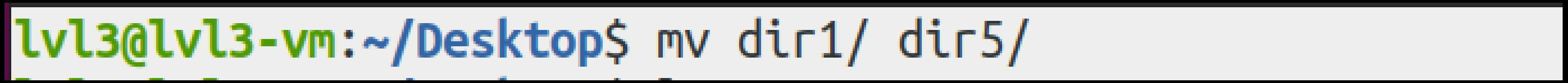
mv Command

- Examples
  - ○ Rename a file

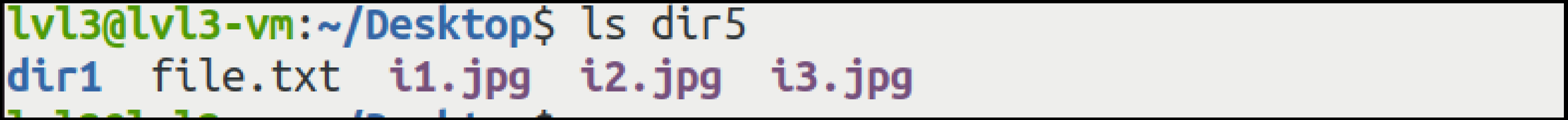


```
lvl3@lvl3-vm:~/Desktop$ mv hello.txt xyz.txt
lvl3@lvl3-vm:~/Desktop$ ls
dir1  dir2  dir3  dir4  dir5  i2.jpg  new.txt  x.txt  xyz.txt
```

  - ○ Remove a directory into another directory

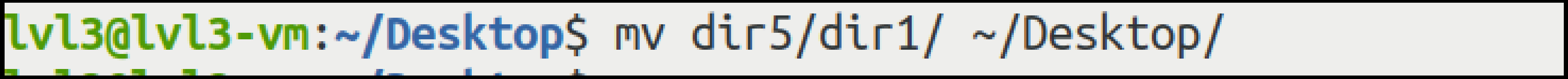


```
lvl3@lvl3-vm:~/Desktop$ mv dir1/ dir5/
```



```
lvl3@lvl3-vm:~/Desktop$ ls dir5
dir1  file.txt  i1.jpg  i2.jpg  i3.jpg
```

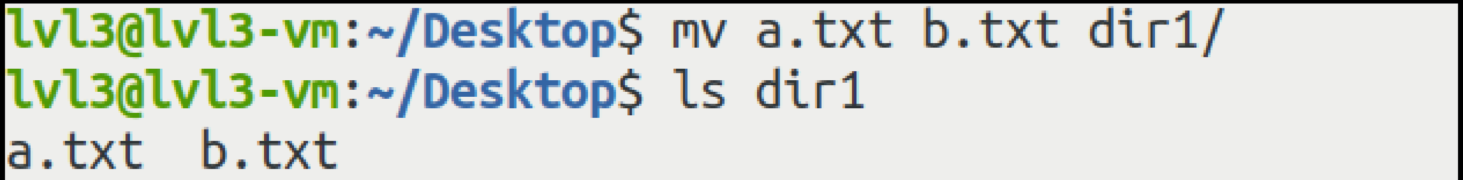
  - ○ Return $dir1$ to the desktop again



```
lvl3@lvl3-vm:~/Desktop$ mv dir5/dir1/ ~/Desktop/
```

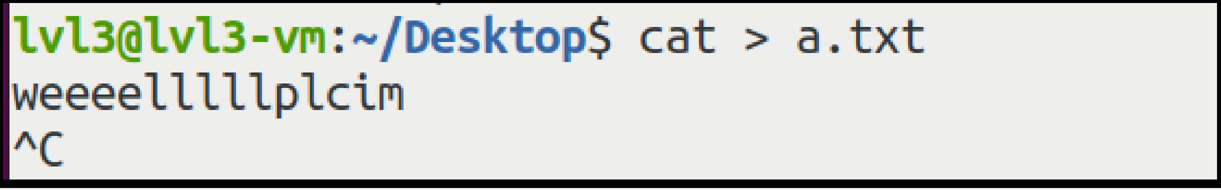
mv Command

- Examples
  - Move multiple files

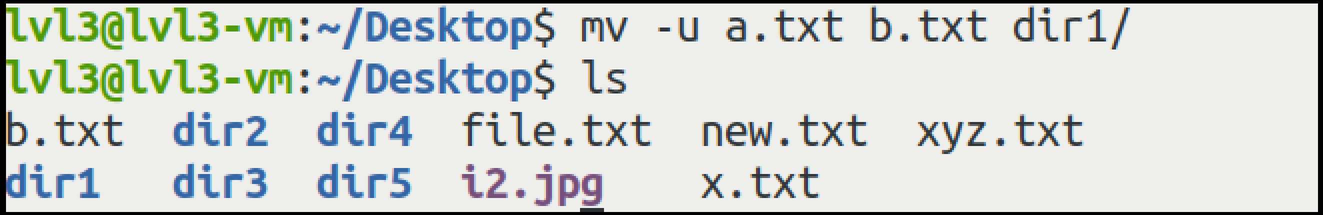


```
lvl3@lvl3-vm:~/Desktop$ mv a.txt b.txt dir1/
lvl3@lvl3-vm:~/Desktop$ ls dir1
a.txt  b.txt
```

  - Use $-u$ option to only move files that either don't exist or are newer than the existing corresponding files in the destination directory.



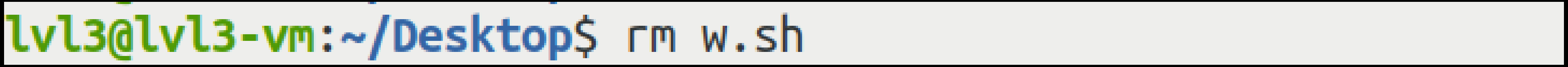
```
lvl3@lvl3-vm:~/Desktop$ cat > a.txt
weeeelllllplcim
^C
```



```
lvl3@lvl3-vm:~/Desktop$ mv -u a.txt b.txt dir1/
lvl3@lvl3-vm:~/Desktop$ ls
b.txt  dir2  dir4  file.txt  new.txt  xyz.txt
dir1   dir3  dir5  i2.jpg    x.txt
```

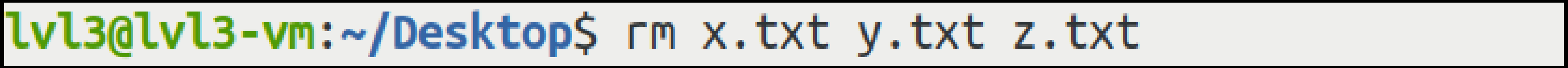
rm Command

- The rm command is used to remove (delete) files and directories.
  - Remove one file



```
lvl3@lvl3-vm:~/Desktop$ rm w.sh
```

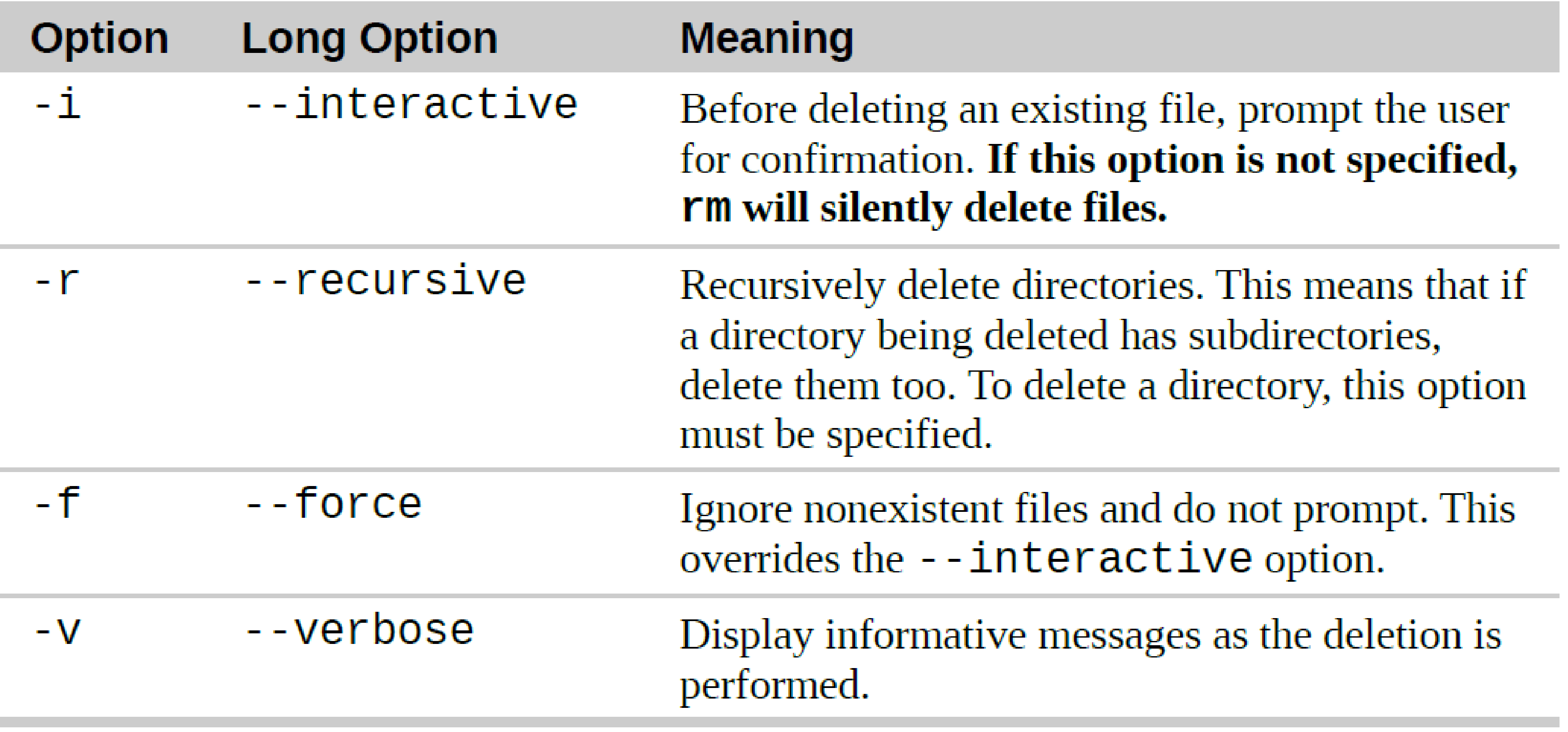
  - Remove multiple files



```
lvl3@lvl3-vm:~/Desktop$ rm x.txt y.txt z.txt
```

rm Command

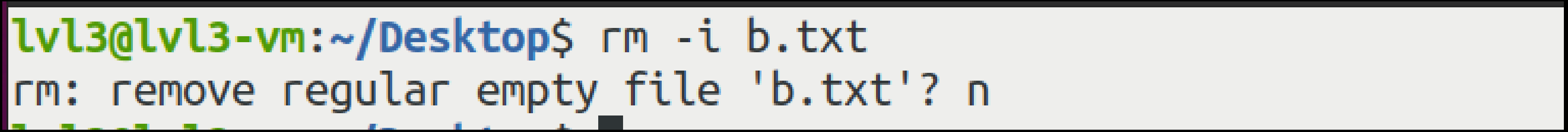
- Options and arguments



Option	Long Option	Meaning
`-i`	`--interactive`	Before deleting an existing file, prompt the user for confirmation. **If this option is not specified, `rm` will silently delete files.**
`-r`	`--recursive`	Recursively delete directories. This means that if a directory being deleted has subdirectories, delete them too. To delete a directory, this option must be specified.
`-f`	`--force`	Ignore nonexistent files and do not prompt. This overrides the `--interactive` option.
`-v`	`--verbose`	Display informative messages as the deletion is performed.

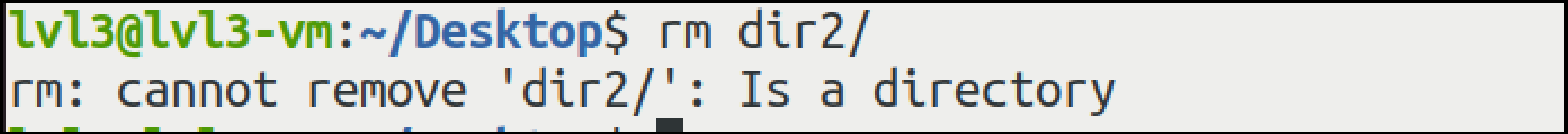
rm Command

- Examples
  - Use $-i$ to ask before deletion.



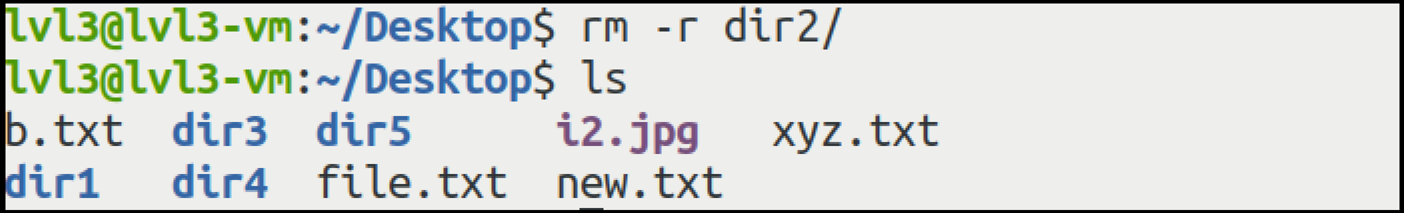
```
lvl3@lvl3-vm:~/Desktop$ rm -i b.txt
rm: remove regular empty file 'b.txt'? n
```

  - Remove a directory with its content
    - cannot remove a directory without $-r$



```
lvl3@lvl3-vm:~/Desktop$ rm dir2/
rm: cannot remove 'dir2/': Is a directory
```

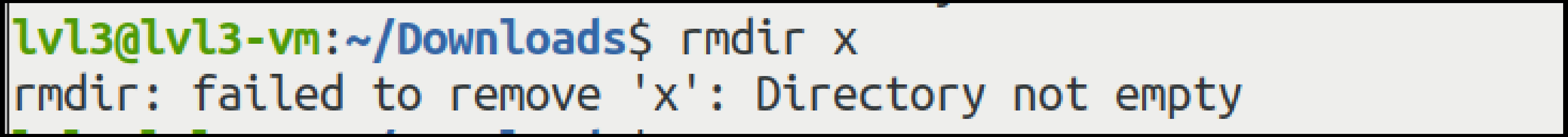
    - Use $-r$ option to remove a directory



```
lvl3@lvl3-vm:~/Desktop$ rm -r dir2/
lvl3@lvl3-vm:~/Desktop$ ls
b.txt  dir3  dir5      i2.jpg   xyz.txt
dir1   dir4  file.txt  new.txt
```

$rmdir$ Command

- $rmdir$ is the same as rm, used for deleting empty directories only.



```
lvl3@lvl3-vm:~/Downloads$ rmdir x
rmdir: failed to remove 'x': Directory not empty
```

Exercise

- Create a new directory $mydir$ in the $Documents$ directory.
- Inside $mydir$, create a text file $rand.txt$.
- Write anything in the file.
- Copy $rand.txt$ six times, $rand1.txt$ to $rand6.txt$ **.
- Change the content of $rand2.txt$.
- Create another directory xyz in the $Desktop$.
- Copy the content of $mydir$ to xyz.
- Delete $mydir$.

Exercise

- Create a new directory $mydir$ in the $Documents$ directory.
  - mkdir ~/Documents/mydir
- Inside $mydir$, create a text file $rand.txt$ with a random content.
  - cd ~/Documents/mydir
  - cat > rand.txt
- Copy $rand.txt$ six times, $rand1.txt$ to $rand6.txt$ **
  - for i in {1..6}; do cp rand.txt "rand$i.txt" ; done
- Change the content of $rand2.txt$
  - vi rand2.txt , ESC , :wq
- Create another directory xyz in the $Desktop$
  - mkdir ~/Desktop/xyz
- Copy the content of $mydir$ to xyz
  - cp -r ~/Documents/mydir/*   ~/Desktop/xyz/
- Delete $mydir$
  - rm -r ~/Documents/mydir

Summary

- cat
- $touch$
- $>$ symbol
- $echo$
- $printf$
- Text editors: $vim, nano, vi$
- $sudo$ apt install
- cp
- mv
- $mkdir$
- rm
- $rmdir$
- Wildcards

TASK

- How to use cat command to display line numbers?
- How to use cat command to concatenate two files?
- What is >> operator in Linux?
- How to save a command result in file?
- How to update Linux using command line?
- What does the command “$echo\ \ast$” do?
- How to remove all jpg files in a directory in one line?