

## EEE 4482 Server Installation and Programming

### Worksheet 2b – Text editor vi

**Objective:** To familiarize with the text editor vi

**Tools:** Windows PC

**Software:** Oracle VM Virtual Box version 6.1.12  
CentOS 7

**Topics covered:**

- Use of text editor vi

**Component list:**

None

## Part 2. Text editor - vi (vim, improved)

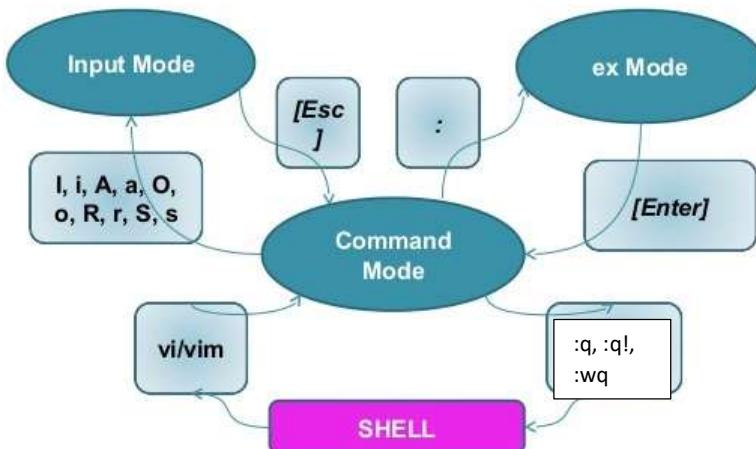
### 2.1. command mode and insert mode

The vi editor starts in **command mode**. In command mode, you can type commands. Some commands will bring you into **insert mode**. In insert mode, you can type text. The **escape key** will return you to command mode.

Table 2.1. getting to command mode

key	action
Esc	set vi(m) in command mode.

### Switching between modes of Vim



### 2.2. start typing (a A i l o O)

The difference between a A i l o and O is the location where you can start typing.

'a' appends after the current character.

'A' appends at the end of the line.

'i' inserts before the current character.

'I' inserts at the beginning of the line.

'o' puts you in a new line after the current line.

'O' puts you in a new line before the current line.

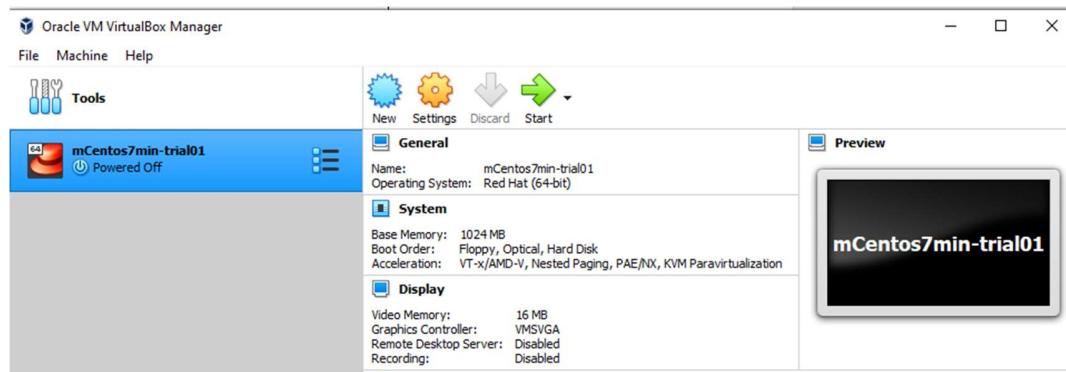
**Table 2.2. switch to insert mode**

command	action
a	start typing after the current character
A	start typing at the end of the current line
i	start typing before the current character
I	start typing at the start of the current line
o	start typing on a new line after the current line
O	start typing on a new line before the current line

## Example : Create a file “MyFile.txt” using vi editor

I want to create a text file named “MyFile.txt” using vi editor with content “Hello world”.

Run the Oracle VM VirtualBox. Select the VM that you created before and press “Start” button as shown below.



We login using login name “root” and the password (“netlab123”).

```
mCentos7min-trial01 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

CentOS Linux 7 (Core)
Kernel 3.10.0-1127.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Tue Sep  1 04:14:04 on ttys000
[root@localhost ~]#
```

Now, you are using the shell to type in command. To check what you have in the directory, in the shell, we type

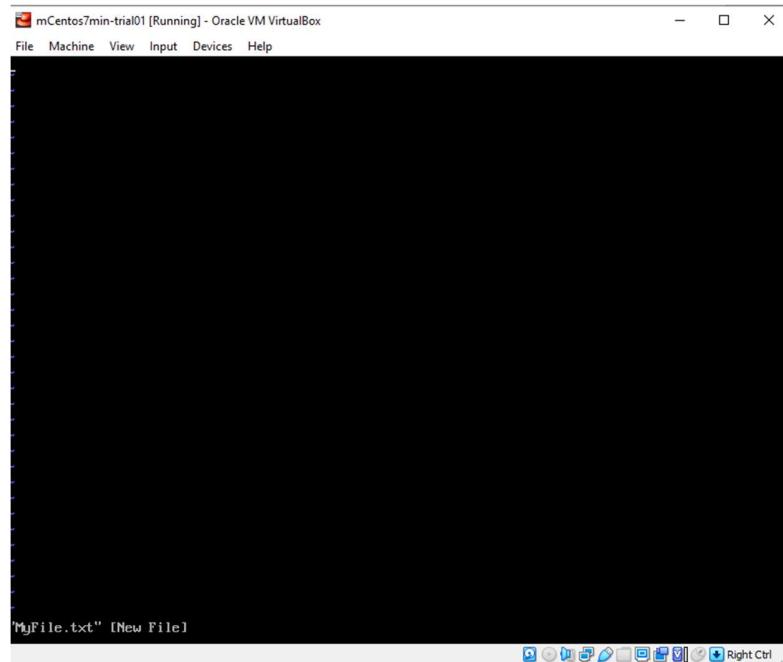
ls

```
[root@localhost ~]# ls
anaconda-ks.cfg
[root@localhost ~]# _
```

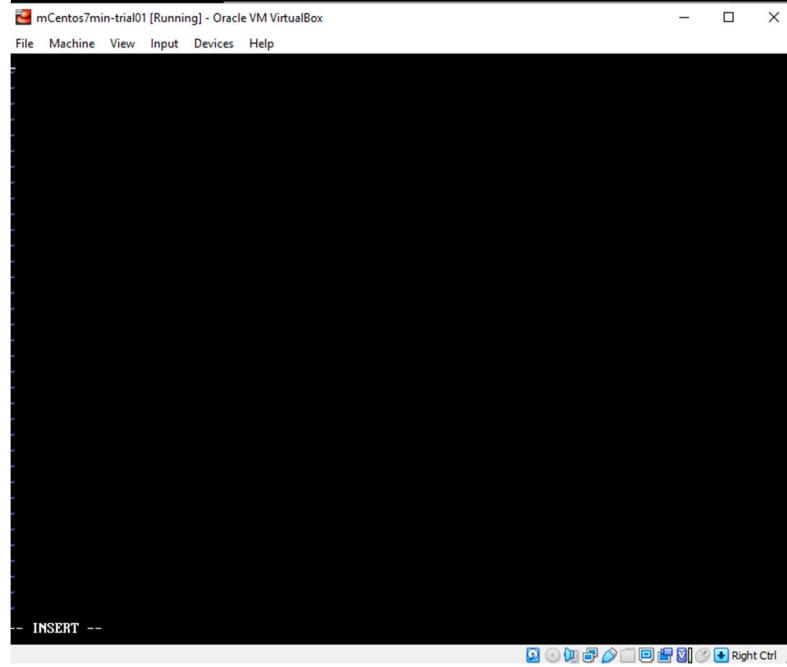
This tells us that there is only one file “anaconda-ks.cfg” in the folder.

We want to open or create a file named “MyFile.txt”. In the shell, we type

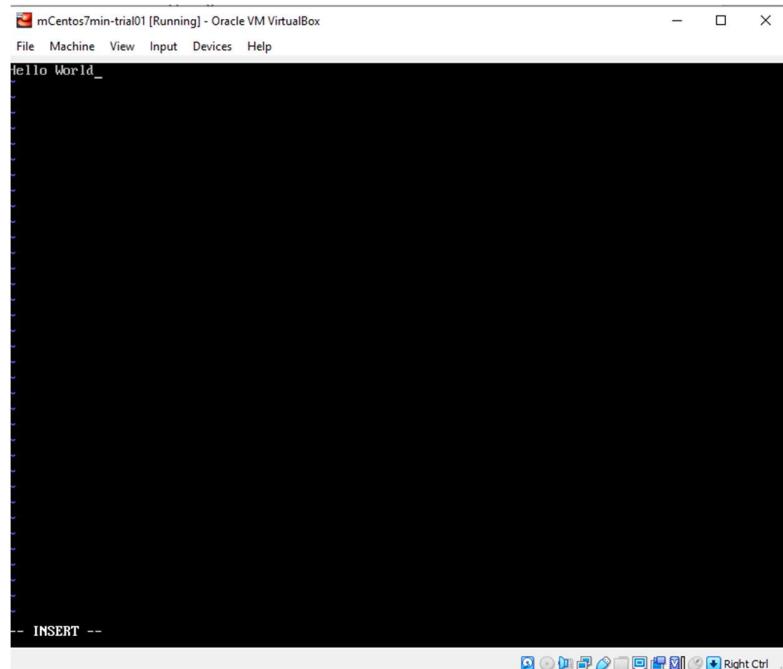
vi MyFile.txt



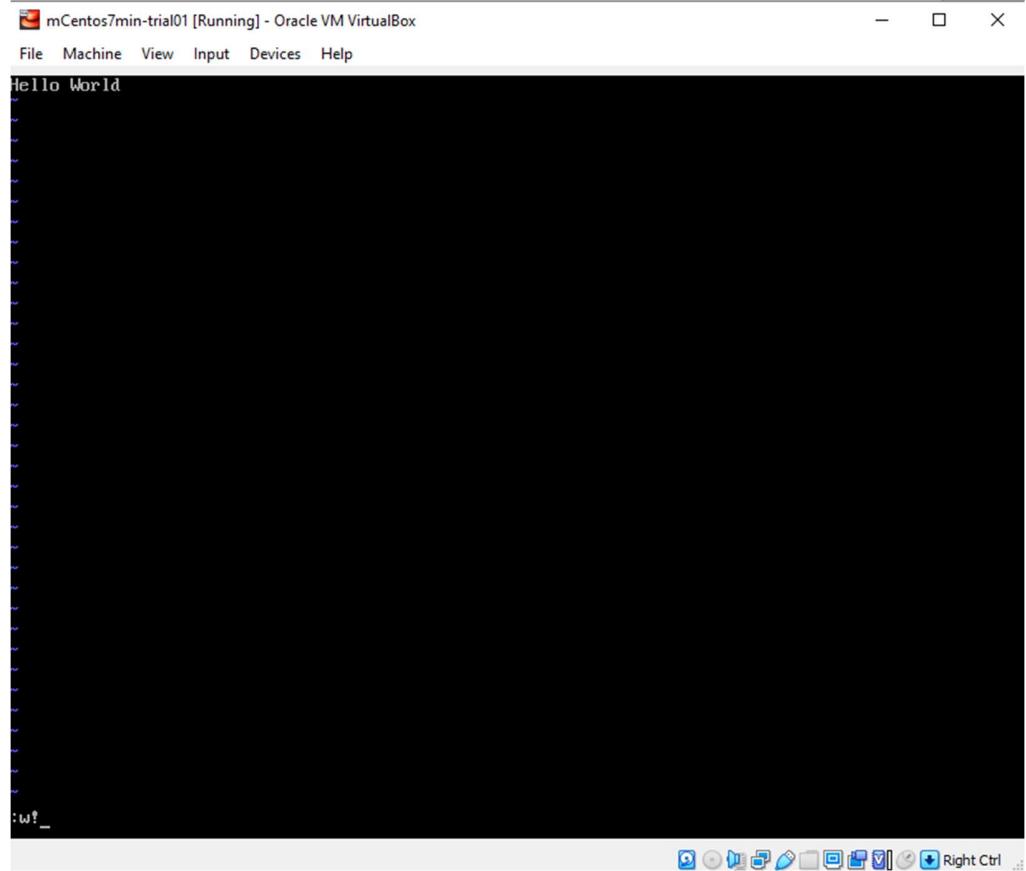
This command runs the vi editor and create a text file named “MyFile.txt” if the file does not exist (vi opens the file if the file exists.). You are now inside the command mode. Next, I want to start typing in the file. Therefore, I press “i” on the keyboard to enter the input mode. If you are in the input mode with command I, you can see the wording “—INSERT —“ at the bottom left of the window.



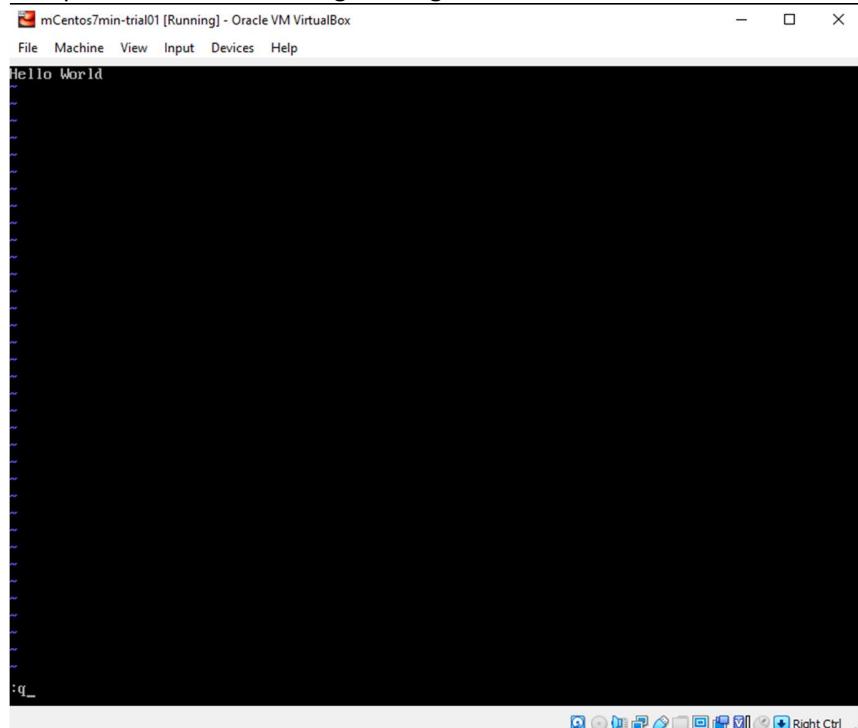
Now, type “Hello world”.



Next, I want to save the file. Press <Esc> key on the keyboard. This makes you enter command mode. Then, type “:w!” to write the file. Here the “!” means that I want to overwrite the file “MyFile.txt”. Note that this command will appear in the bottom left corner as shown below.



Finally, I want to quit the vi editor. Press <Esc> on the keyboard to enter command mode. Type “:q” and press <Enter>. This quits the vi editor.



Now, we return to the shell. Type “ls”. It shows you that we have two files inside the directory including the newly created file “MyFile.txt”.

```
[root@localhost ~]# ls  
anaconda-ks.cfg  MyFile.txt  
[root@localhost ~]#
```

To print the contents of the file, type “cat MyFile.txt”. Note that the capital letters are important. This print out the contents “Hello World” of the file.

```
[root@localhost ~]# ls  
anaconda-ks.cfg  MyFile.txt  
[root@localhost ~]# cat MyFile.txt  
Hello World  
[root@localhost ~]#
```

This is a short example using vi editor. Try using other commands.

## 2.3. replace and delete a character (r x X)

When in command mode (it doesn't hurt to hit the escape key more than once) you can use the x key to delete the current character. The big X key (or shift x) will delete the character left of the cursor. Also when in command mode, you can use the r key to replace one single character. The r key will bring you in insert mode for just one key press, and will return you immediately to command mode.

**Table 2.3. replace and delete**

command	action
x	delete the character below the cursor
X	delete the character before the cursor
r	replace the character below the cursor
p	paste after the cursor (here the last deleted character)
xp	switch two characters

## 2.4. undo and repeat (u .)

When in command mode, you can undo your mistakes with u. You can do your mistakes twice with . (in other words, the . will repeat your last command).

**Table 2.4. undo and repeat**

command	action
u	undo the last action
.	repeat the last action

## 2.5. cut, copy and paste a line (dd yy p P)

When in command mode, dd will cut the current line. yy will copy the current line. You can paste the last copied or cut line after (p) or before (P) the current line.

**Table 2.5. cut, copy and paste a line**

command	action
dd	cut the current line
yy	(yank yank) copy the current line
p	paste after the current line
P	paste before the current line

## 2.6. cut, copy and paste lines (3dd 2yy)

When in command mode, before typing dd or yy, you can type a number to repeat the

command a number of times. Thus, 5dd will cut 5 lines and 4yy will copy (yank) 4 lines. That last one will be noted by vi in the bottom left corner as "4 line yanked".

**Table 2.6. cut, copy and paste lines**

command	action
3dd	cut three lines
4yy	copy four lines

## 2.7. start and end of a line (0 or ^ and \$)

When in command mode, the 0 and the caret ^ will bring you to the start of the current line, whereas the \$ will put the cursor at the end of the current line. You can add 0 and \$ to the d command, d0 will delete every character between the current character and the start of the line. Likewise d\$ will delete everything from the current character till the end of the line. Similarly y0 and y\$ will yank till start and end of the current line.

**Table 2.7. start and end of line**

command	action
0	jump to start of current line
^	jump to start of current line
\$	jump to end of current line
d0	delete until start of line
d\$	delete until end of line

## 2.8. join two lines (J) and more

When in command mode, pressing J will append the next line to the current line. With yyp you duplicate a line and with ddp you switch two lines.

**Table 2.8. join two lines**

command	action
J	join two lines
yyp	duplicate a line
ddp	switch two lines

## 2.9. words (w b)

When in command mode, **w** will jump to the next word and **b** will move to the previous word. w and b can also be combined with d and y to copy and cut words (dw db yw yb).

**Table 2.9. words**

command	action
w	forward one word
b	back one word
3w	forward three words
dw	delete one word
yw	yank (copy) one word
5yb	yank five words back
7dw	delete seven words

## 2.10. save (or not) and exit (:w :q :q! )

Pressing the colon : will allow you to give instructions to vi (technically speaking, typing the colon will open the **ex** editor). **:w** will write (save) the file, **:q** will quit an unchanged file without saving, and **:q!** will quit vi discarding any changes. **:wq** will save and quit and is the same as typing **ZZ** in command mode.

**Table 2.10. save and exit vi**

command	action
:w	save (write)
:w fname	save as fname
:q	quit
:wq	save and quit
ZZ	save and quit
:q!	quit (discarding your changes)
:w!	save (and write to non-writable file!)

The last one is a bit special. With **:w!** vi will try to **chmod** the file to get write permission (this

works when you are the owner) and will **chmod** it back when the write succeeds. This should always work when you are root (and the file system is writable).

## 2.11. Searching (/ ?)

When in command mode typing / will allow you to search in vi for strings (can be a regular expression). Typing /foo will do a forward search for the string foo and typing ?bar will do a backward search for bar.

**Table 2.11. searching**

command	action
/string	forward search for string
?string	backward search for string
n	go to next occurrence of search string
/^string	forward search string at beginning of line
/string\$	forward search string at end of line
/br[aeio]l	search for bral brel bril and brol
\<he\>	search for the word <b>he</b> (and not for <b>here</b> or <b>the</b> )

## 2.12. replace all ( :1,\$ s/foo/bar/g )

To replace all occurrences of the string foo with bar, first switch to ex mode with : . Then tell vi which lines to use, for example 1,\$ will do the replace all from the first to the last line. You can write 1,5 to only process the first five lines. The s/foo/bar/g will replace all occurrences of foo with bar.

**Table 2.12. replace**

command	action
:4,8 s/foo/bar/g	replace foo with bar on lines 4 to 8
:1,\$ s/foo/bar/g	replace foo with bar on all lines

## 2.13. reading files (:r :r !cmd)

When in command mode, :r foo will read the file named foo, :r !foo will execute the command foo. The result will be put at the current location. Thus :r !ls will put a listing of

the current directory in your text file.

**Table 2.13. read files and input**

command	action
:r fname	(read) file fname and paste contents
:r !cmd	execute cmd and paste its output

## 2.14. practice: vi(m)

1. Start the vimtutor and do some or all of the exercises.
2. What 3 key sequence in command mode will duplicate the current line.
3. What 3 key sequence in command mode will switch two lines' place (line five becomes line six and line six becomes line five).
4. What 2 key sequence in command mode will switch a character's place with the next one.
5. Copy /etc/passwd to your ~/passwd. Open the last one in vi and press Ctrl v. Use the arrow keys to select a Visual Block, you can copy this with y or delete it with d. Try pasting it.
6. What does dwP do when you are at the beginning of a word in a sentence ?

---

## 2.15. solution: vi(m)

1. Start the vimtutor and do some or all of the exercises.

vimtutor

2. What 3 key sequence in command mode will duplicate the current line.

YYP

3. What 3 key sequence in command mode will switch two lines' place (line five becomes line six and line six becomes line five).

ddp

4. What 2 key sequence in command mode will switch a character's place with the next one.

xp

5. Copy /etc/passwd to your ~/passwd. Open the last one in vi and press Ctrl v. Use the arrow keys to select a Visual Block, you can copy this with y or delete it with d. Try pasting it.

```
cp /etc/passwd ~  
vi passwd  
(press Ctrl-v)
```

6. What does **dwwP** do when you are at the beginning of a word in a sentence?

**dwwP** can switch the current word with the next word.

**GNU Free Documentation License**