Editing files

Unit objectives

After completing this unit, you should be able to:

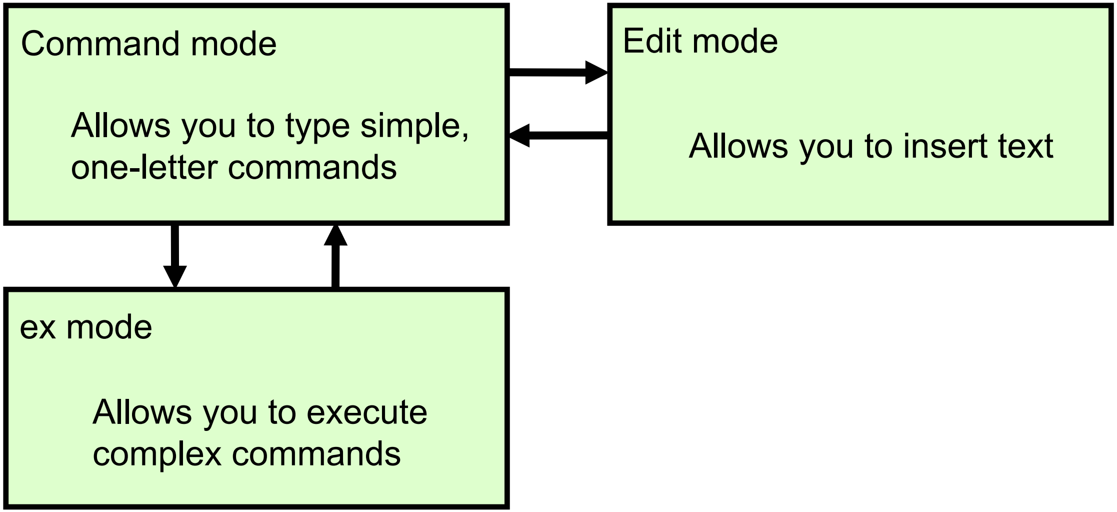
- Determine the type of file using **file**
- Edit text files with Vi
- Discuss other text file editors, such as KEDIT
- Discuss the ways non-text files can be edited

The Vi text editor

- Vi is the default editor in all UNIX operating systems.
- It is usually the only editor available in emergencies.
- It is relatively hard to learn, but it is very powerful.
- As a Linux user, you should be able to use Vi for basic editing tasks.
  - But OK if you prefer another editor for daily work
- Vi in Linux is usually Vim (Vi Improved).
  - Syntax highlighting
  - Arrow keys, Del, BS work in insert mode
  - Multi level undo
  - Mouse support

Vi modes

- Vi knows three modes of operation.
  – Command mode (for simple, one-letter commands)
  – Edit mode (insert text)
  – ex mode (for complicated commands)
- You can easily change between modes.


Command mode
Allows you to type simple, one-letter commands
Edit mode
Allows you to insert text
ex mode
Allows you to execute complex commands

Editing files

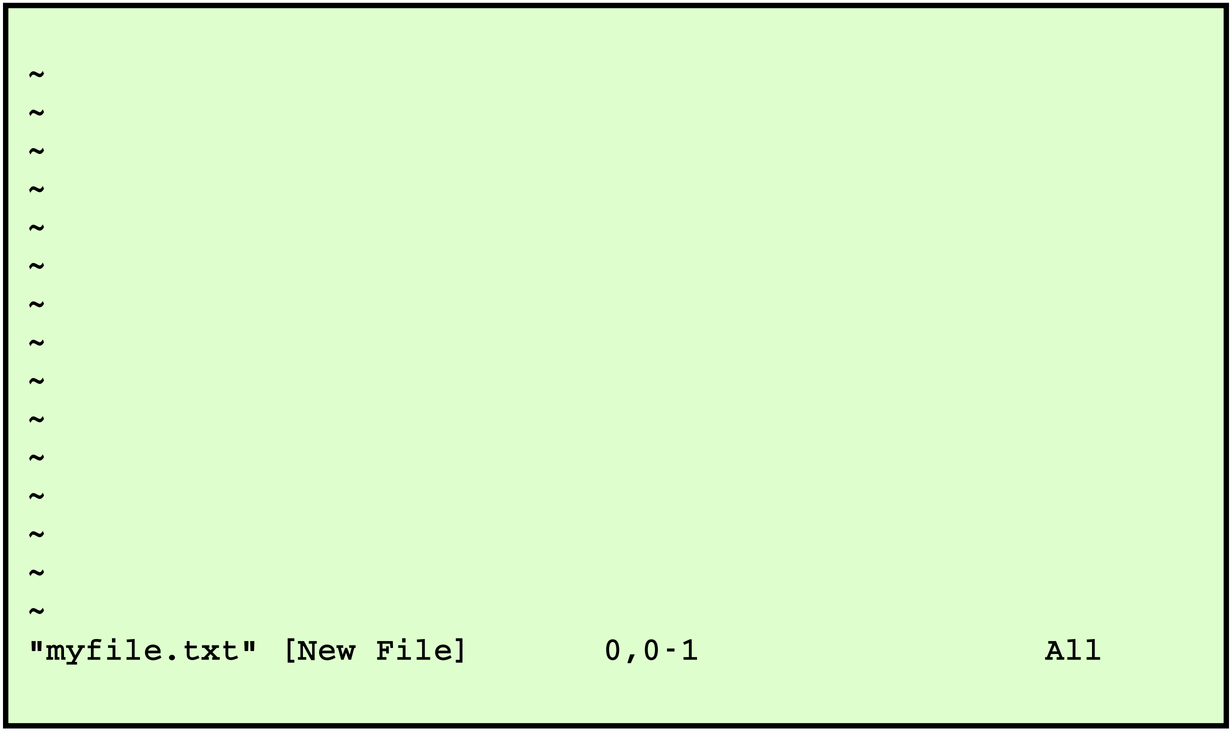
- Use the **file** command to determine the content of a file.

```
$ file /etc/passwd
/etc/passwd: ASCII text
$ file /usr/bin/passwd
/usr/bin/passwd: ELF 32-bit LSB executable
```

- To edit text files, use an editor.
- Non-text files can only be changed using the application that created them or a hex editor.
- However, most configuration files under Linux are text files.

Starting Vi

- `$ vi myfile.txt`



```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"myfile.txt" [New File]      0,0-1                  All
```

Cursor movement in command mode

H

1G

^b page

^u ½ page

<up-arrow>

k

h

<left-arrow>

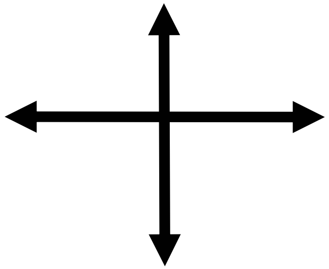
^

b

(

{

M



l

<right-arrow>

$

w word

) sent

} para

j

<down-arrow>

^d

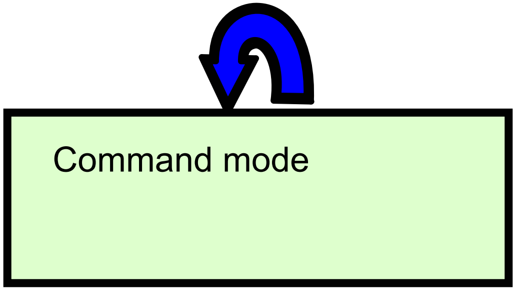
^f

L

G

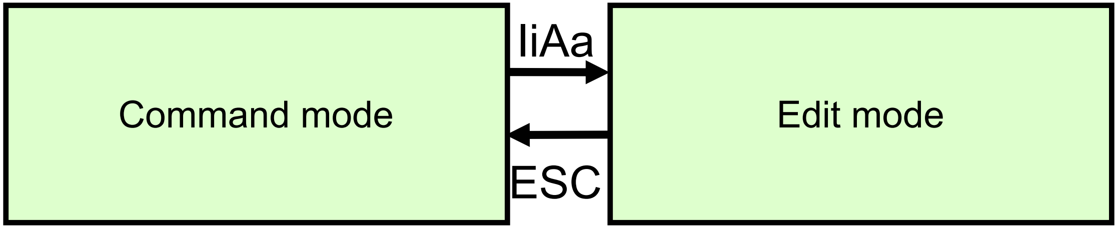
Editing text in command mode

- To delete a single character under cursor: **x**
- To delete a single character left of cursor: **X**
- To replace a single character: **r**
- Undo the last change: **u**
- To repeat last command: **.**
- To join two lines together: **J**

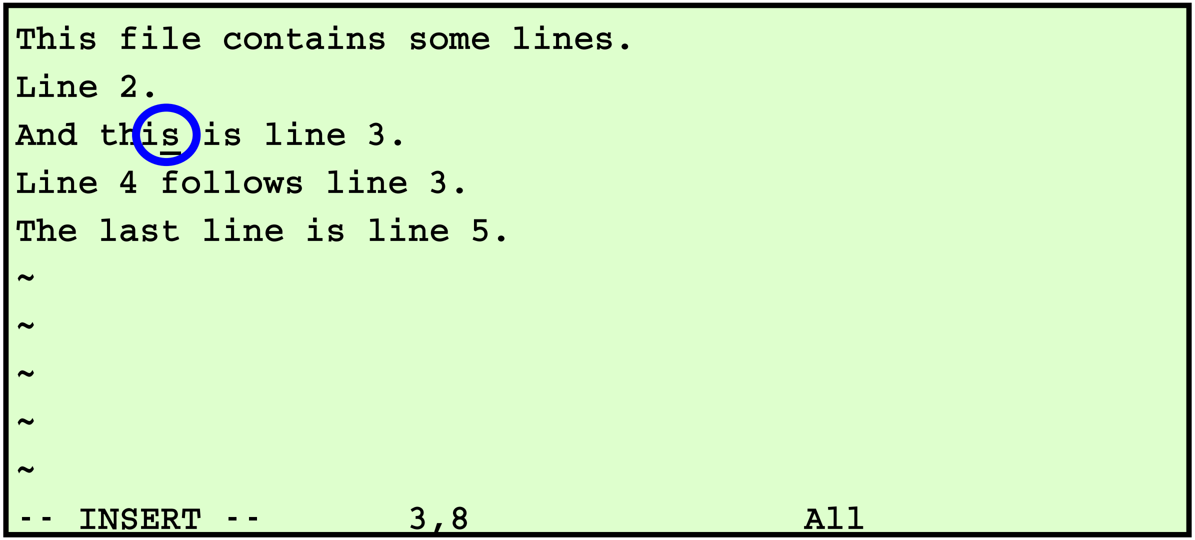


Switching to edit mode

- To insert text at beginning of line: **I**
- To insert text before cursor: **i**
- To append text after cursor: **a**
- To append text at end of line: **A**
- To go back to command mode: **<ESC>**



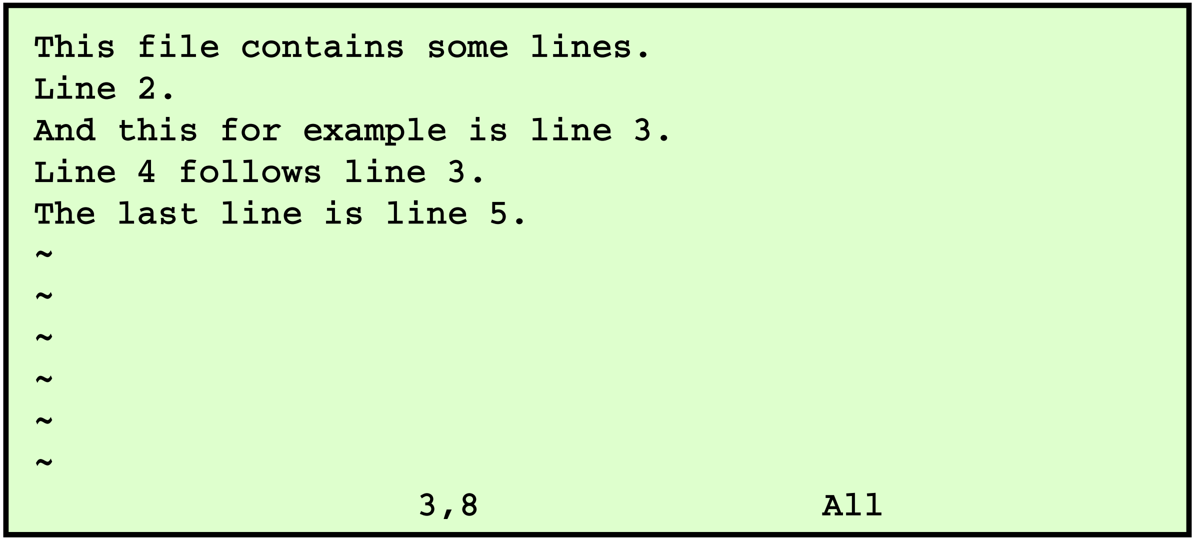
Adding text in edit mode



```
This file contains some lines.
Line 2.
And this is line 3.
Line 4 follows line 3.
The last line is line 5.
~
~
~
~
~
-- INSERT --        3,8                 All
```

- Keystroke **i** switches Vi to edit mode. New characters can be inserted at the current position of the cursor.

Exiting the edit mode



```
This file contains some lines.
Line 2.
And this for example is line 3.
Line 4 follows line 3.
The last line is line 5.
~
~
~
~
~
~
                  3,8                  All
```

- Keystroke **ESC** leaves the edit mode.

Searching for patterns

- To search for a pattern (in command mode): `/<pattern>`
- To repeat the previous search: **n**

```
This file contains some lines.
Line 2.
And that for example is line 3.
Line 4 follows line 3.
The last line is line 5.
~
~
~
~
~
~
/line
```

Replacing patterns

- Advanced search and replace can be done in ex mode.
- To replace old text with new text use the following command:

```
:1,$s /old/new/g
```

```
This file contains some lines.
Line 2.
And that for example is line 3.
Line 4 follows line 3.
The last line is line 5.
~
~
~
~
~
~
: 1,$s/this/that/g
```

Cut, copy, and paste

- To cut a whole line into buffer: **dd**
- To copy a whole line into buffer: **yy**
- To cut a word from the current cursor position to its end: **dw**
- To paste contents of buffers here: **p**
- To cut or copy multiple lines, proceed command by number: **3dd**, **8yy**

Cut and paste

```
This file contains some lines.
Line 2.
And that for example is line 3.
Line 4 follows line 3.
The last line is line 5.
```

- Cut line three by typing **dd**.

```
This file contains some lines.
Line 2.
Line 4 follows line 3.
The last line is line 5.
And that for example is line 3.
```

- Insert it after line four by typing **p**.

Copy and paste

```
This file contains some lines.
Line 2.
And that for example is line 3.
Line 4 follows line 3.
The last line is line 5.
```

- Copy line three by typing **yy**.

```
This file contains some lines.
Line 2.
And that for example is line 3
Line 4 follows line 3.
The last line is line 5.
And that for example is line 3.
```

- Insert it after line five by typing **p**.

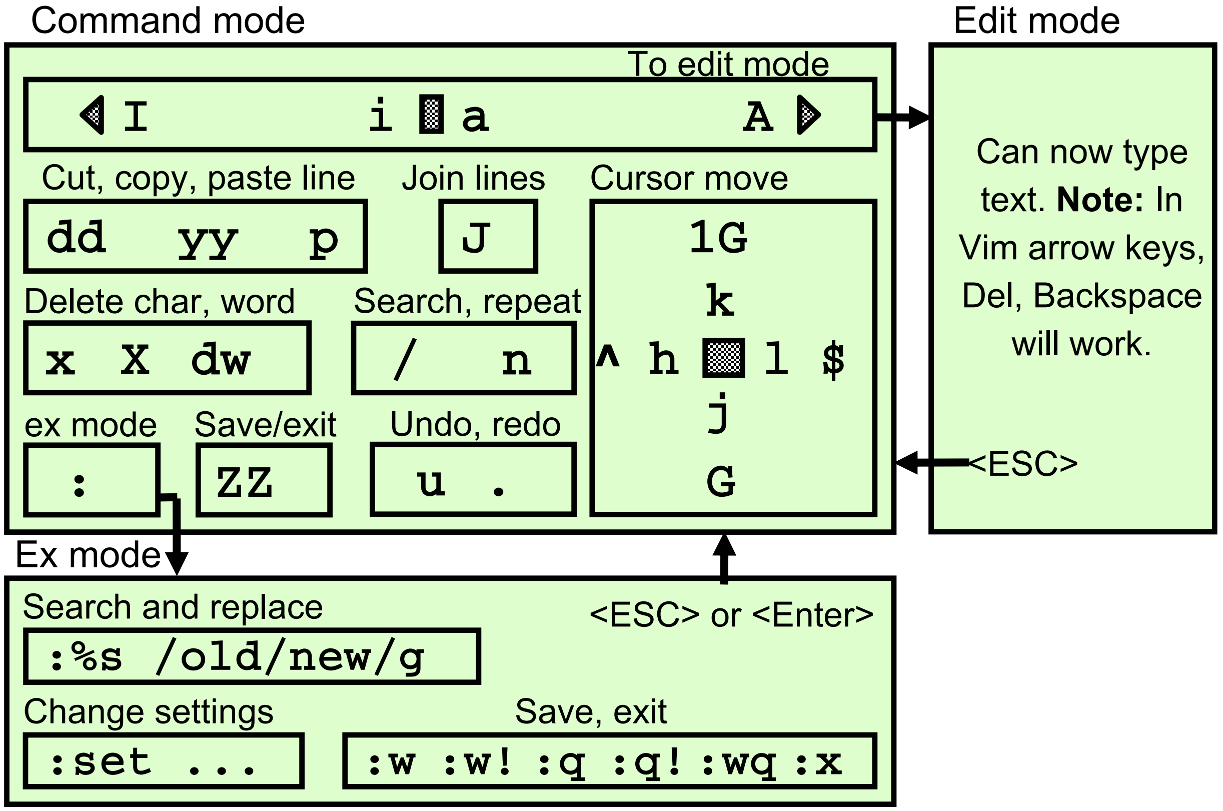
Vi options

- Options entered in ex mode change the behavior of the Vi editor.
  - `:set all`
  - `:set autoindent/noautoindent`
  - `:set number/nonumber`
  - `:set list/nolist`
  - `:set showmode/noshowmode`
  - `:set tabstop=x`
  - `:set ignorecase/noignorecase`
  - `:set wrapmargin=x`
  - `:set hlsearch/nohlsearch`
  - `:syntax on/off`
  - `:set fileformat=dos/unix`
- To make an option available to all Vi sessions, put it into a `.exrc` or `.vimrc` file in your home directory.

Exiting Vi

- To save in ex mode
  - **:w**
- To forcefully save file in ex mode
  - **:w!**
- To quit without saving in ex mode
  - **:q**
- To forcefully exit in ex mode
  - **:q!**
- To save and exit in ex mode (recommended)
  - **:wq**
- To save and exit in ex mode, shorter
  - **:x**
- To save and exit in command mode
  - **ZZ**

Vi cheat sheet



Command mode

To edit mode

`I` `i` `a` `A`

Cut, copy, paste line

`dd` `yy` `p`

Join lines

`J`

Cursor move

`1G` `k` `^` `h` `l` `$` `j` `G`

Delete char, word

`x` `X` `dw`

Search, repeat

`/` `n`

ex mode

`:`

Save/exit

`ZZ`

Undo, redo

`u` `.`

Edit mode

Can now type text. **Note:** In Vim arrow keys, Del, Backspace will work.

<ESC>

Ex mode

Search and replace

```
:%s /old/new/g
```

<ESC> or <Enter>

Change settings

```
:set ...
```

Save, exit

```
:w :w! :q :q! :wq :x
```

Other editors

- A typical Linux distribution comes with a large number of editors.
- Text mode editors:
  - Pico (really simple)
  - Original Vi
  - Emacs (even more powerful and complicated than Vi)
- Graphical mode editors:
  - KVim, KEDIT, KWrite
  - gVim, gedit
- Hex editors allow you to change non-text files if you know the internal structure.
  - KHexEdit
  - Emacs (in hexl-mode)

Unit review

- The most common editor on any UNIX is Vi.
- Vi has three modes of operation: command mode, edit mode, and ex mode.
- Vi makes a copy of the file you are editing in an edit buffer. The contents are not changed until you save the changes.
- A typical Linux distribution comes with a lot of other editors as well.

Checkpoint

1. True or False: You need to learn Vi because Vi is the best editor for any job.

2. What does the **file** command do?
   a. It looks at the extension to determine the type of file.
   b. It looks at the first few characters of the file and compares this to a database of known file types.
   c. It asks the kernel for information about the file.
   d. It makes a wild guess.

3. What is a hex editor?

Checkpoint solutions

1. True or False: You need to learn Vi because Vi is the best editor for any job.

   The answer is false. Vi is just the most common editor available on Unix-style systems and is thus more likely to be there when you need to edit a file. Indeed, it might be the only editor available. Having Vi skills can be quite handy on unfamiliar Unix-style systems.

2. What does the **file** command do?

   a. It looks at the extension to determine the type of file.

   b. It looks at the first few characters of the file and compares this to a database of known file types.

   c. It asks the kernel for information about the file.

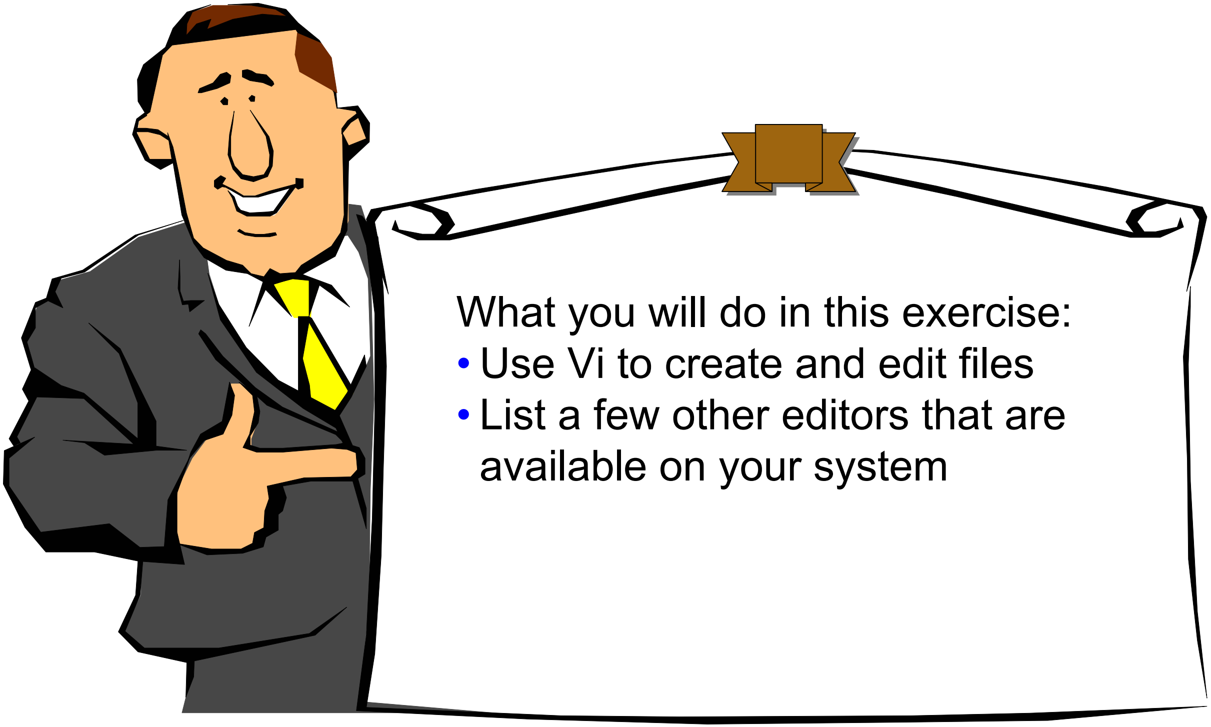
   d. It makes a wild guess.

   The answer is it looks at the first few characters of the file and compares this to a database of known file types..

3. What is a hex editor?

   The answer is a hex editor is an editor that treats the contents of a file as a series of bytes, displays those bytes in hexadecimal format, and allows the hexadecimal representation of the contents of the file to be edited.

Exercise: Editing files



What you will do in this exercise:

- Use Vi to create and edit files
- List a few other editors that are available on your system

Unit summary

Having completed this unit, you should be able to:

- Determine the type of file using **file**
- Edit text files with Vi
- Discuss other text file editors, such as KEDIT
- Discuss the ways non-text files can be edited