### VI For Novices

FILE: A file is a collection of information. Examples: a letter, a memo, a program, a list of names and ss#s, etc.

**FILENAMES**: Every file has a name that identifies it. These **filenames** are chosen by the computer-users who create the files in the first place. On the Sun Sparcstation, a filename can be any combination of letters and digits.

**EDITOR**: An **editor** is a program on a computer that helps you create files.

VI: The editor on the Sun Sparcstation is called vi. Probably it once stood for something like "visual interactive".

**STARTING VI**: To start vi, you need to log into the computer and wait for the shell prompt, which probably looks something like:

atrium24>

Then you type

atrium24> vi filename

where *filename* is the name of a file that you would like to create or already exists. Be sure to hit RETURN after typing the filename.

**THE VI SCREEN**: When vi starts the entire screen changes, and the file you are working with is displayed. Each line in the file is shown on the screen (unless there are too many lines to fit). Where the file ends (where there are no more lines) a tilde ( $\sim$ ) appears. If this is a new file that is being created then there are no lines yet and only tildes appear. In that case you will see something like:

```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"firstfile" [New file]
```

(Assuming you had typed "vi firstfile").

VI MODES: Basically there are two kinds of things you can do in vi. You can type data that will become part of your file or you can issue commands to vi (like delete a character or a line). These two ways of working are called modes. In input mode, you type data into your file. In command mode you give commands to vi. Vi

remembers which mode you are in. If you are in <u>command mode</u>, you <u>cannot enter data</u> into your file; if you are in <u>input mode</u> you <u>cannot give vi commands</u>. So there has to be a way of switching back and forth between command and input modes.

CHANGING MODES (1): Vi always starts in command mode.

CHANGING MODES (2): To go from command mode to input mode: type i or a

CHANGING MODES (3): To go from input mode to command mode: hit the ESC key

NOTE: None of the commands in vi (like i a ESC) require hitting return after the command unless otherwise indicated.

**INPUT MODE**: Once you are in **input mode**, everything you type goes into the file you are making. There is no "wrap-around"— when you want to make new lines in vi, you have to hit the RETURN key, just like on a typewriter. If you make a mistake, you can use the backspace key to go backwards over it and fix it. When you want to do something else (like delete a line, save your work, or just go home) you have to his ESC to leave input mode.

## **COMAND MODE**: Here are a few of the commands that you can use in vi:

command	what it does		Needs a RETURN?
X	erases the character under the cursor		NO
dl	erases the entire line that the cursor is one		NO
ZZ	saves the file and exits the editor		NO
:q!	exits the editor WITHOUT saving the file		YES
i	goes into input mode		NO
a	goes into input mode		NO
space	moves forward one character		NO
h	moves backwards one character		NO
j	goes down one line		NO
k	goes up one line		NO
J	joins the line the cursor is on with the next line		NO
u	undoes the last command!		NO
SUMMARY:	moving commands: erasing commands:	h j k space dd x	
	leaving commands: join command: oooops command:	:q! ZZ J u	

# NOTES:

• case is significant: J≠j be sure that you don't set caps-lock.

**BEEP**: When vi is in command mode, if you type something it doesn't understand, it will make a beeping sound. For example, if you are at the end of a line and you type space (to move forward to the next character) it will beep because it can't go on, or if you hit ESC (when you are already in command mode).

Sometimes a vi user can feel lost and be unsure of which mode he or she is in. If that ever happens, hit ESC. After

doing that, you will <i>always</i> be in command mode. Then you can stay is by typeing <b>i</b> or <b>a</b> .	in that mode if you like or go into input mode

### Survival VI

#### **General Characteristics:**

- EXTREMELY powerful, flexible, convenient
- EXTREMELY unfriendly, bordering on hostile
- EXTREMELY redundant (10 ways to do the same thing)

**Modes:** input mode, command mode (no visible difference!)

Command-->input mode: i a

Input--->command mode: ESC

Moving Commands: (note:case is significant!)

small movements: h j k SPACE screen top/middle/bottom: H M B forward/backward one line: + - beginning/end of same line: ^ \$

next/previous screen: ctrl-F ctrl-B next/previous word: w b

end of current word:

scroll down/up: ctrol-D ctrl-U

jump to line N: NG (example: 178G takes you to line 178) jump to last line: \$G (note: in general "\$" means "last")

find pattern: /pattern (example: /unix takes you to next instance of unix)

**Deleting Commands:** 

delete current line: dd delete current word: dw delete current character: x

Joining current and next lines: J

**File Commands:** :f (shows file being editted)

:f name (changes name of file being editted)

:w (save what you have under the name of the file being editted)

:e name (edit a different file)

:q (quit)

NOTE: ! after w,e or q means FORCE (i.e. changes will be lost)

Global Substitution: :line#,line#s/matchpattern/replacepattern/ (just first instance on each line)

:line#,line#s/matchpattern/replacepattern/g (every instance on each line)

Global Pattern Commands: :g/matchpattern/p print every line with that matchpattern

:g/matchpattern/d delete every line with that matchpattern

Match patterns are regular expressions. Special Characters Are:

- . matches any character
- \* 0 or more repititions of previous character
- [] matches 1 character within the brackets. Example: [abxy] will match a "b" or an "x" or a "y" or an

"a"

- ^ matches beginning of line (not a character)
- \$ matches end of line (not a character)