UNIVERSITY OF CALIFORNIA

LICK OBSERVATORY TECHNICAL REPORTS

No. 30

WORD PROCESSING MADE EASY

Sue Robinson

Santa Cruz, California

August, 1982

TABLE OF CONTENTS

WORD PROCESSING MADE EASY

Lick Observatory Technical Report No.30

Sue Robinson

INTRODUCTION

This paper has been written to help a person, who is unfamiliar with computers, learn to do word processing using the UNIX operating system at the UCSC Computer Center. When I began learning word processing, many of the current manuals seemed to be too complicated, so this paper is presented in the hope that it will make word processing easier to understand. You will derive the most benefit if you follow the examples shown in sequence.

First of all, it is useful to understand that the "operating system" on the computer which runs the video terminal and printer is called UNIX. A number of programs are "run" under UNIX and several different ones are needed to produce an edited paper; they are required for entering text, converting text into paragraphs and pages, printing papers containing tables and equations, checking text for spelling errors, and many other features that we take for granted when we manually type a paper. As mentioned above, UNIX is the main system, and the other programs are run under the control of UNIX. UNIX identifies itself by the percent sign (%) which appears at the beginning of a line on the screen, and requests to UNIX (i.e. program requests) can only be made after % appears.

The first program you will use is called "vi" which is used mainly for video terminals. Program "vi" is an "editor" which enables text to be typed, alterations, additions and corrections to be made, and then stored in the computer. Other programs can be used to check for spelling errors, and to put the text into whatever form is needed for the finished product. This will be discussed later.

The first thing to find out is how to "log on", enter some text, and "log off". Basically the routine is this (see example on next page):

1) tell the computer at the Computer Center, via the video terminal, on which computer you have an account (you must have an account with the Computer Center before you can use the system);

2) give account name and secret password with that particular computer;

3) be accepted by the computer; % will appear to indicate acceptance;

4) tell UNIX what type of terminal you are using so that signals to and from the terminal and the computer can be adjusted accordingly (the wording for the type of terminal should be written, for easy reference, on the terminal itself - if you are not sure, ask someone who is familiar with the terminal);

5) ask for a program;

6) set up a file;

7) enter text;

8) store text into computer;

9) log off.

The following is an example of the steps that you take when starting to use the word processor and should be helpful in using "vi". Characters underlined indicate responses you should give to the questions printed on the screen but should be altered where necessary for your own particular situation. Some peculiarities regarding "logging on" are noted at the end of the example.

TURN ON POWER to video terminal.

LOGGING ON

WHICH COMPUTER?3 [cr] - (cr = carriage return)

GO

login: oreck [cr] - (account name)

Password:secret [cr] - (password does not print)

(A message of the day will appear on the screen.)

% setenv TERM tvi912 [cr] - (type of terminal)

% vi filename [cr] - (program "vi", plus a filename)

If you are starting a new file with "filename", then "no lines in the buffer" will appear on the screen which will indicate that "filename" has not already been used (does not exist) and that you have started a new file with this name.

If "filename" has already been established as a file by you, then the number of lines and characters in "filename" will be shown in addition to "no lines in the buffer". Text already entered in "filename" will appear on the screen.

You have now logged on and set up a file called "filename".

SOME PECULIARITIES WHICH OCCUR WHEN LOGGING ON

There are times when the VIDEO TERMINAL does not ask "WHICH COMPUTER?" after the power is turned on. If this happens, hit the carriage return key [cr] which will make the computer respond "WHICH COMPUTER?". You can then log on as shown in the above example.

The PRINTER nearly always responds "DOES NOT EXIST" after the power is turned on. Hit the carriage return key [cr] and the printer will then respond with "WHICH COMPUTER?" You can then log on as shown in the above example.

Sometimes the PRINTER will say "WHICH COMPUTER? DISCONTINUED". When this happens, hit the carriage return key [cr] and the printer will respond with "WHICH COMPUTER?" You can then log on as shown in the above example.

"UNAVAILABLE" means that the computer is "down". Try again later.

WHEN THE COMPUTER RESPONDS "BUSY WAIT"

After you have told the Computer Center that you want computer 3, sometimes the video terminal or printer will respond "BUSY WAIT 002?" (or some number other than 002). This means that the Computer Center has too many people trying to use the computer, and you are second in line to log on when a vacancy (line) exists. If you do not want to wait, then you respond by typing "n" (for "no") after "BUSY WAIT 002?" and turn off the power (or you can just turn off the power without responding at all). If you want to wait for a line, then type "y [cr]". The computer will respond with "WAITING". When the computer is able to accept you, it will continue with the rest of the steps for logging on. The following is an example of how to respond to the computer when it is busy and you want to wait to log on:

WHICH COMPUTER?3 [cr]

BUSY WAIT 002?y [cr]

WAITING

Eventually, when a line becomes available, the computer will respond with the rest of the steps for logging on.

If the delay is longer than you anticipated and you decide to change your mind and not wait, just turn off the power to the video terminal or printer.

WRITING TEXT, STORING IT, AND LOGGING OFF

After you have set up a file called "filename", it is a good idea to give line numbers to the text you are about to enter. This is done by:

[esc]:set number [cr] - (produces the text in numbered lines)

A line number is put at the beginning of each line in your file. This is most useful when alterations and additions are needed.

Having set numbers, you can now enter text. Text is entered by first hitting the escape key [esc], followed by the letter a (for append). The letter a does not appear on the screen. Enter text. Unlike using an ordinary typewriter, text does not have to be typed line by line to end at approximately the same place as the previous line. Even if you type one word per line, the computer will print out the text in a continuous form to fill each line.

Simple corrections can be made as errors occur by using the backspace key followed by the correct letter(s). See "Corrections to Existing Text" for others.

Finish text and hit the escape key [esc].

THE ESCAPE KEY [esc] IS HIT AFTER APPENDING, AND BEFORE ALL COMMANDS.

The text must now be stored in "filename" on the computer as follows:

[esc]:w [cr] - writes the text into the computer and saves it for future use. This should be done frequently during the time you are entering text otherwise you could lose all your text if the computer fails. Type [esc]a, for append, to continue with text.

[esc]:q [cr] - quits the "vi" program, and gets back to UNIX (%).

Should you want to quit without writing ("write" means to store the text you have typed so that you can use it again later), type :q!, otherwise computer warns that text has not been stored.

% sign will appear after quitting, which means you must either start anew at % with vi filename (or with one of the other programs which you will learn about later), OR hit d while depressing [ctrl] key (called "control d"), which will produce "logout". Sometimes the computer asks you to type "logout". The power can then be turned off at the terminal.

CAUTION: It can be easy to fall into the trap of turning off the terminal and forgetting to log off. If you do not log off, your account will be charged for the time even if the terminal is turned off.

Note: [esc]ZZ may be used to write and quit in lieu of [esc]:w plus [esc]:q.

The preceding sections give a simple example of logging on, typing text, and logging off.

Note: "filename" has been used in this paper to define only one of many filenames which can be created within your account.

ADDING TO EXISTING TEXT

Having logged on and set line numbers, you will notice that the cursor (the lighted rectangle, the size of one character, which moves along as you type), is at the beginning of the text already entered. To bring the cursor to the end of the existing text type [esc]G. This should bring the cursor to the beginning of a new line following the last line of text entered. Type [esc]a for append.

On some occasions the cursor will go to the beginning of the last line of text entered. If that should happen, hit [esc]o which will bring the cursor to the beginning of a new line for typing. DO NOT TYPE a FOR APPEND IN THIS INSTANCE, since o puts you into the append mode automatically.

You can now add to the previous text which was stored on file.

SUMMARY OF COMMANDS FOR WRITING, ADDING TO TEXT, AND LOGGING OFF

[esc]a	appends text
[esc]:w[cr]	writes and saves it
[esc]:q[cr]	quits the "vi" program and gets you back to UNIX (%) (only if text has been written)
[esc]:q![cr]	quits the "vi" program without writing (and gets you back to UNIX [%])
[esc]ZZ	may be used to write and quit (gets you back to UNIX)
[esc][ctrl]d	will produce "logout" (only after quit)
[esc]G	brings cursor to last line of text entered in file
[esc]o	opens next line and sets APPEND mode (O opens previous line AND APPENDS)

CORRECTIONS TO EXISTING TEXT

1. POSITIONING THE CURSOR

When a word(s) within the text needs altering, correcting, or replacing, it is first necessary to place the cursor at the beginning of the incorrect word:

IF A TYPING ERROR HAS OCCURRED WITHIN THE LINE BEING TYPED, for instance three words back, then hit the [esc] key (because you are giving a command), followed by 3b. The b command tells the cursor to go back one word. The number preceding b indicates how many words to go back. Therefore 3b means: go back three words.

The command w is similar, but w goes forward in the line by one word; 3w would go three words forward in the line.

IF A TYPING ERROR HAS OCCURRED ON ANOTHER LINE, the G command is used, PRECEDED by the line number on which the error has occurred. If you need to go to line 15, type [esc]15G. This will bring you to the beginning of line 15. The b and/or w command (see above) should be used to bring the cursor to the first letter of the word to be corrected.

The arrow keys (preceded by [esc]) can be used, but are actually less efficient.

THE CURSOR IS NOW POSITIONED AT THE FIRST LETTER OF THE WORD TO BE CORRECTED.

2. MAKING CORRECTIONS (having already positioned the cursor)

Simple corrections can be made while entering text, as errors occur, by using the backspace key followed by the correct letter(s).

When a complete word needs correcting or replacing, position the cursor at the beginning of the word and type [esc]cw (CHANGE WORD). A $ sign will appear at the end of the incorrect word. Now type the new word (or words). End the correction by hitting [esc]. The new word(s) will replace the characters from the cursor to the dollar sign. Typing c3w would enable you to replace three words from the cursor to the $ sign with one or more words.

When one letter in a word needs REPLACING, position the cursor by moving forward with the space bar until the cursor is on the letter to be corrected. Type [esc]r; type the correct letter.

If a word or letter needs INSERTING, position the cursor at the place where the insert is to go, and type [esc]i. Now you can type the word, letter, or sentence which is to be inserted. Hit [esc] when finished. Sometimes the line will be too long after inserting words. If so, hit [cr] before [esc] to make two lines.

When a word needs a LETTER DELETED, put the cursor on the letter, type [esc]x which will remove it.

DELETING A WORD is done by positioning the cursor at the beginning of the word to be deleted and typing [esc]dw; d3w will delete the next three words starting from the cursor.

DELETING A COMPLETE LINE from the text is done by typing [esc]dd.

By typing [esc]d$ you can remove all characters between the cursor and the end of the line.

IT IS A GOOD IDEA TO HIT THE [esc] KEY AFTER EACH CORRECTION AND BEFORE CONTINUING WITH THE NEXT. Although this is not always necessary and the terminal will beep at you if you are already in [esc] mode, it is better to follow this procedure until you are really familiar with the different commands, and know which ones leave you in the [esc] mode, and which do not.

WARNINGS

Remember to write ([esc]:w) the corrections, otherwise you will have made them in vain.

It is a good idea to hit the [esc] key after each correction before continuing with the next.

BEWARE - when you are in [esc] mode, the next character struck is a COMMAND. If you forget to append (by typing a or o), and begin typing, the first letter you type will be taken as a command which can cause disastrous results. Should you make this mistake, immediately hit [esc]u (for undo) which will reinstate the text as it was before you gave the mistaken command.

SUMMARY OF COMMANDS FOR CORRECTING TYPING ERRORS

[esc]b	tells the cursor to go back a word (3b tells the cursor to go back three words)
[esc]w	tells the cursor to go forward a word (3w tells the cursor to go forward three words)
[esc]cw	changes word (c3w changes three words)
[esc]r	replaces a letter
[esc]x	deletes letter
[esc]dw	deletes word (d3w deletes three words)
[esc]i	inserts letter or word(s)
[esc]G	brings cursor to beginning of new line for adding text (15G brings cursor to the beginning of line 15)
[esc]dd	deletes line
[esc]d$	deletes characters from cursor to end of line
[esc]u	undoes previous command and restores text as it was prior to that command

WARNINGS

Remember to write ([esc]:w) the corrections.

Hit the [esc] key after each correction before continuing with the next.

BEWARE - when you are in [esc] mode, the next character struck is a command.

TYPING TEXT INTO A FORM FOR PRINTING

Having mastered entering and correcting text, it is now time to learn how to put this into a form which can be printed out. The program which will eventually do this is called "nroff" but the input is done in the same way already learned, i.e., by using the "vi" program.

Commands for "nroff" are inserted into the text using the "vi" program. The "nroff" commands are comprised of a period followed by two letters. For example, .bp is the command for "begin page". Each "nroff" command is typed individually on a separate line (see example below).

There is a tendency to want to precede the text for "nroff" commands with [esc]. Remember, though, that they are not "vi" commands; they are just text entered with the "vi" program at this stage. The "nroff" program will use them as "nroff" commands later.

The following is an example of some "nroff" commands inserted in text followed by the text as it would print out when it is put through the "nroff" program. (See next section for some "nroff" commands and their meanings.)

```
.ce
.ul
NROFF PROGRAM
.sp
.ul
A short summary:
.pp
The "nroff" program gets its information from text
entered using the vi program,
with non-print commands inserted. These commands are
given by typing a period (.) at the beginning of a line,
followed by command letters.
```

This prints out as:

NROFF PROGRAM

A short summary:

The "nroff" program gets its information from text entered using the vi program, with non-print commands inserted. These commands are given by typing a period (.) at the beginning of a line, followed by command letters.

SOME NROFF COMMANDS, THEIR MEANINGS, AND USES

.bp BEGIN PAGE. This command is given at the beginning of a paper before the text is entered or wherever text must begin on a new page. The rest of the text will have the pages set automatically. When you want to start text, for instance, with page 6, then you would give the command .bp 6.

.ll LINE LENGTH. The average width of paper is 8-1/2" which means that one line is 102 characters (12 to the inch) in length from edge to edge. However, margins should be taken into account on both sides of the paper when determining the line length. If a one-inch margin is needed at both the left and right margins, then the line length will be 6-1/2" and the command will be .ll 78. When this command is used in conjunction with the page offset command (see below), the right hand margin is automatically established.

.po PAGE OFFSET. This allows the text to be indented from the left margin. If a one-inch margin is needed, then a .po 12 command will give a left margin of 12 characters, or one inch.

.na NON-ADJUST. This means that the right margin will not be right justified (blocked). Normally the "nroff" program justifies all the text, and therefore the .na command must be given if you do not want the right margin justified. The .na commmand was given for this paper when it was printed off.

.nh NO HYPHENS. Most journals do not like hyphenated words submitted in papers. This command will assure that only the words which are deliberately hyphenated by you in the text will be split between lines. If this command is not given, then the printed version will have many hyphenated words at the ends of the lines - some of which are not very suitable for hyphenating.

.he HEADING (printed at the top of each page). Usually this is used for page numbering, but any kind of heading can be printed with this command. If, for instance, you only want to have pages numbered consecutively in the top center of each page, then the .he command is followed by ''%''. Note that these are four single quote marks. The percent sign is substituted by automatic numbering, starting with page 1. Should you want to insert the date at the top on the right hand side of each page, then the command would be .he ''%'May 1982'. Note that the single quote marks separate the page numbering from the date. As a further step, you may want to include a name at the top left hand margin of each page. This would be done by .he 'Name'%'May 1982'.

Comment: When the .bp (begin page) command is given BEFORE the .he command, the first page is unnumbered, and the following page begins with page 2. When the .bp command is given AFTER the .he command, the first page gets numbered 1.

.fo FOOT. This is similar to the .he command, except that the page numbering etc. is printed at the foot of the page.

.m1
.m2
.m3
.m4 MARGINS at the top and bottom of the page. Margin 1 (.m1) is the area between the top of the page and the page numbering. Margin 2 (.m2) is the area between the page numbering and the beginning of the text. Similarly, margins 3 and 4 are between the end of the text and the foot page numbering, and between the foot page numbering and the bottom of the page respectively. The command .m1 3 would leave three lines of space between the top of the page and the page numbering. Another three lines of space between the page numbering and the text would be given by .m2 3. Similar commands would be given for the foot of the page.

.ls LINE SPACE. Double spacing or triple spacing (or any number of spacings) is done with this command. Double spacing is obtained with the .ls 2 command, and triple spacing with .ls 3.

.sp SPACE. This gives one line of space between the text. If more lines are needed, say four lines, then .sp 4 command will give you four lines of space.

.br BREAK. Begins a new line without leaving a space between text.

.pp PARAGRAPH. This command automatically leaves a line space and indents 5 spaces to start each paragraph.

.ce CENTER. All the text on the line which follows the .ce command will be centered.

.ul UNDERLINE. All text on the line following this command will be underlined.

.in INDENT. When a section of text needs to be indented, the .in command is used. If, for instance, the section is to be indented 10 spaces, then .in 10 will indent the section 10 spaces. All text will be indented 10 spaces until a counter-command of .in -10 is given.

.ti TEMPORARY INDENT. This command is similar to the indent command, except that it affects ONLY the first line of text after the command, (therefore no counter-command is required). The .ti command is often used in conjunction with the .in command, for example, where an indented section needs to have the first line indented further (or exdented, as for references).

CHECKING FOR SPELLING ERRORS

Before you print out your paper, you should put the text through a spelling program. This can be done either on the video terminal or on the printer, depending on your preference. Both have their advantages. If you check the spelling while still using the video terminal, the alterations can be made more readily. However, if the list of spelling errors is printed out on the printer, then this saves you time writing out the words which need to be changed. Whichever you use, terminal or printer, the procedure is virtually the same.

When you have finished entering text with the "vi" program, make sure you have written and stored the text ([esc]:w and [esc]:q OR [esc]ZZ). If you are going to use the video terminal for the "spell" program DO NOT LOG OFF. This is not necessary -- you still want to communicate with UNIX, because you are going to call up another program.

Up until now the "vi" program has been used. This time you are going to use the "spell" program.

If you use the PRINTER, the command for calling up the spell program (after you have logged on and the % sign appears) is:

% spell filename [cr]

There will be some delay, depending on the demand upon the computer, before the spelling errors will be printed out. Once you have a printed list of the errors, you should log off the printer, and log on to the video terminal again so that the errors can be corrected.

If you are using the VIDEO TERMINAL, the command for calling up the spell program (after the % sign appears) is:

% spell filename|more [cr]

By adding "|more" to the command, it will give the spelling errors page by page on the screen, stopping at each page. When you hit the space bar, the next page will appear on the screen. The errors should be written down in their INCORRECT form so that they can be found easily within the text.

When you have the spelling error list, either printed out from the printer or hand-written from the video terminal, call up "filename" with the "vi" program using the video terminal. Once again, you will be editing, so that is why "vi" is used.

You already know how to correct and alter words. This time the words that need changing within the text have to be found. This is done using the [esc]/ command.

If, for instance, the incorrect word "insance" has been given by the "spell" program, then by giving the command

[esc]/insance[cr]

the computer will search through the entire text until it finds that word. The cursor will be positioned on the first letter of the word which can then be corrected. Having corrected the word, it is a good idea to check again through the text to see whether the mistake has occurred anywhere else. This time, instead of typing the whole command [esc]/insance[cr], it is only necessary to type

[esc]/[cr]

because the word is stored in memory until another word is given for the computer to find. If the incorrect word does not appear anywhere else within the text, then the computer will tell you "pattern not found" and you can go on to the next correction.

WARNING: Remember to write all the corrections into the computer (using [esc]:w), otherwise you will still have the incorrect words throughout your text.

PRINTING OUT TEXT

To log onto the printer for printing out the text, the procedure is almost the same as it is for the video terminal, except that you are going to use the "nroff" program instead of the "vi" program.

Most of the work has basically been done through the "vi" program, therefore all you need to know is the command for getting a printout. After logging on and getting the % sign, the "nroff" program command to convert the input of the "vi" program into ordinary text (without tables or equations) is:

% nroff -me filename|col [cr]

The "|col" is a necessary addition to the "nroff" program for our particular printer at Lick Observatory. Without it the pages would get printed higher and higher on each successive page.

WARNING: Don't forget to log off when you have finished printing, otherwise you will be charged for the time, even though you have turned off the power.

Note: "|" = upper case "\" on most video terminals and printers.

SUMMARY OF PROGRAMS USED IN THIS PAPER, PLUS A FEW OTHERS (AND THEIR USES)

In the examples given below, "filename" is used to define a file set up by you; "newfile" is used similarly to define a second file.

% vi filename	"vi" program is used to enter and edit text
% spell filename	checks spelling of text (on printer)
% spell filename|more	checks spelling of text (on video terminal)
% nroff -me filename|col	prints out text (without tables or equations) - for text with equations, see Appendices
% tbl filename|nroff -me|col	prints out text which includes tables (but not equations) - for text or tables with equations, see Appendices
% ls	lists the files in your account alphabetically.
% rm filename	removes "filename" from your file
% cat filename	(FOR PRINTER) shows text as it has been entered in "filename", including all "nroff" commands
% cat filename|more	(FOR VIDEO TERMINAL) as above - hit space bar for next page to appear on screen
% cat filename>newfile	makes a copy of "filename" called "newfile"

MOVING TEXT TO ANOTHER POSITION

[esc]"a6dd	removes 6 lines of text (starting at line with cursor) and puts it into storage buffer "a"
[esc]"a6p	when this command is given, the 6 lines of text stored in buffer "a" will be inserted where needed - position the cursor at the line preceding the line where the text is to be inserted.

PRINTING OUT SPECIFIC PAGES

The following commands can be inserted in all the "nroff" commands. For example, the following command prints page 6 only:

```
% nroff -me -o6 filename|col

OR  % neqn filename|nroff -me -o6 -Tnecsym>newfile
    % necprint newfile
```

Commands for printing out specific pages are:

-o6	prints page 6 only
-o6,9	prints pages 6 and 9 only
-o6-9	prints from page 6 to page 9
-6	prints from beginning of manuscript to page 6
-o6-$	prints from page 6 to end of manuscript

By now you are probably quite comfortable at entering text and getting a good copy printed out. Other manuals that once seemed too technical can be looked at again with more meaning. I hope that is the case. As I said at the beginning, this paper is purely to get you started and this is where it leaves off and the other manuals take over. Appended are examples of some other situations which will probably be needed, including footnotes, tables, and equations. No explanations should be necessary; they are included as an easy reference to save you time searching through other manuals.

ACKNOWLEDGEMENTS

An important aspect of learning word processing is the exchange of ideas between "users". Many of these exchanges have contributed to the production of this paper. Some of the people I would especially like to acknowledge are: Dennie Van Tassel whose classes on text editing at the UCSC Computer Center gave me a good start; Doug Duncan who let me sit by him at the video terminal as he worked on his Ph.D. thesis at Lick Observatory; Martin Gaskell who gave me some good format notes for manuscripts; and Gerri McLellan of the Lick Observatory Publications Office who passed on to me useful knowledge gleaned while we worked along together.

PRINTING OUT PAPERS CONTAINING EQUATIONS

In the Lick Observatory Publications Office we use a "necprint" command to print manuscripts containing equations and/or tables with equations. The "necprint" program reprocesses the output from the "nroff" program so that equations come out right. You must establish a "necprint" file in your account before you can run off equations and/or tables with equations. This is the procedure:

```
% vi necprint

#
onintr fixup
sleep 10                (10 = ten)
stty nl                 ( l = letter el)
cat $argv
fixup:
  stty -nl              ( l = letter el)

% chmod 700 necprint    ("necprint" becomes a program)
```

The sequence of two commands needed to run off a manuscript containing equations is:

```
% neqn filename|nroff -me -Tnecsym> newfile
% necprint newfile
```

Use the following procedure to print out manuscripts containing tables and equations:

```
% neqn filename|tbl|nroff -me -Tnecsym> newfile
% necprint newfile
```

GREEK AND MATH SYMBOLS[1]

The following symbols are available with the "Tech Math" thimble on the NEC Spinwriter printer used at Lick Observatory, University of California, Santa Cruz:

The four characters preceding the Greek and math symbols represent the text which creates them.

\(*a	α	\(pl	+
\(*b	β	\(mi	−
\(*c	ξ	\(ua	↑
\(*d	δ	\(da	↓
\(*e	ε	\(eq	=
\(*f	ϕ	\(sl	/
\(*g	γ	\(>=	≥
\(*h	θ	\(<=	≤
\(*l	λ	\(~=	≃
\(*m	μ	\(ap	~
\(*p	π	\(!=	≠
\(*r	ρ	\(->	→
\(*s	σ	\(<-	←
\(*t	τ	\(lc	⌈
\(*w	ω	\(lf	⌊
\(*y	η	\(rc	⌉
\(*C	Ξ	\(rf	⌋
\(*D	Δ	\(de	°
\(*F	Φ	\(pd	∂
\(*G	Γ	\(gr	∇
\(*H	Θ	\(is	∫
\(*L	Λ	\(pt	∝
\(*P	Π	\(br	|
\(*Q	Ψ	\(mu	x
\(*W	Ω	\(+-	±
		\(if	∞

Note: The semicolon (;) is usually reproduced badly in papers containing equations and/or Greek and math symbols. To overcome this problem, type \z,\u.\d wherever a semicolon is needed.

---

[1] Jim Warner from the Electronics Facility at Natural Sciences II, UCSC, developed the programming which enabled our NEC Spinwriter at Lick Observatory to print out equations, as well as Greek and math symbols. His valuable assistance to Lick Observatory in this matter is greatly appreciated.

EQUATIONS

Here are some examples of equations, followed by the text which created them. Notice that equation (4) is in two parts with the equal signs lined up. Centered equations cannot be lined up in this way, therefore the equation is printed either at the left [by entering .EQ L (4)] or indented [by entering .EQ I (4)]. The words "mark" and "lineup" are used to line up the equations where needed.

Note: All spaces within the equation will be closed up when printed out, therefore a tilde (~) is typed wherever a space is needed in the equation.

The sequence of commands to run off equations is:

```
% neqn filename|nroff -me -Tnecsym>newfile
% necprint newfile
```

$$N(\mu_x,\mu_y) = N_c\ \rho_{x,c}\ \rho_{y,c} + N_f\ \rho_{x,f}\ \rho_{y,f} \qquad (1)$$

```
.EQ (1)
N(\(*m sub x , \(*m sub y ) ~=~ N sub c ~ \(*r sub x,c
~ \(*r sub y,c ~+~ N sub f ~ \(*r sub x,f ~ \(*r sub y,f
.EN
```

$$\rho_{x,c}\rho_{y,c} + \frac{1}{2\pi\ \sigma_x\ \sigma_y}\ \exp\ -\frac{(\mu_x - \mu_{x,o})^2}{2\sigma_x^2}\ \exp\ -\frac{(\mu_y - \mu_{y,o})^2}{2\sigma_y^2} \qquad (2)$$

```
.EQ (2)
\(*r sub x,c \(*r sub y,c ~+~ {1} over {2 \(*p ~ \(*s sub x ~
\(*s sub y} ~~ exp ~ sup - {{(\(*m sub x ~-~ \(*m sub x,o )}sup 2}
over {2 \(*s sub x sup 2} ~~ exp ~ sup - {{(\(*m sub y ~-~ \(*m sub
y,o )}sup 2} over {2 \(*s sub y sup 2}
.EN
```

APPENDIX

$$\iint_{box} \rho_{x,c}\ \rho_{y,c}\ d\mu_x\ d\mu_y \sim \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_{x,c}\ \rho_{y,c}\ d\mu_x\ d\mu_y = 1 \qquad (3)$$

```
.EQ (3)
\(is \(is sub box ~ \(*r sub x,c ~ \(*r sub y,c ~ d \(*m sub x
~ d \(*m sub y ~ \(ap ~ sub -\(if \(is sup \(if ~
sub -\(if \(is sup \(if ~ \(*r sub x,c ~ \(*r sub y,c ~ d \(*m sub x ~
d \(*m sub y ~=~ 1
.EN
```

$$\rho_{x,f} = b + c + d + e$$

$$\rho = f + g + h + j + k \qquad (4)$$

```
.EQ I
\(*r sub x, f mark ~=~ b~+~c~+~d~+~e
.EN
.EQ I (4)
\(*r lineup ~=~ f~+~g~+~h~+~j~+~k
.EN
```

EQUATIONS WITHIN TEXT

In-line equations should be as simple as possible. Most equations can be done satisfactorily using the \u (up) and \d (down) commands for superscripts and subscripts.
[Note: \u causes the text to be printed a half line space up; \d prints a half line space down. Make sure to balance your "ups" and "downs".]

Warning: The format shown for numbered equations above does not necessarily work for in-line equations, e.g., \(*r\dx,c\u is used within the text to produce $\rho_{x,c}$, whereas within an equation it is produced by \(*r sub x,c.

A good manual on the subject is "Typesetting Mathematics - User's Guide" (Second Edition) by Brian W. Kernighan and Lorinda L. Cherry, Bell Laboratories, Murray Hill, New Jersey.

APPENDIX

TABLES

Below is an example of a simple table, followed by the format which created it:

```
% tbl filename|nroff -me|col
```

TABLE I. Plates Measured.

PLATE	EPOCH	EMULSION	FILTER
0.50	1950.17	103a-O	-
E.50	1950.17	103a-E	Red Plexiglass
O-1311	1954.98	103a-O	-
E-1311	1954.98	103a-E	Red Plexiglass

[Note: # = tabulator key]

```
.TS
center;       [centers table]
c s s s       [centers 1st line over 4 columns (c = center;  s = span)]
1 c c c       [2nd line: left justifies 1st column; centers columns 2, 3 & 4]
1 n n c.      [3rd & following lines: left justifies 1st column; lines up
.sp 3                        figures in columns 2 & 3; centers column 4]

TABLE I.  Plates Measured.                          [1st line]
.sp 2
PLATE# EPOCH#  EMULSION#       FILTER               [2nd line]
_
.sp
0.50#    1950.17#103a-O# -                          [3rd and following
E.50#    1950.17#103a-E# Red Plexiglass                   lines]
O-1311# 1954.98#103a-O# -
E-1311# 1954.98#103a-E# Red Plexiglass
.sp
_
.TE
```

APPENDIX

TABLES WITH EQUATIONS

Below is an example of a table with equations, followed by the format which created it:

```
% neqn filename|tbl|nroff -me -Tnecsym>newfile
% necprint newfile
```

NGC 5430 EMISSION LINE FLUXES

(10^{-15} erg cm^{-2} s^{-1})

λ, Å	Line	SE Knot: Observed	Dereddened	Nucleus
3727	[O II]	129	546	22.6
4101	Hδ	19.8	75	---
4650	N IV+C III	25.3	82	---

[Note: # = tabulator key]

```
.TS
center;
c6 s6 s6 s6 s6
c s s s s
c c c s c
c c c c c
n c n n n.
NGC 5430 EMISSION LINE FLUXES
.sp 2
(10\u-15\d erg cm\u-2\d s\u-1\d)
.sp
_
.sp
##                SE Knot: ##
\(*l, \zA\u\uo\d\d#     Line#   Observed#        Dereddened#      Nucleus
_
.sp
3727#   [O II]# 129#    546#    22.6
4101#   H\(*d#  19.8#   75#     ---
4650#   N IV+C III#     25.3#   82#     ---
_
.TE
```

(Table is widened by adding 6 [characters] to each column. Tables can also be narrowed using this method, e.g., cl sl sl sl sl.)

APPENDIX

FOOTNOTES

Below is an example of some footnotes, followed by the format which created them:

```
% neqn filename|nroff -me -Tnecsym>newfile
% necprint newfile
```

LITTLE RED RIDING HOOD[1]

Once upon a time[2] there was a little girl named Little Red Riding Hood. She was a ....

```
.ce
LITTLE RED RIDING HOOD\**
.(f
\**The name has been changed to protect the innocent.
.)f
.pp
Once upon a time\**
.(f
\**Most classics start this way!
.)f
there was a little girl named Little Red Riding Hood.  She was a ....
```

---

[1]The name has been changed to protect the innocent.
[2]Most classics start this way!

APPENDIX

SAMPLE FORMAT FOR A MANUSCRIPT

```
.ll 76
.po 14
.bp
.he ''%''
.na
.nh
.ls 2
.m1 3
.m2 4
.m3 3
.m4 3
```

```
.de pp
.sp 2
.ti 5
.ne 2+\\n(.Vu
..
```

Defines a paragraph and assures that the beginning of the paragraph will be followed by at least two lines of text.

```
.de mh
.sp 3
.ne 12+\\n(.Vu
..
```

Defines main heading (.mh) and assures that 12 line spaces of text will follow it; main heading will begin on next page if there is insufficient space on current page.

```
.de sh
.sp 2
.ne 8+\\n(.Vu
..
```

Defines subheading (.sh) and assures that 8 line spaces of text will follow it; subheading will begin on next page if there is insufficient space on current page.

```
.de re
.in 5
.ti 0
..
```

Defines reference (.re); first line of reference will be exdented to left margin with .ti 0 command, but other lines of reference will be indented 5 spaces.

```
.EQ
delim $$
define ang '\zA\u\uo\d\d'
.EN
```

Defines Å; type ang in the text instead of \zA\u\uo\d\d. Other equations can be defined similarly.

```
.ce
LITTLE RED RIDING HOOD
.sp
.pp
Once upon a time ...
```

Note: \z causes letter following to remain stationary and the next character to be superimposed upon the first.