MicroEMACS Screen Editor Tutorial

June 19, 1985

Mark Williams Company
1430 West Wrightwood Avenue
Chicago, Illinois 60614
Telephone: (312) 472-6659

Printed in U.S.A.

Table of Contents

1. Introduction

This is a tutorial for the COHERENT system's interactive screen editor MicroEMACS.

This tutorial is written for two types of reader: the one who has never used a screen editor and needs a full introduction to the subject, and the one who has used a screen editor before but wishes to review specific topics.

Related documents include the *Introduction to the COHERENT System*, which describes the COHERENT system and introduces many useful programs, *ed Interactive Editor Tutorial*, and *sh Shell Command Language Tutorial*.

Advanced users who are familiar with screen editors may wish to consult entry on MicroEMACS in the COHERENT system's **Command Manual**.

What is MicroEMACS?

MicroEMACS is an interactive screen editor. An editor allows you to type text into your computer, name it, store it, and recall it later for editing. *Interactive* means that MicroEMACS will accept your editing command, execute it, display the result for you immediately, and then wait for your next command. *Screen* means that you can use nearly the entire screen of your terminal as a writing surface: you can move your cursor up, down, and around your screen to create or change text, much as you move your pen up, down, and around a piece of paper.

These features, plus the others that will be described in the course of this tutorial, make MicroEMACS a tool that is powerful, yet easy to use. You can use MicroEMACS to create or change computer programs, essays or letters, electronic mail messages, or any other type of text file.

Text that you create with MicroEMACS can be processed with **nroff**, the COHERENT system's text formatter, to create attractive printed documents. Together, MicroEMACS and **nroff** give you a word processing system that would otherwise cost hundreds of dollars.

The COHERENT system's version of MicroEMACS was adapted from a public-domain program written by David G. Conroy, and this document is based on the descriptions in his essay *MicroEMACS: Reasonable Display Editing in Little Computers*. MicroEMACS is derived from the mainframe display editor EMACS, which was created at the Massachusetts Institute of Technology by Richard Stallman. EMACS is popular among persons who work with computers, and is the parent or grandparent of a number of well-known word processors.

Do the exercises

The following sections have exercises that illustrate each topic being discussed. These exercises will help you understand exactly how each feature works. We recommend that you type in each exercise as you come to it in the text. Even if you understand the concepts being discussed, working the exercises will reinforce the lesson and will help you grow comfortable in using MicroEMACS.

2. Basic Editing

MicroEMACS is the COHERENT system's interactive screen editor.

Interactive means MicroEMACS accepts a command from you, executes it, displays the result on your terminal immediately, and then waits for your next command. In this way, MicroEMACS differs from **sed**, the COHERENT system's noninteractive text editor.

Screen means MicroEMACS allows you to use nearly the entire screen of your terminal as a writing surface. You can move your cursor up, down, and around the screen to enter text or make changes, much as you move your pen up, down, and around a sheet of paper. In this way, MicroEMACS differs from the COHERENT system's line editor **ed**, which, although interactive, does not allow you to touch the text with your cursor in order to make changes.

The first half of this tutorial describes basic editing with MicroEMACS. Mastering the commands described in the next few subsections will allow you to create a document, store it, and edit it thoroughly. Advanced techniques, such as assembling text from several buffers, using windows, and using arguments, will be covered in the second half.

Describing your terminal to MicroEMACS

Before you can begin, you must describe your terminal to MicroEMACS. Although every terminal uses the same computer codes to describe letters of the alphabet, numerals, and symbols such as '&' or '@', each terminal uses its own special codes to move the cursor around the screen. Before you can use MicroEMACS, your computer must know what codes your terminal employs to move its cursor.

Your COHERENT system has built into it the codes for many popular terminals, including those manufactured by Qume, Lear-Siegler, Hewlett-Packard, Zenith, Volker-Craig, Hazeltine, Visual, and Digital. Use the **export** command to tell COHERENT what terminal you are using; **export** uses the following structure:

```
export TERM=terminal
```

Note, however, that the COHERENT system will recognize your terminal *only* if its name is entered properly. Often, that means using an abbreviated form of the name, or entering the name of a different terminal that your terminal mimics. To discover what name you should enter with your **export** command, ask your COHERENT system's administrator to check the file called **/etc/termcap**.

Keystrokes--<esc>, <ctrl>

The MicroEMACS commands use **control** characters and **meta** characters.

Control characters use the *control* key, which is marked **ctrl** on your keyboard; meta characters use the *escape* key, which is marked **esc**.

If you are interested in what these keys actually do, consult the manual *Introduction to the COHERENT System*. To use them with MicroEMACS, however, only requires that you type them correctly.

ctrl is like the **shift** key: hold it down *while* you strike the other key. Here, this will be represented with a hyphen; for example, **control X** will be shown as follows:

```
<ctrl-X>
```

The **esc** key, on the other hand, works like an ordinary character. You should strike it first, *then* strike the letter character you want. *Escape* character codes will not be represented with a hyphen; for example, **escape X** will be represented as:

```
<esc>X
```

Becoming acquainted with MicroEMACS

Now that you have used the **export** command to describe your terminal to MicroEMACS, you are ready for a few simple exercises that will help you get a feel for how MicroEMACS works.

To begin, type the following command to the COHERENT system:

```
me sample
```

Within a few seconds, your screen will have been cleared of writing, the cursor will be positioned in the upper left-hand corner of the screen, and a command line will appear at the bottom of your screen.

Type the following text. You will need to type a carriage return at the end of each line:

```
There is nothing which has yet been contrived by
man, by which so much happiness is produced as by
a good tavern or inn.
```

Notice how the text appeared on the screen character by character as you typed it, much as it would appear on a piece of paper if you were using a typewriter.

Now, type <ctrl-X><ctrl-S>; that is type <ctrl-X>, and then type <ctrl-S>. It does not matter whether you type capital or lower-case letters. Notice that a message has appeared at the bottom of your screen. This command has permanently stored, or *saved*, what you typed. This text is now preserved until you give the proper COHERENT command to erase it.

Type the next few commands, which demonstrate some of the tasks that MicroEMACS can perform for you. These commands will be explained in full in the sections that follow; for now, it is enough for you to get a feel for how MicroEMACS works.

Type <esc><. Be sure that you type a less-than symbol '<'. Notice that the cursor has returned to the upper left-hand corner of the screen. Type <esc>F. The cursor has jumped forward by one word, and is now between the words There and is. Type <ctrl-N>. Notice that the cursor has jumped to the next line, and is now under the letter b of the word by. Type <ctrl-A>. The cursor has jumped to the *beginning* of the second line of your text.

Now, type <ctrl-K>. The second line of text has disappeared, leaving an empty space. Type <ctrl-K> again. The empty space where the second line of text had been has now disappeared.

Type <esc>>. Be sure to type a greater-than symbol '>'. The cursor has jumped to the space just below the last line of text. Now type <ctrl-Y>. The text that you erased a moment ago has now been restored.

By now, you should be feeling more at ease with typing MicroEMACS's *control* and *escape* codes. The following sections will explain what these commands mean. For now, exit from MicroEMACS by typing <ctrl-X><ctrl-C>; when the message

```
Quit [y/n]?
```

appears, type y. This will return you to the COHERENT system.

Before you begin

There are one or two potential sources of difficulty that you should watch for as you begin to work with MicroEMACS. As you begin a file, remember that MicroEMACS can handle only small files efficiently. MicroEMACS's operation may visibly slow, and it may eventually stop if your text is too large--that is, more than approximately 500 lines of text.

If your file proves to be too large, either when it is loaded or while you are working on it, you will see the following message:

```
Cannot allocate XX bytes
```

where *XX* indicates the number of bytes that MicroEMACS cannot allocate. When this happens, you should exit from MicroEMACS to the COHERENT system, by typing <ctrl-X><ctrl-C>; using any other command in an attempt to save the changes you made to your text could damage your original file severely. Then use the split command or your line editor to break the file into several parts. Exiting to the COHERENT system will be covered in the next few pages.

If you are working a large text, you should break it into several portions with the interactive line editor **ed**, and edit each portion with MicroEMACS separately. If you do not know how to work with **ed**, consult the *ed Interactive Editor Tutorial* that accompanies your COHERENT system's documentation.

Three sample texts have been included with MicroEMACS. They are called **text1.m**, **text2.m**, and **text3.m**. Before you begin, you should make working copies of these texts, so that if an accident were to occur while you were working on this tutorial the master copies of the sample texts would be preserved.

Make your working copies by typing the following commands into your computer:

```
cp text1.m text1
cp text2.m text2
cp text3.m text3
```

When you first invoke MicroEMACS, your computer will take a moment to set it into operation. If you are going to edit a large text, MicroEMACS may take a few seconds to load it into memory.

You will know MicroEMACS set up and ready to go when the following message appears at the bottom of your screen:

```
[Read XX lines]
```

where *XX* stands for the number of lines in your text file. If you are creating a new text file, MicroEMACS will send you this message:

```
[New file]
```

The next point is extremely important. While the MicroEMACS program is being set up--that is, the time between when you type the **me** command and when the **Read** message appears at the bottom of your screen--your computer is changing the way it interprets *control* characters and *escape* characters. If you type a *control* or an *escape* character during this time, your computer may begin to interpret it the COHERENT way, then shift to interpreting it the MicroEMACS. Your terminal may become so badly jammed that your computer system will have to be rebooted.

Therefore, after you type in the **me** command and before the **Read** message appears at the bottom of your screen, *do not touch your keyboard!*

Beginning a document

You are now ready to invoke MicroEMACS and create a text file. Type the following command line, which tells MicroEMACS that you wish to edit the text called **text1**:

```
me text1
```

This text has been included with your COHERENT system; there is no need to retype it.

The computer will take a moment to set up the MicroEMACS program. As soon as it does so, the following text will appear on your screen:

```
From "Life on the Mississippi":
I know how a prize watermelon looks when it is sunning
its fat rotundity among the pumpkin vines; I know how to tell
when it is ripe without "plugging" it; I know how inviting
it looks when it is cooling itself in a tub of water under
the bed, waiting; I know how it looks when it lies on the
table in the sheltered great floor space between house and
kitchen, and the children gathered for the sacrifice and
their mouths watering; I know the crackling sound it makes
when the carving knife enters its end, and I can see the
split fly along in front of the blade as the knife cleaves
its way to the other end; I can see its halves fall apart
and display the rich red meat and the black seeds, and the
heart standing up, a luxury fit the elect; I know how a
boy looks behind a yard-long slice of that melon, and I
know how he feels; for I have been there.
```

When you type the MicroEMACS command and a file name, MicroEMACS *copies* that text file into a special area in your computer to make it available for editing. If you were creating a new text, as you did earlier with the text called sample, the screen would have appeared blank.

In addition to this text appearing on your screen, your cursor will have moved to the upper left-hand corner of the screen, and the status line will appear near the bottom of your screen as follows:

```
-- MicroEMACS -- text1 -- File: text1 -------------------------
```

The word to the left, MicroEMACS, is the name of the program. The word in the center, text1, is the name of the *buffer* that you are using. (What a buffer is and how it is used will be covered later.) The name to the right is the name of the text file that you will be editing.

3. Moving the Cursor

Now that you have created a text file, you will want to edit it. The first step is to learn to move the cursor. Try out these commands for yourself as they are described in the following pages. That way, you will quickly acquire a feel for handling MicroEMACS's commands. With most computers MicroEMACS will not understand your terminal's arrow keys; therefore, you should memorize these cursor movement commands as quickly as possible, in order to become fluent in using MicroEMACS.

The following display shows the basic cursor movement commands.

<ctrl-B> Move back 1 space
<esc>B Move back 1 word
<ctrl-E> Move to end of line

<ctrl-F> Move forward 1 space
<esc>F Move forward 1 word
<ctrl-A> Move to beginning of line

<ctrl-P> Move to previous line
<ctrl-N> Move to next line

<ctrl-V> Move forward 1 screen
<esc>V Move back 1 screen

<esc>< Move to beginning of text
<esc>> Move to end of text

Moving the cursor backwards

The first set of commands move the cursor backwards. First, type the *end of text* command <esc>> to move the cursor to the bottom of the text. Be sure to type a greater-than symbol '>'.

Type the *backspace* command <ctrl-B>. As before, it does not matter whether the letter is upper case or lower case. Note that the cursor is now located just to the right of the period in your last line of text. Type <ctrl-B> again. The cursor has moved one space to the left, and now is directly under the period.

Type <esc>B. The cursor has moved one *word* to the left, and is now under the letter t of the word there. Type the *beginning of line* command <ctrl-A>. The cursor has jumped to the beginning of the line, and is now under the letter k of the word know.

Moving the cursor forwards

Now practice moving the cursor forwards. Type the *forward* command <ctrl-F>. Note that the cursor has moved one space to the right, and now is under the letter **n** of the word **know**. Type <esc>F. The cursor has moved one *word* to the right, and now is under the space between the words **know** and **how**.

Type the *end of line* command <ctrl-E>. The cursor has jumped to the end of the line, and once again is resting to the right of the period.

From line to line

The next two commands move the cursor up and down the screen. Type the *previous line* command <ctrl-P>. Note that the cursor has jumped from its position to the right of the period on the last line of your text, to being under the second letter **t** of the word **that** in the previous line.

Continue to type <ctrl-P> until the cursor reaches the top of the screen. Note that as you reached the first line in your text, the cursor jumped from under the letter **i** of the word **it** on the second line, to being just right of the colon on the first line of text. When you move your cursor up or down the screen, MicroEMACS will try to keep it at the same position within each line. If the line to which you are moving the cursor is not long enough to have a character at that position, MicroEMACS will move the cursor to the end of the line.

Now, practice moving the cursor back down the screen. Type the *next line* command <ctrl-N>. Note that when the cursor jumped to the next line, it returned to under the letter **i** of the word **it**. MicroEMACS remembered the cursor's position on the line, and returned the cursor there when it jumped to a line long enough to have a character in that position.

Continue pressing <ctrl-N>. The cursor will move down the screen, until it reaches the bottom of your text.

Repetitive motion

Some computers will repeat a command automatically if you simply *hold down* the control key and the character key. Try holding down <ctrl-P> for a moment, and see if it repeats automatically. If it does, that will speed moving your cursor around the screen, because you will not have to type the same command repeatedly.

Moving up and down by a screenful of text

The next two cursor movement commands allow you to roll forward or backwards by one screenful of text.

If you are editing a file with MicroEMACS that is too big to be displayed on your screen all at once, MicroEMACS will display the file in screen-sized portions (on most terminals, 22 lines at a time). The *view* commands **<ctrl-V>** and **<esc>V** allow you to roll up or down by one screenful of text at a time.

Type **<ctrl-V>**. Note that your screen becomes empty. This is because you have rolled forward by the equivalent of one screenful of text, or 22 lines; however, because the text in this example is only 16 lines long, rolling forward 22 lines just empties the screen.

Now, type **<esc>V**. Notice that your text rolls back onto the screen, and your cursor is positioned in the upper left-hand corner of the screen, under the letter **F** of the word **From**.

Moving to beginning or end of text

The last two cursor movement commands allow you to jump immediately to the beginning or end of your text.

The *end of text* command **<esc>>** moves the cursor to the end of your text. Type **<esc>>**. Be sure to type a greater-than symbol '>'; this symbol may have been placed nearly anywhere on your keyboard, although on IBM-style keyboards it appears above the period. Note that your cursor has jumped to the end of your text.

The *beginning of text* command **<esc><** will move the cursor back to the beginning of your text. Type **<esc><**. Be sure to type a less-than symbol '<'; on IBM-style keyboards it appears above the comma. Note that the cursor has jumped back to the upper left-hand corner of your screen.

These commands will move you immediately to the beginning or the end of your text, regardless of whether the text is one page long or 20.

Cursor movement strategy

When you edit a large text, you will have to move the cursor often. This can be very time-consuming. Two rules will help you save time by moving the cursor efficiently.

1. Plan your cursor movements so that you reach your target with the fewest keystrokes possible.

2. If you are a good typist, avoid using the **<esc>** key if possible, because using the **<esc>** key usually forces you to lift your left hand from the home position.

Try the following exercises to sharpen your command of cursor movement. Each exercise is followed by its solution. Be sure not to look at the solution until you have at least attempted to solve the problem. The exercises should be done in order, because each one builds on the ones that come before.

1. Your cursor should be in the upper left-hand corner of the screen. If it is not, type <esc><. Now, move the cursor to the space just before the word **children** in line 8--you should be able to do it with ten commands.

Solution: Type <ctrl-N> seven times, then type <esc>F three times.

2. Move the cursor to under the letter **n** of the word **knife** in line 10--you should be able to do it with four commands.

Solution: Type <ctrl-N> twice, then <ctrl-F> twice.

3. Move the cursor to the right of the period on line 16--you should be able to do it with two commands.

Solution: Type <esc>>, then type <ctrl-B>.

4. Move the cursor to the space after the word **fall** on line 12--you should be able to do it with six commands.

Solution: Type <ctrl-P> four times, then type <esc>F twice.

5. Move the cursor to under the letter **k** of the word **kitchen** in line 8--you should be able to do it with five commands.

Solution: Type <ctrl-A>, then type <ctrl-P> four times.

6. Finally, move the cursor to under the letter M of the word **Mississippi** on line 1--you should be able to do it with three commands.

Solution: Type <esc><, type <ctrl-E>, then type <esc>B.

Saving text and quitting

If you do not wish to continue working at this time, you should *save* your text, and then *quit*.

It is good practice to save your text file every so often while you are working on it; then, if an accident occurs, such as a power failure, you will not lose all of your work. You can save your text with the *save* command <ctrl-X><ctrl-S>. Type <ctrl-X><ctrl-S>--that is, first type <ctrl-X>, then type <ctrl-S>. Note that at the bottom of your screen the following message has appeared:

```
[Wrote 16 lines]
```

The text file has again been saved to your computer's memory. Note, too, that MicroEMACS will send you messages from time to time; the messages enclosed in square brackets '[' ']' are for your information, and do not mean that something is wrong. To exit from MicroEMACS, type the *quit* command <ctrl-X><ctrl-C>. This will return you to the COHERENT system.

4. Killing and Deleting

Now that you know how to move the cursor, you are ready to edit your text. To return to MicroEMACS, type the command:

```
me text1
```

Within a moment, **text1** will have been restored to your screen.

By now, you have noticed that MicroEMACS is always ready to insert material into your text; unless you type the **<ctrl>** or **<esc>** keys, MicroEMACS will assume that whatever you type is meant to be text and will insert it onto your screen where your cursor is positioned.

The simplest way to erase text is simply to position the cursor to the right of the text you want to erase and backspace over it. MicroEMACS, however, has a set of commands that allow you to erase large amounts of text easily. These commands *kill* and *delete*; the distinction is important, and will be explained in a moment. The following display summarizes these commands:

<ctrl-D>	Delete 1 character to the right
<esc>D	Kill 1 word to the right
<del>	Delete 1 character to the left
<esc><del>	Kill 1 word to the left
<ctrl-K>	Kill rest of line
<ctrl-Y>	Yank back (restore) killed text

Deleting versus killing

It is important to distinguish between killing and deleting. When text is *deleted*, it is erased completely from memory; however, when text is *killed*, it is moved into a temporary storage area elsewhere in the computer. This storage area is erased when you move the cursor and then kill additional text. Until then, however, the killed text is saved. This aspect of killing allows you to restore text that you killed accidentally, and it also allows you to move or copy portions of text from one position to another.

Erasing text to the right

The first two commands to be presented erase text to the *right*.

Type the *delete* command **<ctrl-D>**. Note that the letter **F** of the word **From** has been erased, and the rest of the line has shifted one space to the left.

Now, type **<esc>D**. The rest of the word **From** has been erased, and the line has shifted three spaces to the left. Note that the cursor is positioned under the *space* before the word **"Life**. Type **<esc>D** again. The word **"Life** has vanished along with the *space* that preceded it, and the line has shifted *six* spaces to the left.

Note that **<ctrl-D>** *deletes* text, but **<esc>D** *kills* text.

MicroEMACS is designed so that when it erases text, it does so beginning at a point immediately to the *left* of the cursor. Therefore, if you wish to erase a word but wish to keep both spaces around it, position your cursor directly under the first character of the word and strike **<esc>D**. If you wish to erase a word *and* the space before it, position the cursor under the space before you strike **<esc>D**.

Erasing text to the left

You can erase text to the *left* by using the delete key **<del>**. This key is usually located in the upper right-hand corner of your keyboard. Occasionally, it is found just below the keypad on the right side of your keyboard.

Be sure to note where it is, because it is most useful.

If your keyboard's **<del>** key does not operate in the way described below, you can also erase text to the left with the command **<ctrl-H>**.

To see how to erase text to the left, first type the *end of line* command **<ctrl-E>** to move the cursor to the right of the colon on the first line of text. Type **<del>**. Note that the colon has vanished.

Type **<esc><del>**. The word **Mississippi"** has disappeared, and the cursor has moved to the second space following the word **the.**

Move the cursor four spaces to the left, so that it is under the letter **t** of the word **the.** Type **<esc><del>**. The word **on** has vanished, along with the space that was immediately to the right of it. As before, these commands erased text beginning immediately to the *left* of the cursor. The **<esc><del>** command can be used to erase words throughout your text.

If you wish to erase a word to the left yet preserve both spaces that are around it, position the cursor under the space immediately to the right of the word and strike **<esc><del>**. If you wish to erase a word to the left plus the space that immediately follows it, position the cursor under the first letter of the *next* word and then strike **<esc><del>**.

Note that typing **<del>** *deletes* text, but typing **<esc><del>** *kills* text.

Erasing lines of text

Finally, the following command erases a lines of text: the *kill* command **<ctrl-K>**. This command erases a line of text, beginning from immediately to the left of the cursor.

To see how this works, move the cursor to the beginning of line 2. Now, strike <ctrl-K>. Note that all of line 2 has vanished, and been replaced with an empty space. Strike <ctrl-K> again. Note that the empty space has vanished, and the cursor is now positioned at the beginning of what used to be line 3, under the letter i of the word

As its name implies, the <ctrl-K> command *kills* the line of text.

Yanking back (restoring) text

Remember that when material is killed, MicroEMACS has temporarily stored it elsewhere. Thus, text that has been killed can be returned to the screen by using the *yank back* command <ctrl-Y>. Type <ctrl-Y>. Note that all of line 2 is now returned; the cursor, however, remains under the letter i of its in line 3.

Killing and deleting--exercises

To fix these distinctions in your mind, perform the next few exercises. Work the exercises in order, as each exercise builds on the ones that came before it, and try not to look at the solution until you have at least tried to solve the problem.

Before you begin, move your cursor back to the upper left-hand corner of your screen by typing <esc><.

1. Erase the word **sheltered** in line 7 and the space that follows it.

Solution: To move the cursor to the correct position, type <ctrl-N> six times; type <esc>F four times; and type <ctrl-F> once. Then type <esc><del>.

2. Erase the word **children** in line 8, and the space that precedes it.

Solution: To move the cursor to the correct position, type <ctrl-N>, then type <esc>F. Type <esc>D.

3. Erase line 4.

Solution: To move cursor, type <ctrl-P> four times, then <ctrl-A>. Type <ctrl-K>.

4. Yank back line 4.

Solution: Type <ctrl-Y>.

Quitting

When you are finished, do not save the text. If you do so, the undamaged copy of the text that you made earlier will be replaced with the present damaged copy. Rather, use the *quit* command <ctrl-X><ctrl-C>. Type <ctrl-X><ctrl-C>. On the bottom of your screen, MicroEMACS will respond:

```
Quit [y/n]?
```

Reply by typing y and a carriage return. If you type n, MicroEMACS will simply

return you to where you were in the text. MicroEMACS will then return you to the COHERENT system.

5. Block Killing and Moving Text

As noted above, text that is killed is stored temporarily within the computer. Killed text, however, may be yanked back onto your screen, and not necessarily in the spot where it was originally killed. This feature allows you to move text from one position to another.

The following table summarizes the commands used to kill a block of text and move it:

<ctrl-K> Kill text to end of line

<ctrl-@> Set mark
<ctrl-W> Kill block of text

<ctrl-Y> Yank back text

Moving one line of text

To test these commands, invoke MicroEMACS for the text **text1** by typing the following command:

```
me text1
```

When MicroEMACS appears, the cursor will be positioned in the upper left-hand corner of the screen.

To move the first line of text, begin by typing the *kill* command <ctrl-K> twice. Now, press <esc>>, to move the cursor to the bottom of text. Finally, yank back the line by typing <ctrl-Y>. The line that reads

```
From "Life on the Mississippi":
```

is now at the bottom of your text.

Note that your cursor has moved to the beginning of the blank line *after* the line you yanked back.

Multiple copying of killed text

When text is yanked back onto your screen, it is *not* deleted from the computer. Rather, it is simply *copied* back onto the screen. This means that killed text can be reinserted into the text more than once. To see how this is done, return to the top of the text by typing <esc><. Then type <ctrl-Y>. The line you just killed now appears twice on your screen.

Note that the killed text will not be erased from its temporary storage until you move the cursor and then kill additional text. If you kill several lines or portions of lines in a row, all of the killed text will be stored in the buffer; if you are not careful, you may yank back a jumble of accumulated text.

Kill and move a block of text

If you wish to kill a block of text, you can either type the *kill* command **<ctrl-K>** repeatedly to kill the block one line at a time, or you can use the *block kill* command **<ctrl-W>**. To use this command, you must first set a *mark* on the screen, an invisible character that acts as a guidepost to the computer. The mark is set with the *mark* command **<ctrl-@>**.

Once the mark is set, you must move your cursor to the other end of the block of text you wish to kill, and then strike **<ctrl-W>**. The block of text will be erased, and will be ready to be yanked back elsewhere.

Try this out on **text1**. Type **<esc><** to move the cursor to the upper left-hand corner of the screen. Then type the *set mark* command **<ctrl-@>**. By the way, be sure to type a '@', not a '2'. MicroEMACS will respond with the message

```
[Mark set]
```

at the bottom of your screen. Now, move the cursor down four lines, and type **<ctrl-W>**. Note how the block of text you marked out has disappeared.

Move the cursor to the bottom of your text. Type **<ctrl-Y>**. The killed block of text has now been reinserted.

When you yank back text, be sure to position the cursor at the *beginning* of the line *below* where you want the text to be yanked back. This will ensure that the text will be yanked back in the proper place, and the cursor will not be moved when the text reappears.

To try this out, move your cursor up four lines. Be careful that the cursor is at the *beginning* of the line. Now, type **<ctrl-Y>** again. Note that the text reappeared *above* where the cursor was positioned, and that the cursor was not moved from its position at the beginning of the line--which is not what would have happened had you positioned it in the middle or at the end of a line.

Although the text you are working with has only 16 lines, you can move much larger portions of text, using only these three commands. Remember, too, that you can use this technique to duplicate large portions of text at several positions, to save yourself considerable time in typing and reduce the number of possible typographical errors.

6. Capitalization, Transposing, and Screen Redraw

The next commands perform a number of useful tasks that will help with your editing. They are as follows:

<esc>C Capitalize a word
<esc>L Lowercase a word
<esc>U Uppercase a word

<ctrl-T> Transpose characters

<ctrl-L> Redraw screen

Before you begin this section, destroy the old text on your screen with the *quit* command <ctrl-X><ctrl-C>, and read into MicroEMACS a fresh copy of the text, as you did earlier.

Capitalization and lowercasing

MicroEMACS has several commands that can automatically capitalize words or make them all upper case or lower case.

Move the cursor to the letter **w** of the word **watermelon** on line 2. Type the *capitalize* command <esc>C. The word is now capitalized, and the cursor is now positioned under the space after it. Move the cursor back so that it is under the letter **m** in **Watermelon**. Press <esc>C again. Note that the word is now spelled **WaterMelon**. When you press <esc>C, MicroEMACS will capitalize the *first* letter the cursor meets.

MicroEMACS can also change a word to all upper case or all lower case. (There is very little need for a command that will change only the first character of an upper-case word to lower case, so it is not included.)

Move the cursor so that it is again to the left of the word **WaterMelon**. It does not matter if the cursor is directly under **W** or under the space to its left; therefore, you can capitalize or lowercase a number of words in a row without having to move the cursor.

Type the *uppercase* command <esc>U. The word is now spelled WATERMELON, and the cursor has jumped to the space after the word.

Again, move the cursor to the left of the word WATERMELON. Type the *lowercase* command <esc>L. The word has changed back to **watermelon**. Now, move the cursor to the left of the word "**Life** on line 1. Type <esc>L once again. Notice that the quotation mark is not affected by the command, but the letter **L** is now lower case. <esc>L not only shifts a word that is all upper case to lower case: it can also un-capitalize a word.

Note that the *uppercase* and *lowercase* commands will stop at the first point of punctuation they encounter after the first letter they find; therefore, if you wish to change the case of a word with an apostrophe in it, you must type the appropriate command twice.

Transpose characters

MicroEMACS allows you to reverse the position of two characters, or *transpose* them, with the *transpose* command **<ctrl-T>**.

Move the cursor to the middle of a line and type **<ctrl-T>**. Note that the character under which the cursor was positioned has been transposed with the character immediately to its *left*. Type **<ctrl-T>** again. The characters are now returned to their original order.

Screen redraw

Occasionally, while you are working on a text another COHERENT user will write or mail you a message. The COHERENT system will write the message directly on your screen, which scrambles your screen. Note that a message sent from another user or a message from the COHERENT system will *not* be recorded into your text; however, you may wish to erase the message and continue editing.

The *redraw screen* command **<ctrl-L>** will redraw your screen to the way it was before the extraneous material was written onto it.

Type **<ctrl-L>**. Notice how the screen flickers and the text is rewritten. Had your screen been spoiled by extraneous material, that material would have been erased and the original text rewritten.

7. Search and Reverse Search

When you edit a large text, you may wish to change particular words or phrases. To do this, you can roll through the text and read each line to find them; or you can have MicroEMACS find them for you. The following display summarizes the MicroEMACS search commands:

<ctrl-S> Search forward
<ctrl-R> Search backwards
<ctrl-G> Cancel a search command

Search forward

To begin, type the *beginning of text* command <esc>[<] to move the cursor to the upper left-hand corner of your screen. Now, type the *search* command <ctrl-S>. MicroEMACS will respond by prompting with the message

```
Search:
```

at the bottom of the screen.

Type in the words **been there**, then press the carriage return. Notice that the cursor has jumped to the period after the word **there** in the last line of your text. MicroEMACS searched for the words **been there**, found them, and moved the cursor there.

If the word you were searching for was not in your text, or at least was not in the portion that lies between your cursor and the end of the text, MicroEMACS would not have moved the cursor, and would have displayed the message

```
Not found
```

at the bottom of your screen.

Reverse search

The *search* command <ctrl-S>, useful as it is, can only search forward through your text. To search backwards, use the *reverse search* command <ctrl-R>. Type <ctrl-R>. MicroEMACS will reply with the message

```
Reverse search [been there]:
```

at the bottom of your screen. The words in square brackets are the words you entered earlier for the *search* command; MicroEMACS remembered them. If you wanted to search for **been there** again, you would just press the carriage return. For now, however, type the word **watermelon** and press the carriage return.

Notice that the cursor has jumped so that it is under the letter **w** of the word **watermelon** in line 2. When you search forward, the cursor will move to the *space after* the word you are searching for, whereas when you reverse search the cursor will be moved to the *first letter* of the word you are searching for.

Search for portions of words

You do not have to search for an entire word; if you wish, you can search for a portion of a word or even a single letter of the alphabet. Note, however, that the *search* and *reverse search* commands do *not* distinguish between upper-case and lower-case letters--if you ask MicroEMACS to search for the letter 'I', it will stop at every occurrence of 'i' as well.

Type **<ctrl-S>**; when MicroEMACS asks what to search for, type **melon**. Note that the cursor jumps to the end of the word **watermelon**. Type **<ctrl-S>** again, then a carriage return. Your cursor now jumps to the comma after the word **melon** in the next-to-last line of the text. Now type **<ctrl-R>**. The prompt will appear as follows:

```
Reverse search [melon]:
```

Type a carriage return. The cursor jumped to the beginning of the word **melon** on line 15. Type **<ctrl-R>** again, then a carriage return. The cursor now has jumped to the letter **m** in **watermelon** on line 2.

When MicroEMACS searches, it does not distinguish between whole words and portions of words; when you ask it to search for **melon**, it will find **melon** whether it is a word by itself or simply part of another word.

Cancel a command

The commands presented earlier to move the cursor or to delete or kill text all execute immediately. Although this speeds your editing, it also means that if you type a command by mistake, it executes before you can stop it.

The *search* and *reverse search* commands, however, wait for you to respond to their prompts before they execute. If you type **<ctrl-S>** or **<ctrl-R>** by accident, MicroEMACS will interrupt your editing and wait patiently for you to initate a search that you do not want to perform. You can evade this problem, however, with the *cancel* command **<ctrl-G>**. This command tells MicroEMACS to ignore the previous command.

To see how this command works, type **<ctrl-R>**. When the prompt appears at the bottom of your screen, type **<ctrl-G>**. Three things happen: your terminal beeps, the characters ^G appears at the bottom of your screen, and the cursor returns to where it in your text was before you first typed **<ctrl-R>**. The **<ctrl-R>** command has been cancelled, and you are free to continue editing.

8. Saving Text and Exiting

The last set of basic editing commands allow you to save your text and exit from the MicroEMACS program. They are as follows:

<ctrl-X><ctrl-S> Save text
<ctrl-X><ctrl-W> Write text to a new file

<ctrl-Z> Save text and exit
<ctrl-X><ctrl-C> Exit without saving text

You have used two of these commands already: the *save* command <ctrl-X><ctrl-S> and the *quit* command <ctrl-X><ctrl-C>, which respectively allow you to save text or to exit from MicroEMACS without saving text. (Commands that begin with <ctrl-X> are called *extended* commands; they are used frequently in the advanced editing to be covered in the second half of this tutorial.)

Write text to a new file

If you wish, you may copy the text you are currently editing to a text file other than the one from which you originally took the text. Do this with the *write* command <ctrl-X><ctrl-W>.

To test this command, type <ctrl-X><ctrl-W>. MicroEMACS will display the following message on the bottom of your screen:

```
Write file:
```

MicroEMACS is asking for the name of the file to which you want to write the text. Type **twain**. MicroEMACS will reply:

```
[Wrote 16 lines]
```

The 16 lines of your text have been copied to a new file, called **twain**. Note that the status line at the bottom of your screen has changed to read as follows:

```
-- MicroEMACS -- text1 -- File: twain --------------------------
```

The significance of the change in file name will be discussed in the second half of this tutorial.

Before you copy text to a new file, be sure that you have not selected a file name that is already being used. If you do, whatever is being stored under that file name will be erased, and the text created with MicroEMACS will be stored in its place.

Save text and exit

Finally, the *store* command **<ctrl-Z>** will save your text *and* move you out of the MicroEMACS program. To see how this works, watch the bottom line of your terminal carefully and type **<ctrl-Z>**. If you watched carefully, you would have seen that the message

```
[Wrote 16 lines]
```

flickered on your screen, then the COHERENT system's prompt appeared. MicroEMACS has saved your text, and now you can issue commands directly to the COHERENT system.

9. Basic Editing--Conclusion and Summary

This concludes the presentation of MicroEMACS's basic commands. The second half of this tutorial will introduce you to the advanced features of the MicroEMACS interactive screen editor.

This section introduced the basics of using an interactive screen editor, and presented the basic MicroEMACS editing commands.

If you have mastered the commands and techniques in the first half of this tutorial, you may have no need to work any further, because you now can create a file, edit it, store it, and recall it for further editing.

The tutorial gives instructions on how to invoke MicroEMACS, how to name a text file, and the meaning of the information in the MicroEMACS command line. Remember not to touch the keyboard while MicroEMACS is starting up.

An exercise text is presented, and instructions on how to type in the text and save it are given.

A number of commands can be used to move the cursor around the screen. **<ctrl-F>** and **<esc>F** move the cursor forward on the line. **<ctrl-B>** and **<esc>B** move the cursor backwards on the line. **<ctrl-P>** and **<ctrl-N>** move the cursor to the previous or next lines, respectively. **<ctrl-A>** and **<ctrl-E>** move the cursor to the beginning the end of the line, respectively. **<esc><** and **<esc>>** move the cursor to the beginning or end of the text, respectively. **<ctrl-V>** and **<esc>V** roll the screen forwards or backwards, respectively.

You can erase text in a number of ways. **<ctrl-D>** and **<esc>D** erase text to the right. **<del>** and **<esc><del>** erase text to the left. **<ctrl-K>** erases a line of text (or a portion of a line, should the cursor be positioned in the middle of line). **<ctrl-D** and **<del>** *delete* text, whereas **<esc>D**, **<esc><del>**, and **<ctrl-K>** *kill* text. **<ctrl-Y>** yanks back killed text.

Text can be block killed and moved from one part of your text to another. To mark a block of text for killing, first type **<ctrl-@>** at one end, then move the cursor to the other end. Typing **<ctrl-W>** kills the marked block of text, and **<ctrl-Y>** yanks back killed text.

The following commands allow the user to block-kill and move text. **<ctrl-@>** sets a mark; **<ctrl-W>** deletes all text between the mark and the cursor. **<ctrl-Y>** yanks back the block-killed text wherever the cursor is positioned.

Specific commands capitalize, uppercase, and lowercase words: **<esc>C** capitalizes a word; **<esc>L** lowercases a word; and **<esc>U** uppercases a word. **<ctrl-T>** allows you to transpose characters automatically, and **<ctrl-L>** redraws a scrambled screen.

Words or parts of words can be searched for either forwards or backwards through the text: **<ctrl-S>** searches forward, and **<ctrl-R>** searches backwards. **<ctrl-G>** cancels these commands.

Finally, **<ctrl-X><ctrl-S>** saves text to the file named on the command line; **<ctrl-X><ctrl-C>** allows the user to exit from MicroEMACS without saving text; and **<ctrl-Z>** saves text and moves you back into the COHERENT system.

10. Advanced Editing

The second half of this tutorial will introduce you to the advanced features of MicroEMACS interactive screen editor.

The techniques described here will help you execute complex editing tasks with minimal trouble. You will be able to edit more than one text at a time, display more than one text on your screen at a time, enter a long or complicated phrase repeatedly with only one keystroke, and give commands to the COHERENT system without having to exit from MicroEMACS.

Before beginning, however, you must prepare a new text file--you were probably a little tired of watermelon by now, anyway. Type the following command to the COHERENT system:

```
me text2
```

This text has been included on the diskettes that contained the COHERENT operating system; there is no need to retype it. Within a moment, **text2** will appear on your screen, as follows:

```
From the "Devil's Dictionary":
A penny saved is a penny to squander.
A man is known by the company he organizes.
A bird in the hand is worth what it will bring.
Think twice before you speak to a friend in need.
He laughs best who laughs least.
Least said is soonest disavowed.
Speak of the Devil and he will hear about it.
Of two evils choose to be the least.
Strike while your employer has a big contract.
Where there's a will there's a won't.
```

11. Arguments

Most of the commands described in the first part of this tutorial can be used with *arguments*. An argument is a subcommand that tells MicroEMACS to execute a command repeatedly. With MicroEMACS, arguments are introduced by striking <ctrl-U>.

Arguments--default values

By itself, <ctrl-U> sets the argument at *four*. To test this, first type the *next line* command <ctrl-N>. By itself, this command moves the cursor down one line, from being under the F in the word From on line 1, to being under the A at the beginning of line 2.

Now, type <ctrl-U>. Note that MicroEMACS replies with the message:

```
Arg: 4
```

Now type <ctrl-N>. The cursor jumps down *four* lines, from the letter A in line 2 to the letter H of the word He at the beginning of line 6.

Type <ctrl-U>. The line at the bottom of the screen again shows that the value of the argument is 4. Type <ctrl-U> again. Now the line at the bottom of the screen reads:

```
Arg: 16
```

Type <ctrl-U> once more. The line at the bottom of the screen now reads:

```
Arg: 64
```

Each time you type <ctrl-U>, the value of the argument is *multiplied* by four. Type the *forward* command <ctrl-F>. The cursor has jumped ahead 64 characters, and is now under the period at the end of line 7.

Selecting values

Naturally, arguments do not have to be powers of four. You can set the argument to whatever number you wish, simply by typing <ctrl-U> and then typing in the number you want.

For example, type <ctrl-U>, and then type 3. The line at the bottom of the screen now reads:

```
Arg: 3
```

Type the *delete* command <esc><del>. MicroEMACS has deleted three words to the left.

Arguments can be used to increase the power of any *cursor movement* command, or any *kill* or *delete* command (with the sole exception of **<ctrl-W>**, the *block kill* command).

Deleting with arguments--an exception

Killing and *deleting* were described in the first part of this tutorial. They were said to differ in that text that was killed was stored in a special area of the computer and could be yanked back, whereas text that was deleted was erased outright. However, there is one exception to this rule: any text that is deleted using an argument can also be yanked back.

Move the cursor to the upper left-hand corner of the screen by typing the *begin text* command **<esc><**. Then, type **<ctrl-U><ctrl-D>**. Note that the word **From** has disappeared. Move the cursor to the right until it is between the words **Devil's** and **Dictionary**, then type **<ctrl-Y>**. The word **From** has been moved within the line (although the spaces around it have not been moved). This routine is very handy, and should greatly speed your editing.

Remember, too, that unless you move the cursor between one set of deletions and another, the computer's storage area will not be erased, and you may yank back a jumble of text.

Arguments--exercises

The next few exercises show how arguments can be used to make your editing commands more powerful and efficient. Before beginning, type **<esc><** to move the cursor to the upper left-hand corner of your screen.

1. Lowercase the word **Devil** in line 8. Use no more than three commands. (<ctrl-U> plus a number and command counts as one command.)

Solution: To move the cursor, type **<ctrl-U>7<ctrl-N>**, then **<ctrl-U>3 <esc>F**. Then type **<esc>L**.

2. Kill the last four lines of the text. Use no more than two commands.

Solution: To move the cursor, type **<ctrl-A>**. Then type **<ctrl-U><ctrl-K>**.

3. Make two copies of the killed lines at the top of your text. Use no more than two commands.

Solution: To move the cursor, type **<esc><**. Then type **<ctrl-U>2<ctrl-Y>**.

4. Finally, delete the last 23 characters of the second line of text. Use no more than four commands.

Solution: To move the cursor, type **<esc><**, **<ctrl-N>**, **<ctrl-E>**, and then **<ctrl-U>23<del>**.

12. Buffers and Files

Before beginning this section, replace the mangled copy of the text on your screen with a fresh copy. Type the *quit* command <ctrl-X><ctrl-C> to exit from MicroEMACS without saving the text; once the COHERENT system's prompt appears, return to MicroEMACS by typing:

```
me text2
```

Now look at the status line at the bottom of your screen. It should appear as follows:

```
-- MicroEMACS -- text2 -- File: text2 -------------------------
```

As noted in the first half of this tutorial, the name on the left of the command line is that of the program. The name in the middle is the name of the *buffer* with which you are now working, and the name to the right is the name of the *file* from which you read the text.

Definitions

A *file* is a text that has been given a name and has been permanently stored in your computer. A *buffer* is a portion of the computer's memory that has been set aside for you to use, that may be given a name, and into which you can put text temporarily. You can put text into the buffer by typing it in from your keyboard or by *copying* it from a file.

Unlike a file, a buffer is not permanent: if your computer were to stop working (because you turned the power off, for example), a file would not be affected, but a buffer would be erased.

You must *name* your files because you work with many different files, and you must have some way to tell them apart. Likewise, MicroEMACS allows you to *name* your buffer, because with MicroEMACS you can work with more than one buffer at a time.

File and buffer commands

MicroEMACS gives you a number of commands for handling files and buffers. The following display summarizes them.

<ctrl-X><ctrl-W>	Write text to file
<ctrl-X><ctrl-F>	Rename file
<ctrl-X><ctrl-R>	Replace buffer with named file
<ctrl-X><ctrl-V>	Switch buffer or create a new buffer
<ctrl-X>K	Delete a buffer
<ctrl-X><ctrl-B>	Display the status of each buffer

Write and rename commands

The *write* command <ctrl-X><ctrl-W> was reviewed earlier, when the commands for saving text and exiting were discussed. To review, <ctrl-X><ctrl-W> changes the name of the file into which the text is saved, and then writes a copy of the text into that file.

Type <ctrl-X><ctrl-W>. MicroEMACS responds by printing

```
Write file:
```

on the last line of your screen.

Type **junkfile**, then a carriage return. Two things happen: First, MicroEMACS writes the message

```
[Wrote 11 lines]
```

at the bottom of your screen. Second, the name of the file shown on the status line has changed from **text2** to **junkfile**. MicroEMACS is reminding you that your text will be saved from now on to the file **junkfile**, unless you change the file name once again.

The *file rename* command <ctrl-X><ctrl-F> allows you rename the file to which you are saving text, *without* automatically writing the text to it. Type <ctrl-X><ctrl-F>. MicroEMACS will reply with the prompt:

```
Name:
```

Now type **text2**. Note that MicroEMACS does *not* send you a message that lines were written to the file; however, the name of the file shown on the status line has changed from **junkfile** back to **text2**. Until you change the name of the file again, every time you save the text from this buffer, it will be copied into the file **text2**.

Replace text in a buffer

The *replace* command <ctrl-X><ctrl-R> allows you to replace the text in your buffer with the text taken from file.

Suppose, for example, that you had edited text2 and saved it, and now wished to edit text1. You could exit from MicroEMACS, then re-invoke MicroEMACS for the file text2, but this is cumbersome. A more efficient way is to simply replace the text2 in your buffer with text1.

Type <ctrl-X><ctrl-R>. MicroEMACS replies with the prompt:

```
Read file:
```

Type text1. Notice that text2 has rolled away and been replaced with text1. Now, check the status line. Notice that although the name of the *buffer* is still text2, the name of the *file* has changed to text1. You can now edit text1; when you save the edited text, MicroEMACS will copy it back into the file text1--unless, of course, you choose to rename the file.

Visiting another buffer

The last command of this set, the *visit* command <ctrl-X><ctrl-V>, allows you to create more than one buffer at a time, to jump from one buffer to another, and move text between buffers. This powerful command has numerous features.

Before beginning, however, straighten up your buffer by replacing text1 with text2. Type the *replace* command <ctrl-X><ctrl-R>; when MicroEMACS replies by asking

```
Read file:
```

at the bottom of your screen, type text2.

You should now have the file text2 read into the buffer named text2.

Now, type the *visit* command <ctrl-X><ctrl-V>. MicroEMACS replies with the prompt

```
Visit file:
```

at the bottom of the screen. Now type text1. Several things will now happen. text2 rolls off the screen and is replaced with text1; the status line changes to show that both the buffer name and the file name are now text1; and the message

```
[Read 16 lines]
```

appears at the bottom of the screen.

This does *not* mean that your previous buffer has been erased, as it would have been had you used the *replace* command <ctrl-X><ctrl-R>. text2 is still being kept "alive" in a buffer and is available for editing; however, it is not being shown on your screen at the present moment.

Type <ctrl-X><ctrl-V> again, and when the prompt appears, type text2. text1 scrolls off your screen and is replaced by text2, and the message

```
[Old buffer]
```

appears at the bottom of your screen. You have just jumped from one buffer to another.

Move text from one buffer to another

The *visit* command **<ctrl-X><ctrl-V>** not only allows you jump from one buffer to another: it allows you to *move text* from one buffer to another as well. The following example shows how you can do this.

First, kill the first line of **text2** by typing the *kill* command **<ctrl-K>** twice. This removes both the line of text *and* the space that it occupied; if you did not remove the space as well the line itself, no new line will be created for the text when you yank it back. Next, type **<ctrl-X><ctrl-V>**; when the prompt

```
Visit file:
```

appears at the bottom of your screen, type **text1**. When **text1** has rolled onto your screen, type the *yank back* command **<ctrl-Y>**. The line you killed in **text2** has now been moved into **text1**.

Checking buffer status

The number of buffers you can use at any one time is limited only by the size of your computer. You should create only as many buffers as you need to use immediately; this will help the computer run efficiently.

To help you keep track of your buffers, MicroEMACS has the *buffer status* command **<ctrl-X><ctrl-B>**. Type **<ctrl-X><ctrl-B>**. Note that the status line has moved up to the middle of the of the screen, and the bottom half of your screen has been replaced with the following display:

```
C   Size Buffer          File
-   ---- ------          ----
*    913 text1           text1
*    423 text2           text2
```

This display is called the *buffer status window*. The use of windows will be discussed more fully in the following section.

The letter **C** over the leftmost column stands for **Changed**. An asterisk on a line indicates that the buffer has been changed, while a space means that the buffer has not been changed. **Size** indicates the buffer's size, in number of characters; **Buffer** lists the buffer name, and **File** lists the file name.

Now, kill the second line of **text1** by typing the *kill* command **<ctrl-K>**, then type **<ctrl-X><ctrl-B>** once again. Note that size of the buffer **text1** has been reduced from 913 character to 881, to reflect the decrease in the size of the buffer.

To make this display disappear, type the *one window* command <ctrl-X>1. This command will be discussed in full in the next section.

Renaming a buffer

One more point must be covered with the *visit* command. The COHERENT system will not allow you to have more than one file with the same name, in order to avoid confusion; for the same reason, MicroEMACS will not allow you to have more than one *buffer* with the same name.

Ordinarily, when you visit a file that is not already in a buffer, MicroEMACS will create a new buffer and give it the same name that the file you are visiting has. However, if for some reason you already have a buffer with the same name as the file you wish to visit, MicroEMACS will stop and ask you to give a new, different name to the buffer it is creating.

For example, suppose that you wanted to visit a new *file* called **sample**, perhaps from another directory, but you already had a *buffer* named **sample**. MicroEMACS would stop and give you this prompt at the bottom of the screen:

```
Buffer name:
```

When you named this new buffer, MicroEMACS would proceed to read **text2** into it.

Delete a buffer

If you wish to delete a buffer, simply type the *delete buffer* command <ctrl-X>K. This command will allow you only to delete a buffer that is hidden, not one that is being displayed.

Type <ctrl-X>K. MicroEMACS will give you the prompt:

```
Kill buffer:
```

Type **text2**. Because you have changed the buffer, MicroEMACS asks:

```
Discard changes [y/n]?
```

Type **y**, and then type the *buffer status* command <ctrl-X><ctrl-B>; the buffer status window will no longer show the buffer **text2**. Note that although the prompt refers to *killing* a buffer, the buffer is in fact *deleted* and cannot be yanked back.

Summary

These buffer and file commands allow you to edit more than one text at once, and move text between buffers and files. Just how useful these commands are will be seen when you cover the next topic, *windows*.

13. Windows

Before beginning this section, it will be necessary to create a new text file. Exit from MicroEMACS by typing the *quit* command **<ctrl-X><ctrl-C>**; then reinvoke MicroEMACS for the text file **text1** with the command:

```
me text1
```

Now, copy **text2** into a buffer by typing the **visit** command **<ctrl-X><ctrl-V>**. When the message

```
Visit file:
```

appears at the bottom of your screen, type **text2**. MicroEMACS will read **text2** into a buffer, and show the message

```
[Read 11 lines]
```

at the bottom of your screen.

Finally, copy a new text, called **text3**, into a buffer. Type **<ctrl-X><ctrl-V>** again. When MicroEMACS asks which file to visit, type **text3**. The message

```
[Read 92 lines]
```

will appear at the bottom of your screen.

The first screenful of text will appear as follows:

```
From "Gulliver's Travels":
    I said there was a society of men among us,
bred up from their youth in the art of proving
by words multiplied for the purpose, that
white is black, and black is white, according
as they are paid.  To this society all the rest
of the people are slaves.
    "For example.  If my neighbor hath a mind to my
cow, he hires a lawyer to prove that he ought to
have my cow from me.  I must then hire another to
defend my right; it being against all rules of law
that any man should be allowed to speak for himself.
Now in this case, I who am the true owner lie under
two great disadvantages.  First, my lawyer being
practiced almost from his cradle in defending
falsehood is quite out of his element when he would
be an advocate for justice, which as an office
unnatural, he always attempts with great awkwardness,
if not ill-will.  The second disadvantage is that my
lawyer must proceed with great caution, or else he
will be reprimanded by the judges, and abhorred by
his brethren, as one who would lessen the practice
-- MicroEMACS -- text3 -- File: text3 ------------------
```

At this point, text3 is on your screen, and text1 and text2 are hidden.

You could edit first one text and then another, while remembering just how things stood with the texts that were hidden; but it would be much easier if you could display all three texts on your screen simultaneously. MicroEMACS allows you to do just that, by using *windows*.

Window commands

A *window* is a portion of your screen that is set aside and can be manipulated independently from the rest of the screen. MicroEMACS's commands for manipulating windows are summarized in the following display.

<ctrl-X>2	Create a window
<ctrl-X>1	Delete extra windows
<ctrl-X>N	Move to next window
<ctrl-X>P	Move to previous window
<ctrl-X>Z	Enlarge window
<ctrl-X><ctrl-Z>	Shrink window
<ctrl-X><ctrl-N>	Scroll down
<ctrl-X><ctrl-P>	Scroll up
<esc>!	Move within window
<ctrl-X>B	Switch buffer

Creating windows and moving between them

The best way to grasp how a window works is to create one and work with it. Type the *create a window* command **<ctrl-X>2**.

Your screen is now divided into two parts, an upper and a lower. The same text is in each part, and the command lines give **text3** for the buffer and file names. Also, note that you still have only one cursor, which is in the upper left-hand corner of the screen.

The next step is to move from one window to another.

Type the *next window* command **<ctrl-X>N**. Your cursor has now jumped to the upper left-hand corner of the *lower* window.

Type the *previous window* command **<ctrl-X>P**. Your cursor has returned to the upper left-hand corner of the top window.

Now, type **<ctrl-X>2** again. The window on the top of your screen is now divided into two windows, for a total of three on your screen. Type **<ctrl-X>2** again. The window at the top of your screen has again divided into two windows, for a total of four.

It is possible to have as many as 11 windows on your screen at one time, although each window will show only the control line and one or two lines of text. Note that neither **<ctrl-X>2** nor **<ctrl-X>1** can be used with arguments.

Now, type the *one window* command **<ctrl-X>1**. Note that all extra windows have been eliminated, or *closed*.

Enlarging and shrinking windows

When MicroEMACS creates a window, it divides the window in which the cursor is positioned into half. You do not have to leave the windows at the size MicroEMACS creates them, however. If you wish, you may adjust the relative size of each window on your screen, using the *enlarge window* and *shrink window* commands.

Type **<ctrl-X>2** twice. Your screen is now divided into three windows: two in the top half of your screen, and the third in the bottom half.

Now, type the *enlarge window* command **<ctrl-X>Z**. The window at the top of your screen is now one line bigger, because it has borrowed a line from the window below it. Type **<ctrl-X>Z** again. Once again, the top window has borrowed a line from the middle window.

Now, type the *next window* command **<ctrl-X>N**, to move your cursor into the middle window. Again, type the *enlarge window* command **<ctrl-X>Z**. The middle window has borrowed a line from the bottom window, and is now one line larger.

The *enlarge window* command <ctrl-X>Z allows you to enlarge the window your cursor is in by borrowing lines from another window, provided that you do *not* shrink that other window out of existence. Every window must have at least two lines in it, one command line and one line of text.

The *shrink window* command <ctrl-X><ctrl-Z> allows you to decrease the size of a window. Type <ctrl-X><ctrl-Z>. The middle window is now one line smaller, and the bottom window is one line larger, because the line borrowed earlier has been returned.

The *enlarge window* and *shrink window* commands can also be used with arguments introduced with <ctrl-U>. However, remember that MicroEMACS will not accept an argument that would shrink another window out of existence.

Displaying text within a window

Displaying text within the limited area of a window can present special problems. The *view* commands <ctrl-V> and <esc>V will roll window-sized portions of text up or down, but you may become disoriented when a window shows only four or five lines of text at a time. Therefore, three special commands are available for displaying text within a window.

Two commands allow you to move your text by one line at a time, or *scroll* it: the *scroll up* command <ctrl-X><ctrl-N>, and the *scroll down* command <ctrl-X><ctrl-P>.

Type <ctrl-X><ctrl-N>. Note that the line at the top of your window has vanished, a new line has appeared at the bottom of your window, and the cursor is now at the beginning of what had been the second line of your window.

Now type <ctrl-X><ctrl-P>. The line at the top that had vanished earlier has now returned, the cursor is at the beginning of it, and the line at the bottom of the window has vanished. These commands allow you to move forward in your text slowly, so that you do not become disoriented.

Both of these commands can be used with arguments introduced by <ctrl-U>.

The third special movement command is the *move within window* command <esc>!. This command moves the line your cursor is on to the top of the window.

To try this out, move the cursor down three lines by typing <ctrl-U>3<ctrl-N>, then type <esc>!. (Be sure to type an exclamation point '!', not a numeral one '1', or nothing will happen.) Note that the line to which you had moved the cursor is now the first line in the window, and three new lines have scrolled up from the bottom of the window. You will find this command to be very useful as you become more experienced at using windows.

All three special movement commands can also be used when your screen has no extra windows, although you will not need them as much.

One buffer

Now that you have been introduced to the commands for manipulating windows you can begin to use windows to speed your editing.

To begin with, scroll up the window you are in until you reach the top line of your text. You can do this either by typing the *scroll up* command **<ctrl-X><ctrl-P>** several times, or by typing **<esc><**.

Kill the first line of text with the *kill* command **<ctrl-K>**. Note that the first line of text has vanished from all three windows. Now, type **<ctrl-Y>** to yank back the text you just killed. The line has reappeared in all three windows.

The main advantage to displaying one buffer with more than one window is that each window can display a different portion of the text. This can be quite helpful if you are editing or moving a large text.

To demonstrate this, kill the last four lines at the end of the text, and move them to the beginning of the text. First, move to the end of the text in your present window by typing the *end of text* command **<esc>>**. Kill the last four lines.

You could move the killed lines to the beginning of your text by typing the *beginning of text* command **<esc><**; however, it is more convenient simply to type the *next window* command **<ctrl-X>N**, which will move you to the beginning of the text as displayed in the next window. Note that MicroEMACS remembers a differnt cursor position for each window.

Now yank back the four killed lines by typing **<ctrl-Y>**. You can observe simultaneously that the lines have been removed from the end of your text and that they have been restored at the beginning.

Multiple buffers

Windows are especially helpful when they display more than one text. Remember that at present you are working with *three* texts, **text1**, **text2**, and **text3**, although your screen is displaying only **text3**. To display a different text in a window, use the *switch buffer* command **<ctrl-X>B**.

Type **<ctrl-X>B**. When MicroEMACS asks

```
Use buffer:
```

at the bottom of the screen, type **text1**. The text in your present window will be replaced with **text1**. Note that the command line in that window has changed, too, to reflect the fact that the buffer and the file names are now **text1**.

Moving and copying text among buffers

Copying text among buffers is now quite easy. To see how this is done, kill the first line of **text1** by typing the **<ctrl-K>** command twice. Yank back the line immediately by typing **<ctrl-Y>**. Remember, the line you killed has *not* been erased from its special storage area, and may be yanked back any number of times.

Now, move to the previous window by typing **<ctrl-X>P**, then yank back the killed line by typing **<ctrl-Y>**. This technique allows you to *copy* text from one window to another as well as to *move* it within a window.

Checking buffer status

The *buffer status command* **<ctrl-X><ctrl-B>** can be used when you are already displaying more than one window on your screen.

When you want to remove the buffer status window, use either the *one window* command **<ctrl-X>1**, or move your cursor into the buffer status window using the *next window* command **<ctrl-X>N** and replace it with another buffer by typing the *switch buffer* command **<ctrl-X>B**.

Saving text from windows

The final step is to save the text from your windows and buffers. Close the lower two windows with the *one window* command **<ctrl-X>1**. Remember, when you close a window, the text that it displayed is still kept in a buffer that is *hidden* from your screen.

When you use the *save* command **<ctrl-X><ctrl-S>**, only the text in the window in which the cursor is position will be written to its file. If only one window is displayed on the screen, the *save* command will save only its text.

Exercises

The following exercises will help you master the use of windows and buffers. Before you begin, exit from MicroEMACS by typing the *quit* command **<ctrl-X><ctrl-C>**.

1. Invoke MicroEMACS to edit **text1**. Display **text2** in a separate window. Copy the first four lines of **text1** to **text2**. Destroy **text1**'s buffer. Exit from MicroEMACS without saving the text.

Solution: To invoke MicroEMACS for **text1** type:

```
me text1
```

Before **text2** can be displayed on a separate window, it must be copied into a buffer; type the *visit* command **<ctrl-X><ctrl-V>**. When MicroEMACS asks

```
Visit file:
```

type **text2**.

When **text2** has been copied into its buffer, type the *create window* command <ctrl-X>2, and read **text1** into the window with the *switch buffer* command <ctrl-X>B.

To copy the top four lines from **text1** to **text2**, first kill the first four lines by typing <ctrl-U>4 <ctrl-K>. Then yank them back immediately by typing <ctrl-Y>; jump to **text2**'s window by typing the *next window* command <ctrl-X>N, and finally yank back the four killed lines again by typing <ctrl-Y>.

To destroy the buffer holding the copy of **text1**, type the *one window* command <ctrl-X>1, then type the *delete buffer* command <ctrl-X>K. When MicroEMACS asks

```
Kill buffer:
```

type **text1**. When MicroEMACS asks

```
Discard changes [y/n]?
```

type y.

Finally, to exit without saving text, type the *quit* command <ctrl-X><ctrl-C>.

2. Display **text1**, **text2**, and **text3** in three equally sized windows. Scroll through each text in turn. Check the status of the buffers to see if any were altered, then exit from MicroEMACS without saving the texts.

Solution: Invoke MicroEMACS to edit **text1** by typing

```
me text1
```

and then copy **text2** and **text3** into buffers by using the *visit* command <ctrl-X><ctrl-V> twice.

Create three windows on your screen by typing the *create a window* command <ctrl-X>2 twice.

Move among the windows by using the *next window* command <ctrl-X>N and the *previous window* command <ctrl-X>P; then make sure the windows are of even size by using the *enlarge window* command <ctrl-X>Z and the *shrink window* command <ctrl-X><ctrl-Z>.

Read **text1** and **text2** into your extra windows by using the *switch buffer* command <ctrl-X>B.

To scroll through the texts, use the *scroll down* command <ctrl-X><ctrl-N> and the *scroll up* command <ctrl-X><ctrl-P>.

When you are done scrolling, check the buffers' status with the *buffer status* command **<ctrl-X><ctrl-B>**. Close the buffer status window by typing the *one window* command **<ctrl-X>1**.

Exit from MicroEMACS by typing the *quit* command **<ctrl-X><ctrl-C>**.

14. Keyboard Macros

Another helpful feature of MicroEMACS is that it allows you to create a *keyboard macro*.

Before beginning this section, reinvoke MicroEMACS to edit **text3** by typing the command

```
me text3
```

The term *macro* means a number of commands or characters that are bundled together. Although MicroEMACS allows you to create only one macro at a time, this macro can consist of the most common *phrase* or the most common *command* or *series of commands* that you use while editing your file.

Keyboard macro commands

The keyboard macro commands are as follows:

<ctrl-X>(Begin macro collection
<ctrl-X>) End macro collection
<ctrl-X>E Execute macro

To begin to create a macro, type the *begin macro* command <ctrl-X>(. Be sure to type an open parenthesis '(', not a numeral '9'. MicroEMACS will reply with the message

```
[Start macro]
```

Type the following phrase:

```
interactive screen editor
```

Then type the *end macro* command <ctrl-X>). Be sure you type a close parenthesis ')', not a numeral '0'. MicroEMACS will reply with the message

```
[End macro]
```

Move your cursor down two lines and execute the macro by typing the *execute macro* command <ctrl-X>E. The phrase you typed into the macro has been inserted into your text.

Replacing a macro

To replace this macro with another, go through the same process. Type <ctrl-X>(. Then type the *buffer status* command <ctrl-X><ctrl-B>, and type <ctrl-X>). Remove the buffer status window by typing the *one window* command <ctrl-X>1.

Now execute your keyboard macro by typing the **execute macro** command <ctrl-X>E. The *buffer status* command has executed once more.

Note that whenever you exit from MicroEMACS, your keyboard macro is erased, and must be retyped when you return.

Search and replace

You can create a *search and replace* routine by bundling a number of commands together under a keyboard macro.

Suppose, for example, that you wished to replace the word **lawyer** throughout **text3** with the word **doctor**. First, type the *begin macro* command <ctrl-X>(. Then, type <ctrl-S>; when MicroEMACS gives you the prompt

```
Search:
```

type **lawyer**. This command will execute, and move the cursor to the space immediately after the first occurrence of the word **lawyer**.

Use the <del> key to erase the word **lawyer**. Type the word **doctor** in its place.

Now that the search and replace routine is set, type the *end macro* command <ctrl-X>). To execute the macro, first install an argument by typing <ctrl-U> three times; then type the *execute macro* command <ctrl-X>E. Although the word **lawyer** does not occur 64 times in this text, you must make your argument large enough so that your search and replace macro will find every occurrence of the word you want to change.

MicroEMACS will search through your file for the word **lawyer**, and replace it with **doctor**. The cursor will stop after the last occurrence of **lawyer**, and the message

```
Not found
```

will appear at the bottom of your screen. This message means that although MicroEMACS continued to search for **lawyer** through the end of your file, it could not find it again.

15. Sending Commands to COHERENT

The last commands you need to learn are the *program interrupt* commands **<ctrl-X>!** and **<ctrl-C>**. These commands allows you to interrupt your editing, give a command directly to the COHERENT system, and then resume editing without affecting your text in any way.

The command **<ctrl-X>!** allows you to send *one* command to the operating system. To see how this command works, type **<ctrl>!**. Note that the prompt **!** has appeared at the bottom of your screen. Type **lc**. Observe that the directory's table of contents scrolls across your screen, followed by the message **[end]**.

To return to your editing, simply type a carriage return. The *interrupt* command <ctrl-C> suspends editing indefinitely, and allows you to send an unlimited number of commands to the operating system. To see how this works, type **<ctrl-C>**. After a moment, the COHERENT system's prompt will appear at the bottom of your screen. Type **time**. The COHERENT system will reply by printing the time and date. To resume editing, then simply type **<ctrl-D>**.

If you wish, you can suspend MicroEMACS's operation, tell the COHERENT system to invoke another copy of the MicroEMACS program, edit a file, then return to your previous editing. To see how this is done, type <ctrl-C>. When the prompt appears at the bottom of your screen, type

```
me text1
```

It doesn't matter that you are already editing **text1**. MicroEMACS will simply copy the text1 file into a new buffer and let you work as if the other MicroEMACS program you just interrupted never existed.

Exit from this second MicroEMACS program by typing the *quit* command **<ctrl-X><ctrl-C>**. Then type **<ctrl-D>**. Your original MicroEMACS program has now been resumed. Note, however, that none of the changes you made in the secondary MicroEMACS program will be seen here.

It is not a good idea to use multiple MicroEMACS programs to edit the same program: it is too easy to become confused as to which edits were made to which version.

The only time this is advisable, is if you wish to test to see how a certain edit would affect your text: you can create a new MicroEMACS program, test the command, and then destroy the altered buffer and return to your original editing program without having to worry that you might make errors that are difficult to correct.

16. Advanced Editing--Conclusion and Summary

This concludes the tutorial for the MicroEMACS interactive screen editor. Congratulations on your diligence in working through it! MicroEMACS and its related EMACS-based screen editors are now at your command, and you have acquired a skill that will serve you well.

This section introduced the advanced editing techniques available with MicroEMACS.

Arguments can be used with most cursor movement commands and all text deletion commands to set the number of times they execute. The command <ctrl-U> introduces arguments. The default for <ctrl-U> is 4, with each subsequent entry of <ctrl-U> multiplying the argument by 4. The value of an argument can be changed by typing <ctrl-U>, followed by an integer.

Text is edited in a buffer, and is copied into a file when the user issues the *save* commands <ctrl-X><ctrl-S> or the *write* command <ctrl-X><ctrl-W>. <ctrl-X><ctrl-F> will rename the file; and <ctrl-X><ctrl-W> renames the file and automatically copies text to it.

MicroEMACS can handle more than one buffer at a time. <ctrl-X><ctrl-V> moves you from one buffer to another, and allows you to create a buffer should the buffer you requested not already exist.

<ctrl-X><ctrl-R> replaces the text in a buffer with the text drawn from a specified file. <ctrl-X>K deletes a buffer.

<ctrl-X><ctrl-B> displays information on the status of each buffer.

The screen can be divided into windows, which can display either the same buffer or different buffers. <ctrl-X>2 creates a window by dividing the present window in half, whereas the command <ctrl-X>1 erases all extra windows.

<ctrl-X>Z and <ctrl-X><ctrl-Z> enlarge and shrink windows, respectively. <ctrl-X>N moves the cursor to the next, or lower, window, whereas <ctrl-X>P moves the cursor to the previous window.

<ctrl-X><ctrl-N>, <ctrl-X><ctrl-P>, and <esc>! respectively scroll the window up, scroll it down, and move the line on which the cursor rests to the top of the window. <ctrl-X>B displays a different buffer within a window.

MicroEMACS allows you to create a keyboard macro that executes a set of commands or inserts text. <ctrl-X>(opens the macro, <ctrl-X>) closes it, and <ctrl-X>E executes it.

<ctrl-C> interrupts the operation of MicroEMACS, so that the user can send commands directly to the COHERENT system. You can resume working with MicroEMACS by typing <ctrl-D>.

17. Summary of Commands

<ctrl-@> Set mark at current position.

<ctrl-A> Move to start of line.

<ctrl-B> (Back) Move backward by characters.

<ctrl-C> Suspend MicroEMACS and move temporarily to the COHERENT system.

<ctrl-D> (Delete) Delete next character.

<ctrl-E> (End) Move to end of line.

<ctrl-F> (Forward) Move forward by characters.

<ctrl-G> Abort from a command.

<ctrl-K> (Kill) With no argument, kill from current position to end of line; if at the end, kill the newline. With argument 0, kill from beginning of line to current position. Otherwise, kill *argument* lines forward (if positive) or backward (if negative).

<ctrl-L> Redraw the screen.

<ctrl-N> (Next) Move to next line.

<ctrl-P> (Previous) Move to previous line.

<ctrl-R> (Reverse) Prompt for search string and search backward.

<ctrl-S> (Search) Prompt for search string and search forward.

<ctrl-T> (Transpose) Transpose the characters before and after the current position.

<ctrl-U> Specify an argument, as described above.

<ctrl-V> Move forward by pages.

<ctrl-W> Kill text from current position to mark.

<ctrl-X>(Begin a macro definition. MicroEMACS collects everything typed until the end of the definition for subsequent repeated execution.

<ctrl-X>) End a macro definition.

<ctrl-X>1 Display only the current window.

<ctrl-X>2 Split the current window; usually followed by **<ctrl-X>B** or **<ctrl-X><ctrl-V>**.

<ctrl-X>B (Buffer) Prompt for a buffer name and display the buffer in the current window.

<ctrl-X>E (Execute) Execute macro.

<ctrl-X>K (Kill) Prompt for a buffer name and delete it.

<ctrl-X>N (Next) Move to next window.

<ctrl-X>P (Previous) Move to previous window.

<ctrl-X>Z Enlarge the current window by *argument* lines.

<ctrl-X><ctrl-B>
Create a window that shows the size, buffer name, and file name for each buffer, and also shows whether a buffer has been changed. IP
<ctrl-X><ctrl-C>
Prompt, and exit unconditionally if 'Y' given.

<ctrl-X><ctrl-F>
(File name) Prompt for a file name for current buffer.

<ctrl-X><ctrl-N>
Move current window down by *argument* lines.

<ctrl-X><ctrl-P>
Move current window up by *argument* lines.

<ctrl-X><ctrl-R>
(Read) Prompt for a file name, delete current buffer, and read the file.

<ctrl-X><ctrl-S>
(Save) Save current buffer to the associated file.

<ctrl-X><ctrl-V>
(Visit) Prompt for a file name and display the file in the current window.

<ctrl-X><ctrl-W>
(Write) Prompt for a file name and write the current buffer to it.

<ctrl-X><ctrl-Z>
Shrink current window by *argument* lines.

<ctrl-Y> (Yank) Copy the kill buffer into text at the current position; set current position to the end of the new text.

<ctrl-Z> Save current buffer to associated file and exit.

<Del> If no argument, delete the previous character. Otherwise, kill *argument* previous characters.

<esc>! Move current line to a position in the window given by *Argument*; the position is in lines from the top if positive, in lines from the bottom if negative, and the center of the window if 0.

<esc>> Move to end of buffer.

<esc>< Move to beginning of buffer.

<esc>B (Back) Move backward by words.

<esc>C (Capitalize) Capitalize the next word.

<esc>D (Delete) Kill the next word.

<esc><Del> Kill the previous word.

<esc>F (Forward) Move forward by words.

<esc>L (Lower) Convert the next word to lower case.

<esc>U (Upper) Convert the next word to upper case.

<esc>V Move backward by pages.

Index