TEK USERS MANUAL

Part No. 061-3216-00
Product Group 07

4405

ARTIFICIAL INTELLIGENCE SYSTEM

Please Check at the Rear of this Manual for NOTES and CHANGE INFORMATION

First Printing APR 1986

Tektronix®
COMMITTED TO EXCELLENCE

WARRANTY FOR SOFTWARE PRODUCTS

Tektronix warrants that this software product will conform to the specifications set forth herein, when used properly in the specified operating environment, for a period of three (3) months from the date of shipment, or if the program is installed by Tektronix, for a period of three (3) months from the date of installation. If this software product does not conform as warranted, Tektronix will provide the remedial services specified below. Tektronix does not warrant that the functions contained in this software product will meet Customer's requirements or that operation of this software product will be uninterrupted or error-free or that all errors will be corrected.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for such service in accordance with the instructions received from Tektronix. If Tektronix is unable, within a reasonable time after receipt of such notice, to provide the remedial services specified below, Customer may terminate the license for the software product and return this software product and any associated materials to Tektronix for credit or refund.

This warranty shall not apply to any software product that has been modified or altered by Customer. Tektronix shall not be obligated to furnish service under this warranty with respect to any software product a) that is used in an operating environment other than that specified or in a manner inconsistent with the Users Manual and documentation or b) when the software product has been integrated with other software if the result of such integration increases the time or difficulty of analyzing or servicing the software product or the problems ascribed to the software product.

TEKTRONIX DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO PROVIDE REMEDIAL SERVICE WHEN SPECIFIED, REPLACE DEFECTIVE MEDIA OR REFUND CUSTOMER'S PAYMENT IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

PLEASE FORWARD ALL MAIL TO:

Artificial Intelligence Machine
Tektronix, Inc.
P.O. Box 1000 M.S. 60-405
Wilsonville, Oregon 97070
Attn: AIM Documentation

MANUAL REVISION STATUS

PRODUCT: 4405 ARTIFICIAL INTELLIGENCE SYSTEM

This manual supports the following versions of this product: Serial Numbers B010100 and up.

REV DATE	DESCRIPTION
APR 1986	Original Issue

Table of Contents

Figures

Examples

Tables

Section 1
INTRODUCTION

ABOUT PRELIMINARY MANUALS

Some of the 4400 Series manuals are preliminary. They are as complete and accurate as they can be, given the lead time required for writing and printing. Some of the information given in preliminary manuals may be inaccurate or missing.

We of AIM Documentation want each customer to receive complete, final manuals as soon as possible. In order that you not be missed (sometimes addresses get lost or are unavailable) please fill out and return the enclosed, postage-free, card. Of course, we would appreciate any comments you can make to help us make the documentation for this system better. (We also like praise, too, if we've done something right.) If the card is missing, or if you have more comments to send at a later time, send them to:

Tektronix, Inc.
P.O. Box 1000
Wilsonville, OR 97070
D.S. 60-405
Attention: AIM Documentation

ABOUT THIS MANUAL

This is the introductory manual to the 4405 Artificial Intelligence System. (We call it the 4405 from here on.) You should read this manual before attempting to use your new 4405 — it contains useful information that can help you get the most out of the 4405. You'll find instructions on how to unpack and connect the system, a guided tour of the first time you turn it on, a discussion on how to work with the system on a day-by-day basis, suggestions for working with more than one user on the system, and some ways to recover from the inevitable errors to which humans are prone.

YOUR QUALIFICATIONS

This manual is not a tutorial document. Although we've tried to keep this manual clear and simple, it assumes that you're an experienced computer user. You need not have systems programming experience, or be a "Wizard," but you should be past the neophyte stage on a large computer operating system. You should be comfortable with a hierarchical or tree-structured filing system, know how to issue commands and run applications, and, in general, be a "knowledgeable user."

If you're comfortable with an operating system such as Unix®, TOPS-10/20®, VMS®, RSTS®, or any of the other multi-user operating systems that abound, you should have little trouble adapting to the 4405 operating system. If you are not, you should take a course, read through one of the many tutorials, or find a knowledgeable person to help you adjust to the 4405. If you must search out tutorial information on your own, look for tutorials on the Unix operating system. The 4405 operating system is *not* Unix, but the structure and philosophy behind it is not too dissimilar, and tutorial information explaining the Unix operating system is readily available.

UNPACKING THE 4405

If you've just purchased your 4405 — congratulations! You'll want to get it put together and running as soon as possible. First, *don't discard your shipping cartons.* If you ever want to move your 4405 any distance, you should use these containers. Exercise patience, follow instructions, and you should have no trouble bringing the system up for the first time.

To save time and trouble, follow this sequence:

1. Skim over this procedure to get a feel for the sequence.
2. Turn to Appendix A and follow those instructions. This appendix shows you, in detail, how to unpack and connect the pieces that form the 4405.
3. Come back to this point and read the rest of Section 1, the introduction. You need to get an overview of the 4405 before trying to use it.
4. Read Section 2, *The First Time User* next. Follow the examples on your 4405. This section takes you through an initial introduction to the 4405 operating system and shows you a few of its many features.
5. Next read through Section 3, *Using the 4405.* This section talks about normal, day-to-day use of the 4405. You'll want to experiment with the system — you shouldn't have any trouble at this point.
6. Read Section 4, *Software Maintenance,* to get an idea of the maintenance tasks you'll have to deal with. If more than one user will be using the 4405, one person should be responsible for these tasks.
7. Section 5, *Recovery and Rebuild,* is the section we all hope to avoid. If you must, you can find system rebuilding procedures here.

OVERVIEW OF THE 4405

GENERAL INFORMATION

The 4405 is a single-user computer system that has been designed for the efficient development and use of artificial intelligence (AI) applications. (*Single-user* means that, although the 4405 can have many separate user accounts, it is not a time-sharing system; only one user may be logged in at a time.) The 4405 can be used both as a stand-alone applications development system and as a terminal connected to a host computer.

As an applications development system, the 4405 provides a programming environment for the Smalltalk-80 system, LISP, and PROLOG. These languages run under the 4405's multi-tasking operating system. The 4405 also has a hierarchical file system, complete with various graphics and mathematics libraries to assist with program development. In addition, the 4405 contains a 'C' programming environment that allows porting of many applications programs to it.

As a terminal, the 4405 easily interfaces, via a RS-232-C line, to various computers in use by the AI community. The 4405 functions as an ANSI X3.64 compatible terminal with some extensions to allow it to work with most popular screen-oriented editors.

4405 HARDWARE OVERVIEW

The 4405 consists of two major components, the Display/CPU module and mass storage unit (MSU). These, in addition to a keyboard and mouse, make up the basic 4405. The following discussion examines each component in detail.

The 4405 Display/CPU

The Display/CPU is the heart of the 4405 system. Although it is no larger than the display cabinet of a conventional terminal, it contains a monochrome 640 X 480 pixel bit-map display, the central processing unit, one megabyte (standard, two or four additional megabytes available as options) of fast semiconductor memory, and the interface electronics that allow the 4405 to communicate with the outside world.

Connected to the rear of the Display/CPU unit are the power cord, mass storage unit, keyboard, and mouse. There, we also find a volume control for the internal speaker, an audio connector for an external speaker, a parallel printer port, the RS-232 port, the ethernet connector, and a reset button.

The 4405 uses a monochrome 640 X 480 pixel display as a window into a 1376 X 1024 bit-map. The 4405's electronics allow smooth panning of the 640 X 480 window over the virtual display under control of the operating system.

Figure 1-1 shows how the 640 X 480 pixel display relates to the 1376 X 1024 bit-map.

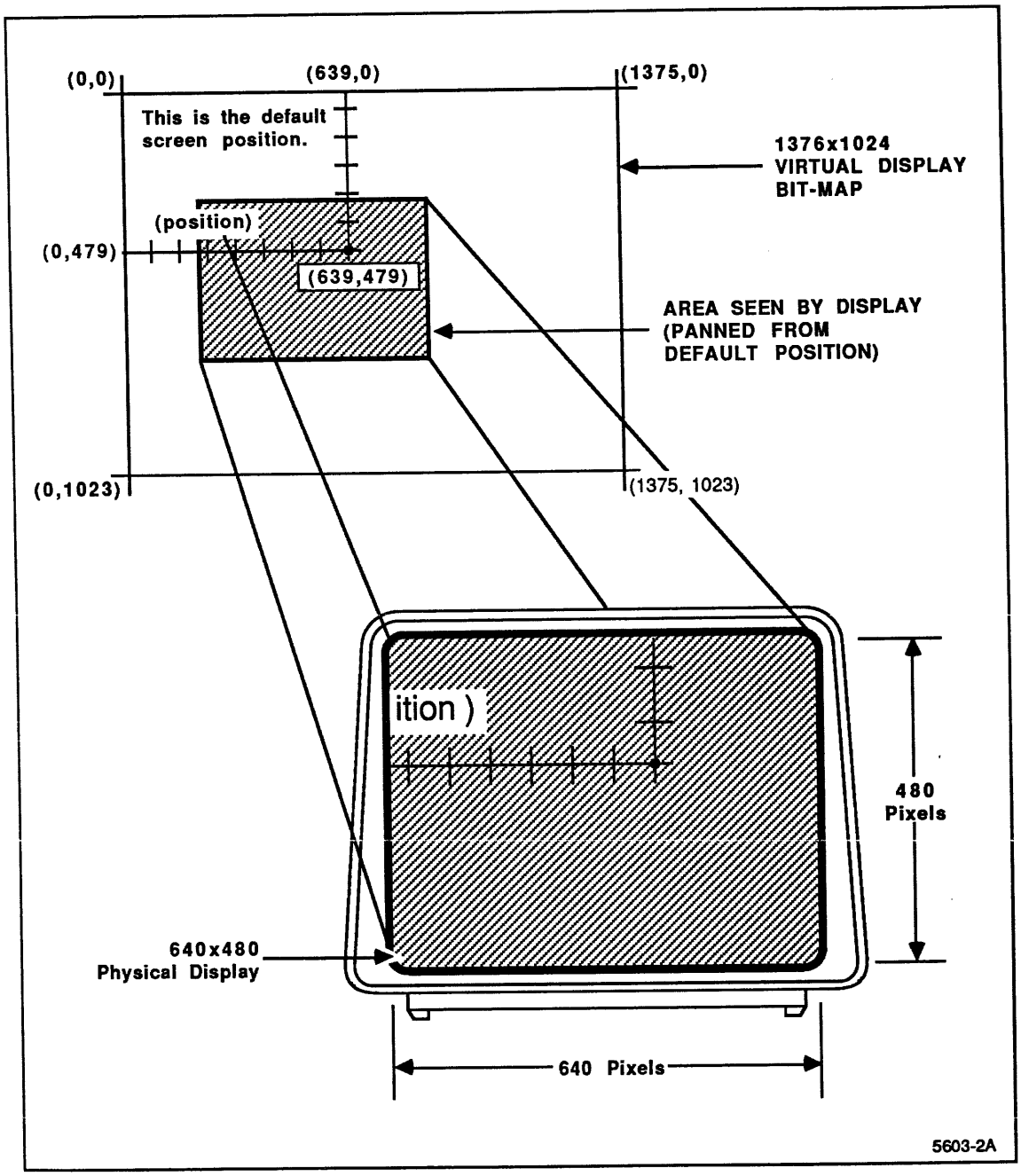

Figure 1-1. 640 X 480 Window Into 1376 X 1024 Bit-Map.

Figure 1-2 shows the front of the Display/CPU unit. Two controls are located at the front of the unit: the brightness control and the power switch. Figure 1-3 shows the rear panel controls and connectors on the Display/CPU.

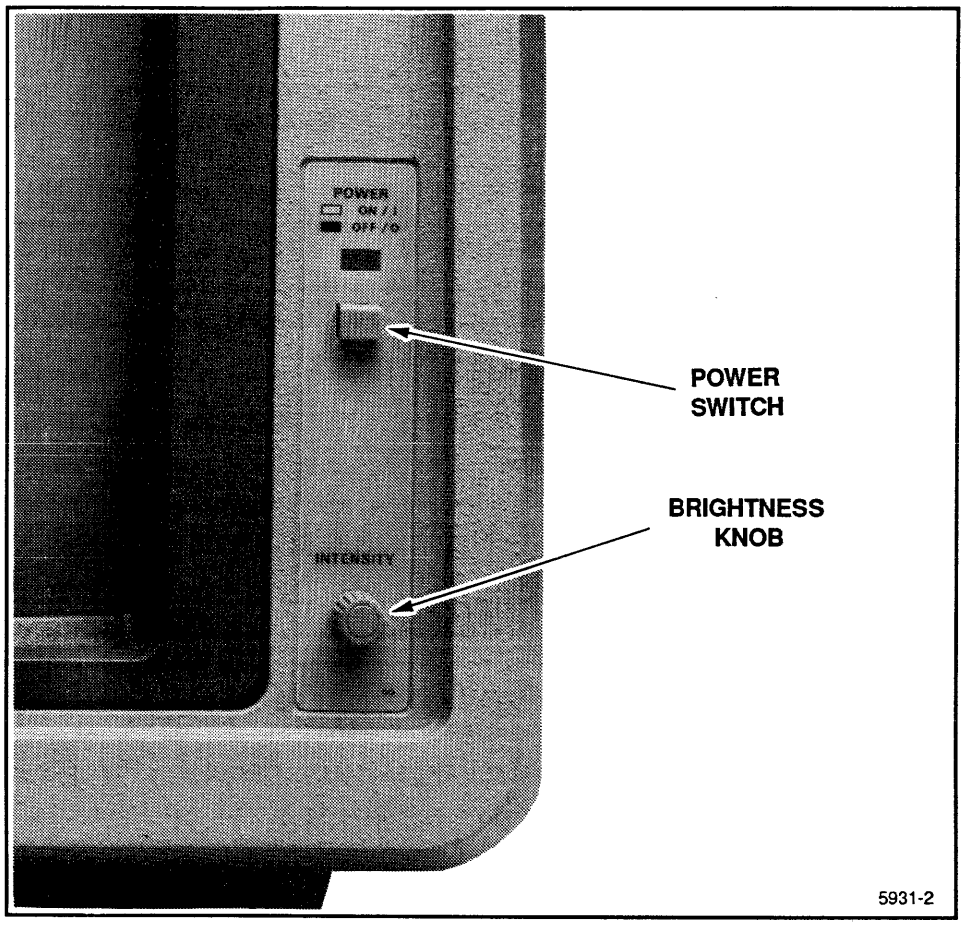

Figure 1-2. Display/CPU Front Panel Controls.

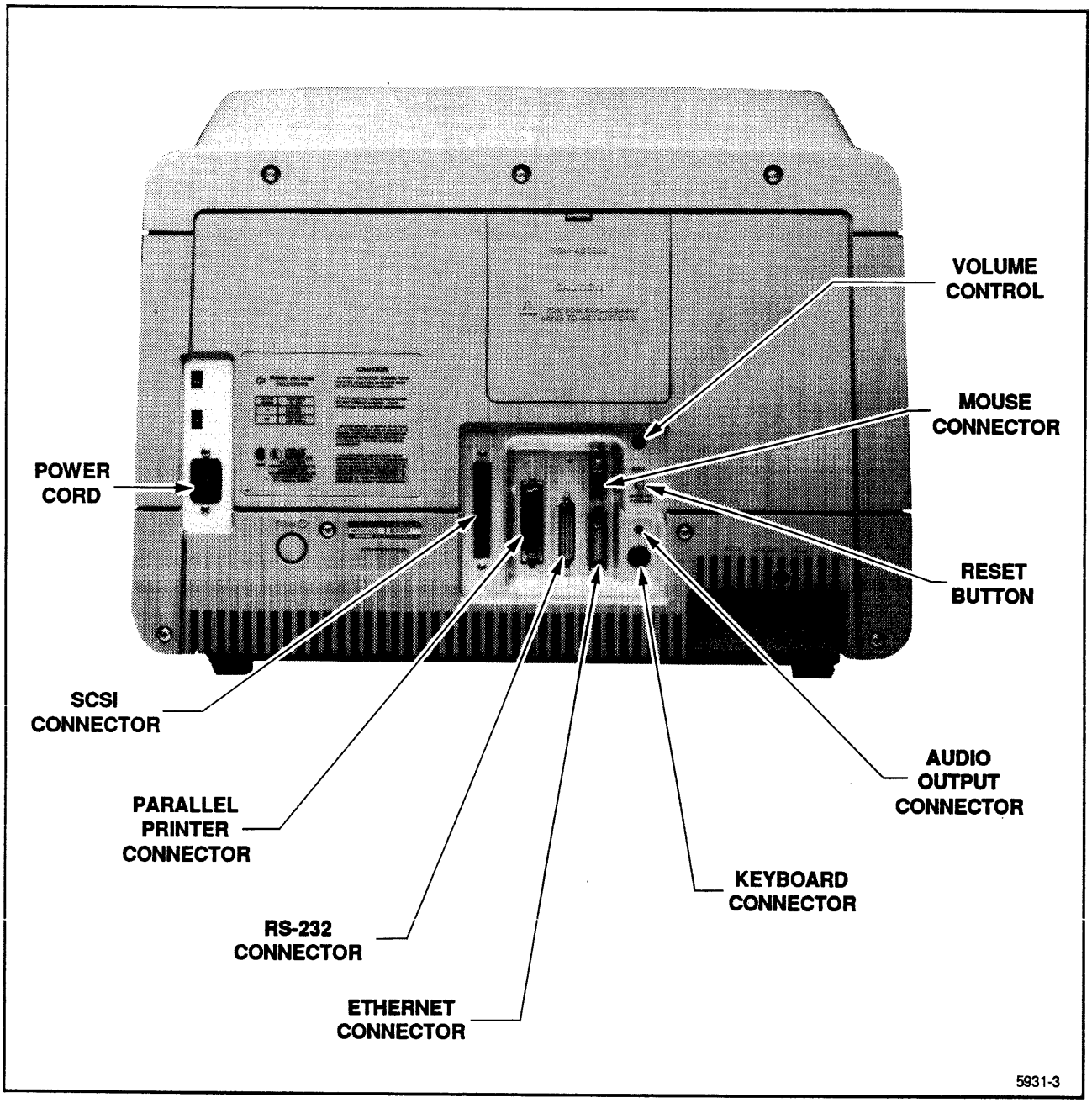

Figure 1-3. Display/CPU Rear Panel.

The 4405 Keyboard

Figure 1-4 shows the 4405 keyboard. This keyboard is similar to that used by the Tektronix 4100 Series terminals. The joydisk, function keys, numeric pad, and keyboard keys are all accessible to the 4405 software. The only unfamiliar key is the up-arrow/left-arrow key used in Smalltalk programming.

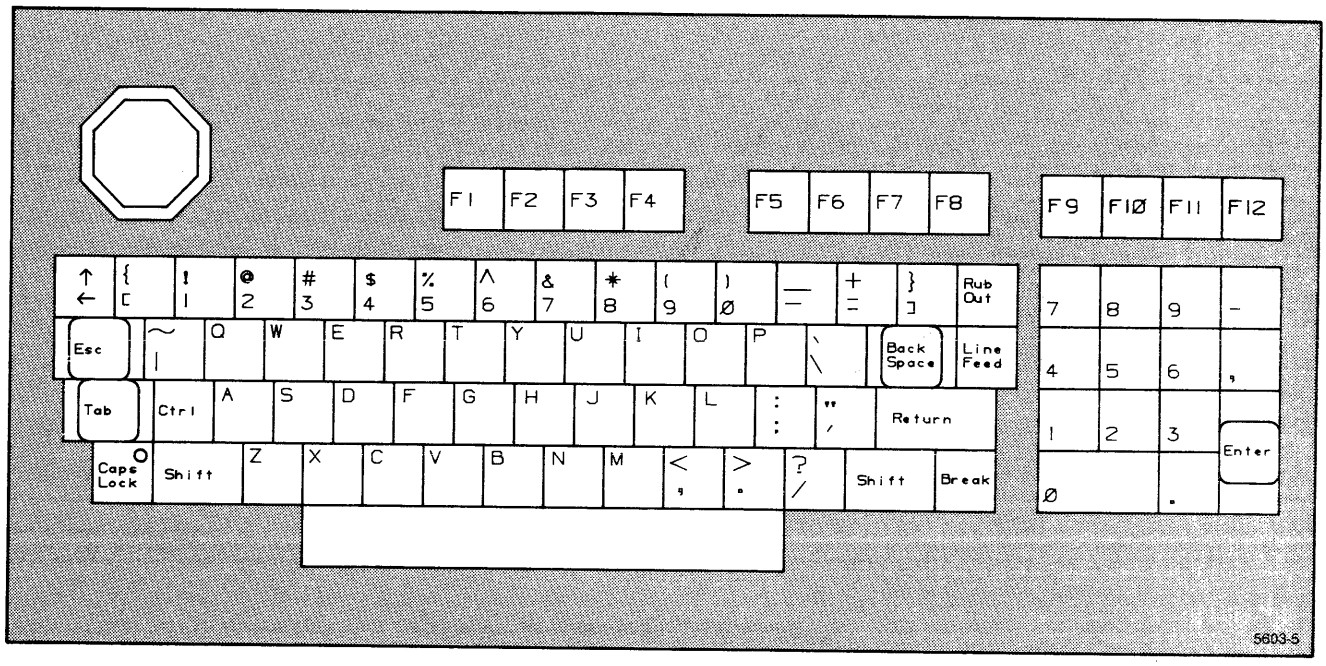

Figure 1-4. The 4405 Keyboard.

The 4405 Mouse

The 4405 mouse, shown in Figure 1-5 is the primary pointing device used with Smalltalk-80. The mouse consists of a rubber coated steel ball (to detect mouse movement over a surface) and three buttons. To use the mouse, place it on a clean surface and guide it with one hand. Use your fingers to press the mouse buttons. The mouse connector plugs into the back of the 4405 Display/CPU unit near the keyboard connector.

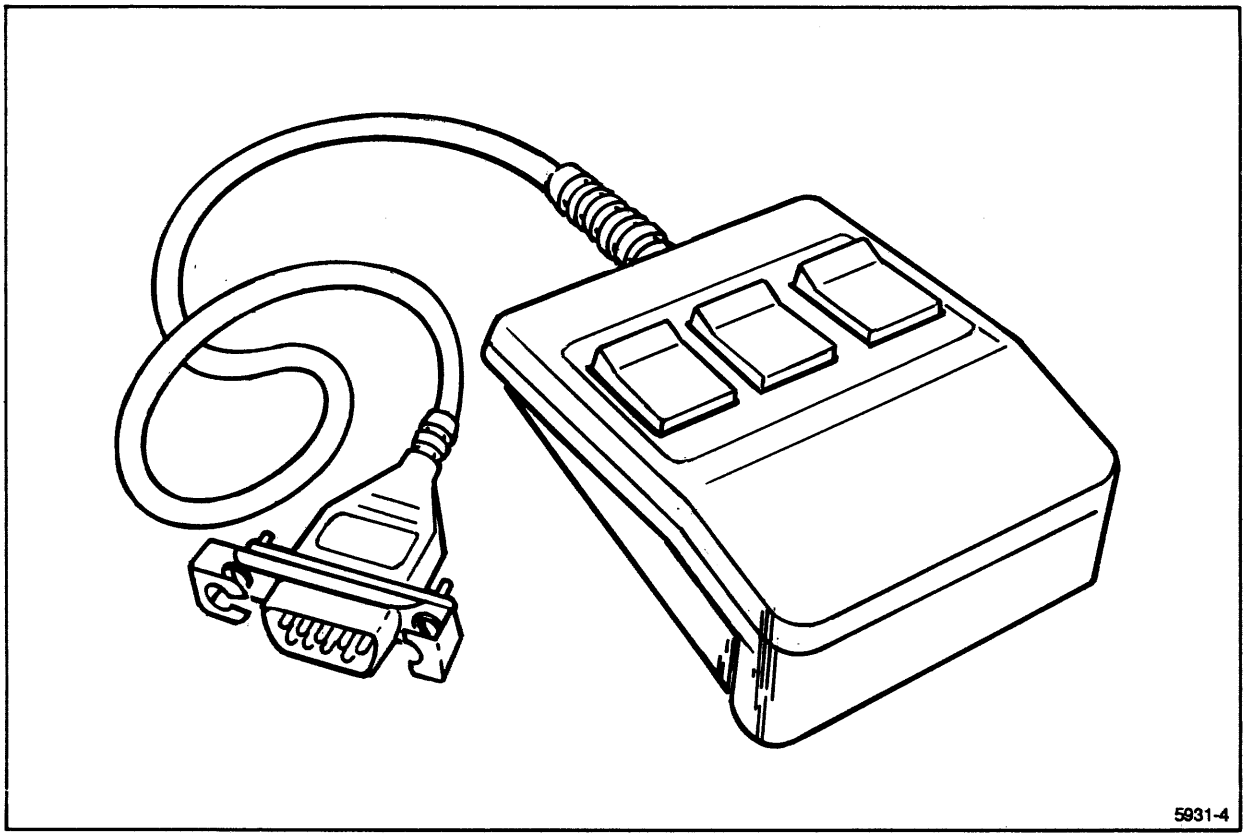

Figure 1-5. The 4405 Mouse.

THE 4405 STANDARD MASS STORAGE UNIT

Figures 1-6 and 1-7 show the front and rear of the standard 4405 mass storage unit (MSU). Figure 1-8 shows the SCSI terminator. The front panel of the MSU contains the winchester disk activity light and flexible disk drive activity light to show when these drives are in use. The rear of the MSU contains the power cord, the SCSI connector and terminator. The only controls used on the MSU are the power switch and the door handle of the flexible disk drive.

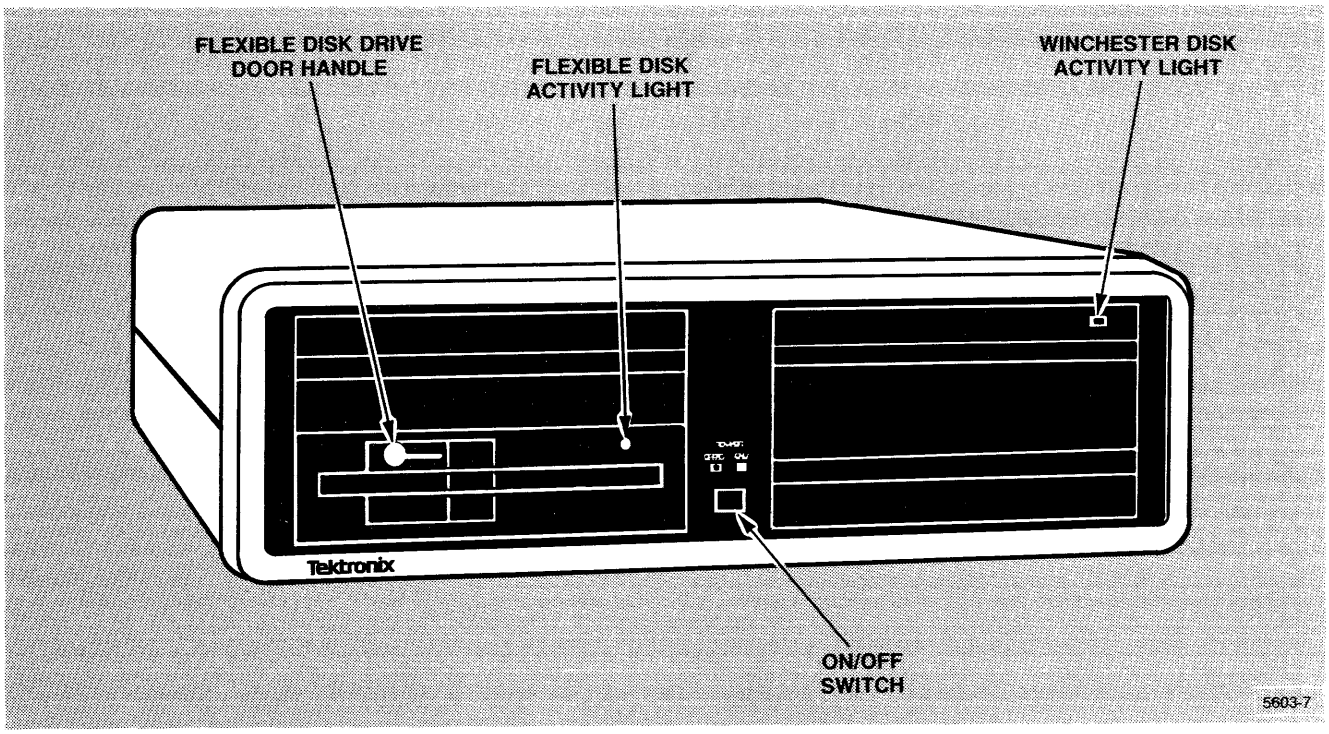

Figure 1-6. Front of MSU.

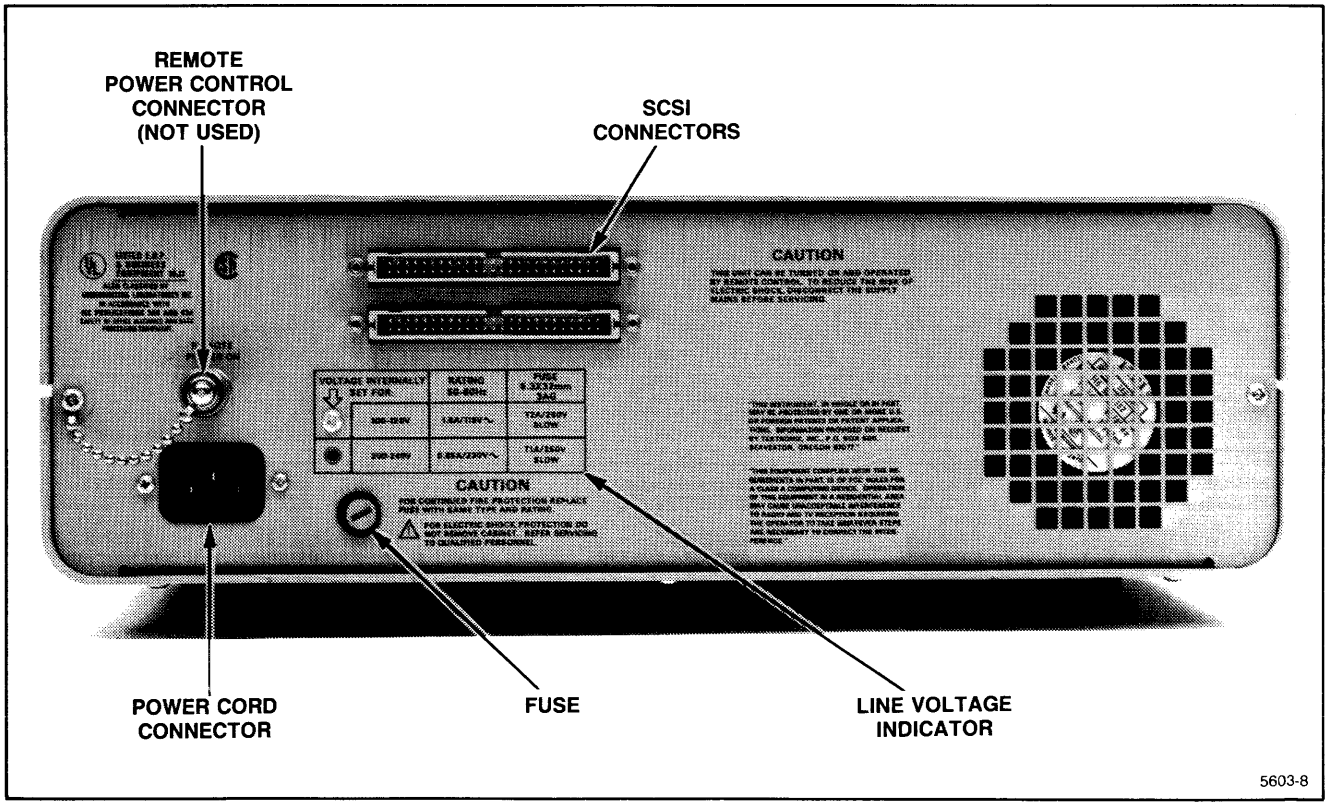

Figure 1-7. Rear of MSU.

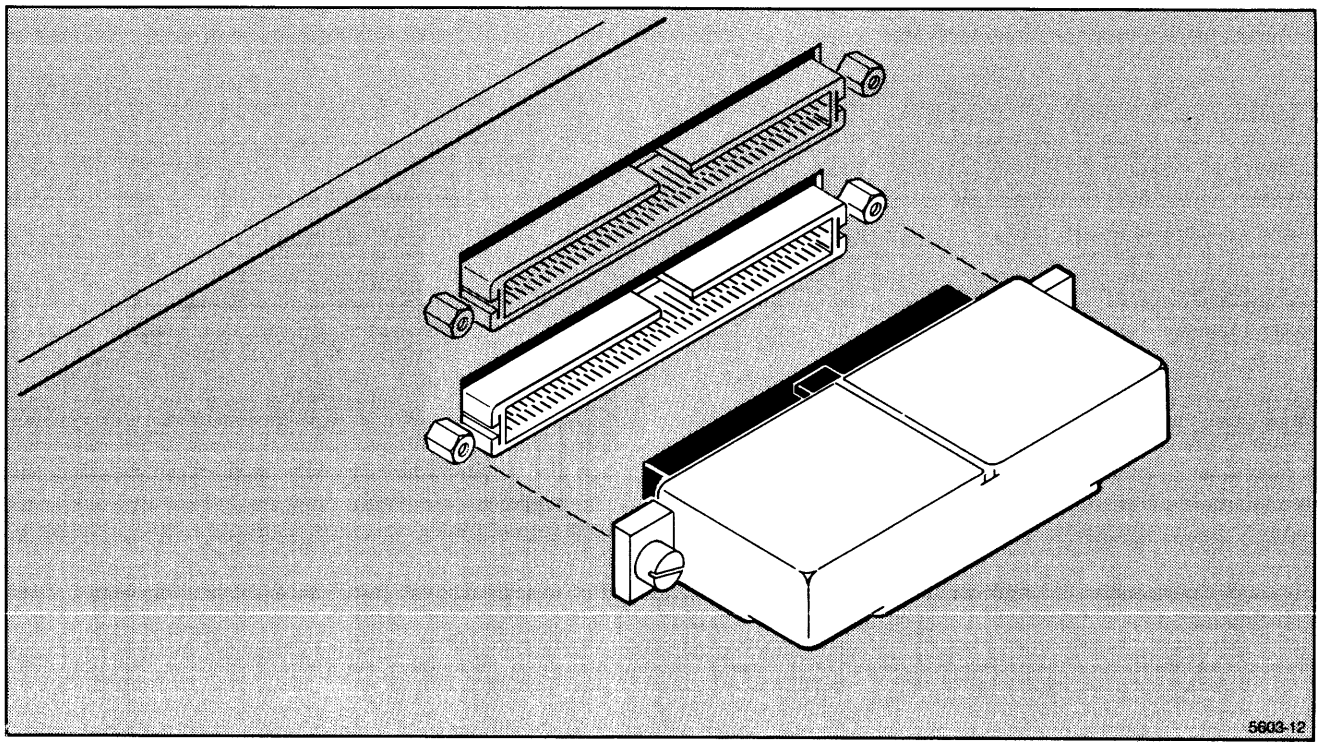

Figure 1-8. SCSI Terminator.

4405 STANDARD AND OPTIONAL SOFTWARE

STANDARD SOFTWARE

Smalltalk-80

The standard programming language for the 4405 is Smalltalk-80, version 2, developed by Xerox corporation. Smalltalk-80 is a object-oriented, general-purpose programming language that makes extensive use of the 4405 bit-mapped screen and mouse. A full description of the Tektronix implementation of Smalltalk-80 is given in the companion manual *An Introduction to the Smalltalk-80 System.*

The 4405 Operating System

The 4405 contains a small, robust operating system, based on UniFLEX®, by Technical Systems Consultants, Inc. The operating system gives one user at a time access to the 4405's resources. The operating system gives you these features:

- Single-user operation
- Multiple user accounts (with password protection)
- A hierarchical, tree-structured file system
- Multi-tasking
- An interactive shell featuring command aliasing, history, shell variables, definable search path, and command line editing
- Command file (script) execution
- User (*public*) and *system* utilities
- A relocating assembler and linking loader
- A 'C' compiler and libraries that include standard 'C' functions and a graphics library that gives access to the 4405's hardware resources
- A symbolic debugging tool for 'C' and assembly language programs
- An ANSI X3.64 terminal emulator, available both to the 4405 operating system and to an external host via a RS-232-C connection
- A line and content oriented text editor
- Printer support
- Remote file transfer and host communications

4405 OPTIONS

In addition to the standard 4405 hardware and software, many additional options are available. Some of these are:

Additional 2 MB Physical Memory

This option, an additional two Megabytes of semiconductor memory gives you a total of three Megabytes of physical memory for your 4405.

Additional 4 MB Physical Memory

This option, an additional two Megabytes of semiconductor memory gives you a total of five Megabytes of physical memory for your 4405.

ETHERNET® Interface

This option adds the hardware and software necessary for the 4405 to function in an ETHERNET network.

Franz LISP

An implementation of the LISP programming language that is widely available on UNIX systems.

Common LISP

This implementation of Common LISP includes some extensions, such as a "Foreign Function Interface," as well as the full language as documented in the book by Guy Steele, *Common LISP, the Language.*

PROLOG

An implementation of this logic programming language.

EMACS

A widely used, extensible, screen-oriented editor with mouse extensions.

Other Options

Additional 4405 options may be available in the future. Please contact your local Tektronix Sales Representative for currently available 4405 options.

Mass Storage Expansion

If you need more mass storage than is available on the standard internal 45 Mb Hard disk, use the applicable configuration of the 4944 Mass Storage Device to extend your storage. The 4944 is available in a number of configurations that add various sizes of hard disk storage and streaming tape support. The 4400 Series operating system automatically senses the presence of supported expansion units and mounts them when the system boots up.

4405 DOCUMENTATION

STANDARD DOCUMENTATION

With your 4405, you received five principal manuals. This standard set of documentation includes:

The 4405 User's Manual (This Book)

This manual should be the first manual you read. It contains the introductory material to the 4405: unpacking and connection procedures, a first-time user's session, hints on daily use of the system, discussions of software configuration, maintenance and data security. This manual contains the general "how-to-do-it" information that you'll need while getting acquainted with your new 4405. You'll probably need to use this manual extensively for the first week or so, then will probably need to refer to it only in emergencies.

An Introduction to the Smalltalk-80 System

This manual is the introduction to Smalltalk-80 as implemented on the 4405. This manual is not a "stand-alone" manual, it is meant to be used in conjunction with two other texts, *Smalltalk-80: the Programming Environment* and *Smalltalk-80: The Language and Its Implementation* published by Addison-Wesley. You will need both these books in order to use Smalltalk-80. The manual *An Introduction to Smalltalk-80* makes extensive references to the Addison-Wesley

books and details the differences between the image in those books and the image on the 4405.

The Introduction does contain some tutorial information, and an excellent overview of the language, as well as providing the intimate details that affect the language user.

The 4400 Series Operating System Reference Manual

The 4400 Series Operating System Reference Manual contains the detailed description of the 4400 Series hardware and software from the programmer's point of view. This book is the basic 4400 Series reference, and contains information on the details of commands, syntax, extensions, and the other operating system level concerns. Look in this manual for detailed information not covered elsewhere.

4400 Series Assembly Language Reference

The 4400 Series Assembly Language Programmer's Reference Manual contains details of the supplied MC68000 (including MC68010 and MC68020) assembler. This manual is a reference manual to the syntax and details of the 4400 Series assembler and system calls and libraries. This manual also contains information on the operation of the linking loader.

4400 Series 'C' Language Reference

The 4400 Series 'C' Language Reference Manual contains details of the implementation of the 'C' language and libraries. This manual also contains some tutorial information on using the "BitBlt" graphics facilities of the 4400 series machines.

DOCUMENTATION OF OPTIONS

Each software option has its own reference manual(s) that details its operation. Refer to these manuals for details of how to use these software packages.

Hardware options come with installation instructions (if the option is customer-installed) and the necessary manuals for operation of that option.

Some manuals for options are small and come without binders. Place these manuals in this binder under *Appendix C, Options.*

OPTIONAL DOCUMENTATION

Smalltalk-80 Documentation

The two books on Smalltalk-80, the "red book" and the "blue book," can be ordered from Tektronix. They are:

- *Smalltalk-80 The Interactive Programming Environment* by Adele Goldberg.
  Tektronix part number 062-8859-00
- *Smalltalk-80 The Language and Its Implementation* by Adele Goldberg and David Robson.
  Tektronix part number 062-8860-00

Service Documentation

Two service manuals are available for the 4405. These are:

The 4405 Field Service Manual

The *4405 Field Service Manual* contains information useful for a technician troubleshooting a 4405 in the field. It contains commonly used adjustments, checks, and information that enables a technician to repair a 4405 to the module level. (Modules are subassemblies that should be repaired at the factory or other service location that has specialized diagnostic instrumentation.)

The 4405 Component-Level Service Manual

This manual is designed for the experienced electronic technician and gives the information necessary to repair the 4405 subassemblies. It is not intended for use in the field; it requires test fixtures and specialized knowledge to use it.

Section 2
THE FIRST TIME

INTRODUCTION

You should now have your 4405 unpacked, connected, and ready for its first use. This section is a commented transcript of a new user trying the 4405 for the first time.

Think of this section a guided tour of the 4405 operating system. Follow along on your own machine — feel free to experiment as you go along. This section won't cover all of the features and capabilities of the operating system, but it will show you enough to get you started.

In this transcript, the actual dialog between the user and the 4405 is indented and in a different type style than the comments and explanations. The actual commands that you type are shown as they appear on the screen; enter them exactly as shown (omit the system prompt, but be sure to include the spaces). Remember, for commands to be executed, you must terminate the command line by pressing the *Return* key — this manual doesn't show that as an explicit part of this dialog. Your session should look very similar when you issue the same commands in the same order.

COMMENTED TRANSCRIPT OF SESSION

CAUTION

If you must leave the session early, type "stop" and wait for the message

"... System shutdown complete ..."

before turning off the power. Any other method of shutting down the 4405 can seriously damage the operating system disk software structure.

TURN ON THE 4405

First, you'll want to see what happens to each unit of the 4405 when you turn on one unit at a time. (Later on, you'll just want to turn on the power to both units and start working.) Press the power switch on the MSU (mass storage unit). Notice that the winchester disk activity light on upper right hand of the MSU front panel comes on until the disk gets up to speed, then shuts off.

Now, press the power switch on the Display/CPU unit. The LED in the key marked *Caps Lock* comes on, flashes for a time, then goes out. When the display screen lights up, you see in the upper left corner the display:

```
Tektronix 4405
```

If this message does not appear, try adjusting the brightness control.

The winchester disk activity light on the MSU then begins to flicker, the screen clears, and a new message appears:

```
      Tektronix 4405 Operating System  Version X.X
           Includes licensed material
For copyright and other proprietary information, use "info" command.

Total user memory = xxxxK

++
```

It usually takes a few minutes for the prompt — the *++* (double plus signs) — to appear. During this time, the 4405 has gone through a power-up self-test, loaded the operating system, automatically logged you in as a user with the user-name *public*, invoked the interactive shell (the program that talks to you), and is now waiting for you to give it a command.

The *Version X.X* message tells you what version of the operating system you have, and the *Total user memory* message tells you how much physical memory is actually available for your use. The *++* is the default system prompt, and the flashing "_" (underscore) is the cursor.

FILES AND DIRECTORIES

You have been automatically logged into the system as user *public*. The command *dir* shows you the visible files in your directory. Type *dir*:

```
++ dir
++
```

As you can see, there are no visible files in your directory (unless someone else has left them there). If someone else has used the system and created files, you'll see the file names they left.

There are a number of files in your directory that are invisible. The operating system normally doesn't show files that begin with a period. To see these files, type *dir* with the *+a* option:

```
+ dir +a
 .     ..    .login     .shellbegin   .shellhistory
++
```

Each of these invisible files has a use. We'll talk about them later. To find out more about the command *dir*, you can use *help*, the on-line help utility. In fact, you can use help to find out about itself. (We won't show the full system response here, as it is so long. Try it for yourself on your 4405.)

```
++ help help
```

(Here the system gives a list of commands for which help is available. If you want no help, just press *Return* to exit.)

```
Help with what command?
dir
```

(The system prints the help file for *dir*.)

```
++
```

We see that the command *dir* also can take an argument — the name of the directory we want to list. The fundamental directory of the system, the root, is named "/". To see what it contains, type (be sure to leave a space between *dir* and /):

```
++ dir /
```
(Here the system lists the files in "/".)
```
++
```

You need not be in a particular directory to give commands concerning files. You simply give the file's *path name* (the sequence of directories from "/") in front of the file name. For example, the file *system.boot* in the directory "/" contains the operating system kernel in addition to other components. The *info* utility shows you the version number and copyright information about this fundamental file. Type:

```
++ info /system.boot
Tektronix 4405 Operating System
Version x.x   Released MMM DD, YYYY
OS Kernel:                                    Other components:
Copyright (C), YYYY, by                 Copyright (C), YYYY, by
Technical Systems Consultants, Inc.          Tektronix, Inc.
All rights reserved.                      All rights reserved.
++
```

When you don't specify a path name, you implicitly give the files the path of your *current working directory*, in this case, the directory */public*. For example, create the following files:

```
++ create aFile
++

++ crdir aDirectory
++

++ dir
aDirectory    aFile
++
```

you've created two files, both empty in the directory */public*. One, aFile, is just that — a file. The other, aDirectory, is a directory, a special file that contains other files. Let's use the *+l* (lower case *L*) option to *dir* to find out more about them. Type:

```
++ dir +l
aDirectory        1 d rwxrwx 2  public 12:00 Nov 29,1985
aFile             0   rw-rw- 1  public 12:01 Nov 29,1985
++
```

The fields in the directory listing (from left to right) are:

1. The name of the file.
2. The size of the file in 512 byte blocks (1 for aDirectory 0 for aFile).
3. The file type (d for directories, b for block devices, c for character devices, and blank for files). The special file, aDirectory, has a *d* for directory; aFile is a normal file (blank).
4. The permissions for the file (rwxrwx for aDirectory, rw-rw- for aFile). The first three characters are the read, write, and execute permissions for the file's owner, the next three

the permissions for all others. See *perms* in the *4400 Series Operating System Reference Manual* for a complete explanation of permissions.

5. A count of how many other files are linked to that file (2 for aDirectory and 1 for aFile).
6. The file's owner (user *public*).
7. The time and date the file was created (or last modified).

MOVING AROUND IN THE DIRECTORY

Use the command *chd* to change your location in the directory structure (your current working directory). *chd* with no argument returns you to your home directory. The command *path* tells you the name of your current working directory:

```
++ path
/public
++

++ chd aDirectory
++

++ path
/public/aDirectory
++
```

If you now create a file, *newFile*, it will have a path name */public/aDirectory/newFile*. To access this file from outside the directory */public/aDirectory*, you must give the path name of the directory */public/aDirectory*, then a /, and as the last element, the name of the file, *newFile*. Let's try it and see.

```
++ create newFile
++

++ dir newFile
newFile      0  rw-rw- 1 public  12:05 Nov 29,1985
++
```

Notice that when you give a file as an argument to *dir*, it returns the same information the *+l* option does. Now, to change directories try *chd* without an argument. It will get you back to your home directory from wherever you might be. You can also combine more than one command on a single command line by separating them with a semicolon. Try that also:

```
++ chd;path
/public
++
```

Where's the file *newFile* we created a moment ago? Let's see:

```
++ dir newFile
*** Error: File doesn't exist.  While Getting file status
on file "newFile"
++
```

It's obviously not in the directory */public*. Let's try the file's path name:

```
++ dir /public/aDirectory/newFile
/public/aDirectory/newFile      0  rw-rw- 1 public  12:05
Nov 29,1985
++
```

The shell also contains a *directory stack*, a list of directories that you can move among. To move to the directory */bin* and save your present directory, use the *pushd* command:

```
++ pushd /bin;path
/bin  /public
/bin
```

You are now in the directory */bin* with the directory */public* stored on the directory stack. To get back to the last stored; directory, use the command *popd*:

```
++ popd
/public
++ path
/public
```

SOME *shell* FEATURES

The interactive shell contains many capabilities to help you in your work. Let's look at a few of them.

history

The shell keeps an ongoing record of the last commands you've given it. This command file (*.shellhistory*) gets updated when you log out of the system. You can recall and edit these commands to save yourself typing.

To recall your previous commands, Type *history* and, if you typed exactly the commands that were given earlier in this session, you see:

```
++ history
1:  dir
2:  dir +a
3:  help help
4:  dir /
5:  info /system.boot
6:  create aFile
7:  crdir aDirectory
8:  dir
9:  dir +l
10:  path
11:  chd aDirectory
12:  path
13:  create newFile
14:  dir newFile
15:  chd;path
16:  dir newFile
17:  dir /public/aDirectory/newFile
18:  pushd /bin;path
19:  popd
20:  path
++
```

Obviously, if you typed something else, your history will differ. To recall the last command you entered, press <Ctrl-P> (hold down the key marked *Ctrl* and press the key marked *P*) or press the top of the joydisk. The last command reappears with the cursor under the first character of the command. Each time you press <Ctrl-P> (or the top of the joydisk), the command line becomes the previous history entry. To move the command line forward in your history, press <Ctrl-N> (or the bottom of the joydisk). Move up and down in your history until the command line reads *path*. Press *Return*, and the 4405 executes this command again:

```
++ path
/public
++
```

Try *history* again. You'll see an additional line:

```
++ history
1:  dir
 ...
18:  pushd /bin;path
19:  popd
20:  path
21:  path
```

Notice that *history* doesn't get stored.

Command Line Editing

You can edit commands, whether you are in the process of entering new ones or have retrieved old ones. Let's edit an old command.

1. Press the top of the joydisk (we'll call that action joyup — it has the same effect as <Ctrl-P>) several times. Notice how you step back through the commands.
2. Hold the top of the joydisk down (it automatically repeats) until you get to the earliest command stored. (The beeping of the bell tells you that you've exhausted your stored history.)
3. Step down through the commands with either <Ctrl-N> or the bottom of the joydisk (joydown).
4. Move down until the command line is blank (at the bottom of your history). Let's take a slight detour and look at some interesting features about recalling history:
   a. Type the letters *cr* on the command line, then stop. Do not press *Return.*
   b. Move up and down through your recalled history with joyup and joydown. Notice that you can only recall three commands: *create newFile*, *crdir aDirectory*, and *create aFile* in addition to the *cr* you entered.

      You can recall only those commands that match the characters to the left of the cursor on your present line — those that begin with *cr.*
   c. Move the cursor one character to the left on the command line (press <Ctrl-B>) so it is just under the *r* in *cr.*
   d. Try moving through your history again. Now you can retrieve any command that begins with *c.*
5. Use joyup and joydown until the command line becomes:

```
++ crdir aDirectory
```

6. Move the cursor around on the command line by using using <Ctrl-F> or Joyright and Joyleft or <Ctrl-B>.
7. Put the cursor on the character *D* in *aDirectory* and press <Ctrl-D> to delete the *D.*
8. Press the *Back Space* key to delete the *a* and replace it with a lower case *d* by pressing the *d* key. At this point, you've nearly finished editing the command line. Notice that we have two types of single character deletion: Back Space and <Ctrl-D>. In addition, you have several more ways to move the cursor on the command line. Press and release the *Esc* key then press *F* to move the cursor right one word. <Esc-B> (the same sequence with the *B* key) moves the cursor left a word. <Ctrl-E> moves the cursor to the end and <Ctrl-A> moves the cursor to the beginning of the command line.
9. Press <Ctrl-E>. The cursor moves to the end of the word *directory* in the command line.
10. Type in *B* and press *Return.* It looks like this:

```
++ crdir directoryB
++
```

You've successfully retrieved a command, edited it to make a new command, and then executed the new command. You'll find that, in many cases, it's much faster to retrieve an old command and edit it than to type in a new command. To see the results of the command you just issued, type:

```
++ dir
aDirectory     aFile     directoryB
++
```

Your history is now:

```
++ history
1. dir
 ...
22:  crdir directoryB
23:  dir
```

When you are editing a command line, should you want to enter a control character such as <Ctrl-P> rather than executing it, enter the quote character, <Ctrl-Q>, followed by the control character you want to insert. (We'll use this later with environment variables and *alias*.) Table 2-1 shows the moving commands you can use on the command line.

Table 2-1
Moving Commands

Function	Control and Escape Keys	Joydisk Equivalents
Move up one line	<Ctrl-P>	joyup
Move down one line	<Ctrl-N>	joydown
Move right one character	<Ctrl-F>	joyright
Move left one character	<Ctrl-B>	joyleft
Move to end of line	<Ctrl-E>	
Move to beginning of line	<Ctrl-A>	
Move right one word	<Esc-F>	
Move left one word	<Esc-B>	

Table 2-2 shows the commands for deleting characters, words, and the entire command line.

Table 2-2
Deleting Commands

Function	Control and Escape Keys
Delete character left	Back Space <Ctrl-H>
Delete character right	<Ctrl-D>
Delete word left	<Esc-H>
Delete word right	<Esc-D>
Delete line to right of cursor	<Ctrl-K>
Delete entire line (Restore it if deleted)	<Ctrl-U>

This discussion has covered a lot of territory so far. You should probably clean up your directory and remove the files and directories we've left there. To test your own understanding, remove the files and directories you created in */public*. At this point, you should test your understanding (and get some command line editing practice) by observing the following restrictions:

- Remove all files and directories in /public one by one.
- Type the word *remove* on the command line, then press *Return*. You will see a prompt for the syntax of the *remove* command.
- Form the other commands by recalling *remove* from your history and edit the command line. Don't just type in your commands.

Hint: Don't forget *help* and the options to *remove*.

The *shell* Environment

Your shell environment is the way in which the shell responds to your commands. You can change the environment in two ways: by setting environment variables and by aliasing commands.

Environment Variables

The shell maintains a list of *environment variables*, (these are *not* the same as Unix *environment variables*) some of which are bound to special keys or functions and others that simply store values. To see what these bindings are, type *set* or *env*.

In response to *set* or *env* with no arguments, the 4405 displays a list of environment variables that are already set. You've met some of these earlier.

Control characters are shown in *snoopy* form; that is, the two-letter combination of C and R stands for the carriage return. You've seen the joydisk variables before. It should be obvious that the lower case versions are unshifted, while the upper case versions are shifted. Joydisk variables that begin with *C* are the control versions of the variables. The ARROW and BREAK key variables are obvious. The function keys are represented by f1 through f12, while the shifted function keys are F1 through F8. Defining these variables effectively programs these keys for

you while you are executing the shell.

PATH is the set of directories that the shell searches before deciding that it cannot recognize a command, and PROMPT is the string that it displays for the system prompt. (To put spaces in a string, you have to enclose the string in single or double quotes — otherwise the shell just takes the first word of the string.) Let's change a few variables and see what happens.

```
++ PROMPT="public++ "
public++

public++ f5=dir
public++

public++ F5='dir +al'
public++
```

Notice that we didn't have to put quotes around *dir*, the defining string for f5, but did around the string for F5 (shifted f5) as it contained a space. Now press function key f5. (Follow with *Return* if you want to do the function, <Ctrl-C> or <Ctrl-U> if you don't.) Now try pressing function key F5 (hold the shift key down and press f5).

Notice that pressing the keys f5 and F5 have the same effect as typing in the commands you bound to them, and that you have to press *Return* to execute the commands.

It is also possible to put a *Return* in the definition of key F5. Since we want the *Return* in the string, we can't just press *Return* (that executes the command line). We use a pair of characters, the quote character (<Ctrl-Q>) followed by a lower-case *n*, to embed the character *Return* in the defining string. It looks like this:

```
public++ F5='dir +al<Ctrl-Q>n'<Return>
public++
```

Type *set* to see the definition, then press F5 to try the programmed function.

To change or remove a shell environment variable definition, you can define it to something else (it will overwrite the old definition) or you can use *unset*:

```
public++ unset f5
public++
```

Alternately, you can define the environment vairable to be null (`F5=`) to remove an environment variable.

Aliases

In addition to environment variables, the shell maintains a list of aliases. When you enter a command line, the shell checks the first command against its alias list, and if the command is aliased, executes the underlying command. To see your list of aliases, type *alias* followed by *Return*.

Let's alias a command, then remove the alias.

```
public++ alias showMe 'dir +as'
public++

public++ showMe
.
..
.home?
.login
.shellbegin
.shellhistory
public++

public++ unalias showMe
public++

public++ showMe
showMe: command not found
public++
```

Like *unset*, you can use *unalias* with the *+a* option to remove all aliases from your shell.

You can use argument designators to extract arguments from commands. With environment commands, the designators apply to the last command executed, while with aliases, they apply to the current command line. To pass all the arguments to the basic command, let's alias *ll* and accept all arguments to it.

```
public++ alias ll 'dir +a $*'
public++
```

Now, to see the contents of three directories in long form, type:

```
public++ ll /etc /bin .
```

Watch carefully as the system displays the directories of */etc*, */bin*, and ., your current directory.

Saving Definitions

You can define environment variables and aliases in a text file, then use the *set* command to pass them to the shell. To set your environment back to what you started with, type:

```
public++ set .shellbegin
++
```

If you look at your environment with *set*, you'll find that PROMPT has been reset. If you haven't used *unalias +a*, any aliases you made still exist since they aren't mentioned in *.shellbegin*. The shell sets the environment automatically from *.shellbegin* when the shell starts up at login time or when it is subsequently invoked. You can edit *.shellbegin* to define any environment variables and aliases you want to have whenever you work on the 4405.

The file *.shellhistory* also saves your history, aliases and variables from one login to the next.

In addition, if you have any other tasks you want to do every time you login, create a script (basically, just a list of commands in a text file — see the O/S reference manual under *script*) and put that script in the file *.login*.

CONTROLLING THE TERMINAL EMULATOR

The 4405 communicates with you via a terminal emulator. This emulator is ANSI X3.61 compatible with some extensions. You can change some of the operating attributes via either ANSI command sequences or by ussing the *conset* command.

ANSI Commands

You can issue ANSI commands to the terminal emulator via the *echo* command. For example, you can change the cursor from the default underline to a block by issuing the command:

```
++ echo '<Ctrl-Q><Esc>[>31h'
++
```

Where the <Ctrl-Q> tells the shell to accept the next character literally.

See the *4400 Series Operating System Reference Manual* under "Terminal Emulation" for details of the supported ANSI commands.

Other Terminal Attributes

In addition to the ANSI attributes, the 4405 terminal emulator has a number of options that are non-ANSI. These include options that allow you to enable or disable raw mode, character echoing, expansion of tabs, action of the <Back Space> key, positive or negative video, fonts, and other attributes. To invoke these attributes, use the *conset* command. (*Conset* without any options displays the current state of the terminal emulator.)

To see the full range of options, type:

```
++ help conset
(the help message for conset appears)
++
```

To change from positive video (black letters on white background) to negative video (white letters on black background), type:

```
++ conset -video
++
```

To change back to positive video, type:

```
++ conset +video
```

RS-232 Terminal Emulation

The 4405 supports an RS-232 port. To make the 4405 emulate an RS-232 terminal, connect the RS-232 cable from the 4405 to a modem or computer port and give the command *remote*. The *remote* command contains provisions for capturing text in a buffer, and a file transfer protocol that can be invoked from a host computer. The source code for an example of the host computer software is given in the file */samples/xfer.c*. This code is suitable for use on a computer using the

Unix operating system and is unsupported code.

The RS-232 port options, such as baud rates, flagging, stop bits, parity, and CTS flagging, are set by using the *commset* command. *Commset* without any options shows the current state of the RS-232 port.

ENDING THE SESSION

Experiment with the operating system commands. Read the O/S reference manual to get an idea of the commands that are available and how they work, then try them. When you are finished, type *stop*.

```
++ stop
```

Don't forget to shut the power off on both the MSU and the Display/CPU units.

Section 3
USING THE 4405

INTRODUCTION

At this point, you should have gone through the first-time user's exercise in Section 2. If you have not done so yet, you should do so before proceeding.

This section covers normal day-to-day use of the 4405 from a user's standpoint. This discussion takes a broader look at the operating system and how it operates than was covered in Section 2. This discussion assumes that you have not logged in as user *system*, are logged in under *public*, and are performing routine tasks.

POWER ON AND SYSTEM BOOT

To start using your 4405, turn on the power switches on both the Display/CPU unit and the Mass Storage Unit. The system then goes through a power-up self-test and initialization procedure, then logs in user *public* if its password has not been set.

Although this procedure is fully automated, quite a lot happens during this short time. If you want to reconfigure the system, or customize your environment, you can do so by altering files used in this procedure. Let's follow the boot process and see what happens.

POWER-UP SELF-TEST AND BOOT

When you first turn the power on to the Display/CPU unit, control goes to a program located in ROM (read-only memory) located on the processor board. This program executes the power-up self test. It checks and initializes the main memory (RAM), the various interfaces, and then attempts to boot the system.

The boot ROM code looks for a file named *system_4405.boot* on the winchester disk. If the file *system_4405.boot* is present on the winchester, the boot ROM attempts to load and execute it. (The system also checks in various other places for boot files, as explained in Section 5's discussion of self-test.) If *system_4405.boot* is not present, the 4405 asks you to enter the name of the boot file.

You can also put the 4405 into an *interactive boot* from self-test. This option is also explained in Section 5 in the discussion of self-test.

BOOTING THE SYSTEM

When *system_4405.boot* executes, it loads the operating system kernel and performs some diagnostics on the file structure. It looks for telltale signs that the system had not been shut down cleanly, and if it finds them, it executes a *system* utility called *diskrepair*. *Diskrepair* makes a thorough analysis of the disk file structure and repairs any defects it finds. It also (since it executes in the verbose mode) prints messages on the screen to let you know what it is doing. *Diskrepair* may shut down the system and ask you to reboot (press the Reset button) if it makes substantial repairs.

If the system had been shut down cleanly (or if *diskrepair* was able to repair the disk without needing to shut the system down) the 4405 then begins the login process for user *public*.

THE LOGIN PROCESS

CHECKING THE PASSWORD FILE

The 4405 begins the login process by first checking in the file */etc/log/password* to see if the user-name (*public* on power-up/reset or the user-name from a *login* prompt) is valid. If the user name is valid and the name has no password associated with it, the system logs the user in.

If the user-name is not valid, or if the user has set a password, the 4405 then prompts the user for a password. (No password will be acceptable for an invalid user-name.) If the user-name and password are valid, the 4405 then logs in the user.

To login from another user name, issue the command *login*
followed by the user name to login under:

```
login <user_name>
```

USER INITIALIZATION

On bootup, or if you login with the user-name *public*, the 4405 runs the shell script */public/.login*. The default version of this script is empty — you can enter whatever commands you want *script* to execute every time you login.

Next, the interactive shell reads its own initialization files, *.shellbegin* and *.shellhistory*, to define your shell environment. This file sets up environment variables, aliases, and restores your history. Finally, the shell issues you a prompt and you are ready to begin.

If you login under another user-name, the 4405 looks for the file *.login*, and the shell looks for the file *.shellbegin* in your home directory.

SETTING PASSWORDS

You can set the password of your home directory with the *password* command. Simply type the command followed by a <Return>, and the system will prompt you for a password. After you type it, the system prompts you to type it again, then sets the password for your user-name. Only the user logged in as *system* can set passwords for others. If you forget your password as a user, a person logged in as *system* can change your password. If you forget the password for *system*, the forgotten password can be removed with a utility on the *DISKREPAIR* diskette (see Section 5, *Recovery and Rebuild* for information on this utility.)

STOPPING THE SYSTEM

STOP

Whenever possible, you should back up your files before stopping the system. For a discussion of backup strategy, see Section 3, *Software Maintenance.*

The command to issue when you want to shut down the 4405 is *stop*. This command terminates the system gracefully; it flushes the contents of buffers to disks, closes open files, terminates background processes, and completes other housekeeping that is necessary to gracefully shut the system down.

The syntax of this command is simply:

```
stop
```

The 4405 maintains several files and buffers while it is running. If the system is not shut down gracefully, some of these files may exist after shut-down, and some of the files it maintains may not contain correct information.

WARNING

In extreme cases, simply turning off the power or pressing the reset button may damage the file structure to the point that the system is no longer usable. To prevent this, it is essential to stop the system gracefully before turning off the power.

POWER OFF

After you issue the command *stop*, the system begins the shutdown process. This procedure may take a few moments.

Wait. After a few more moments (the time depends on the number of background processes to shut down, and other housekeeping jobs) the system gives you the message:

```
... System shutdown complete ...
```

At this point, it is safe to turn off the power to both the MSU and the Display/CPU.

THE OPERATING SYSTEM AND UTILITIES

OVERVIEW

The 4400 Series Operating system consists of a small *kernel* that can execute a number of operations. Most commands and utilities are stored in the file system. As a point of fact, the kernel does not deal directly with the user — the user interface is via a utility called *shell.* If the user input matches a file found in the *search path* (the sequence of directories the *shell* searches for commands), the kernel loads and executes that file if it is an executable file, or executes it via

another utility called *script* if it is a text file with the proper permissions. The search path for *shell* is stored in the environment variable *PATH*. You can view the shell's search path by issuing the command:

```
++ env $PATH
```

The two utilities, *shell* and *script*, do not follow the same search path. The path in *shell* is set by the string argument to the environment variable *PATH*, and that in *script* is set via a built-in command. See the discussion in the *4400 Series Operating System Reference Manual* on *script* to learn more about its search path.

COMMANDS AND COMMAND SYNTAX

The general form of a 4400 Series Operating system command or utility is just the name of the file containing the command followed by a carriage return. In addition, most commands take *options* or *switches* (sequences of characters, usually preceded by a '+' character) that modify the action of the command, and some may require arguments or parameters to the command. The *4400 Series Operating System Reference Manual* gives a full listing of the options, parameters, and arguments to each of the operating system commands and utilities.

Options

Most options are introduced by the plus (+) character immediately before the option character. When a command has more than one single-character option, you can put multiple option characters together on one line following the "+".

For example (ignoring arguments for now), the command *dir* takes several extensions — a, b, d, f, l, r, s, t, and S. These commands alter the format in which *dir* displays the directory. If you want to see a directory of normally invisible files (those that begin with "."), you add the extension *+a*. If you want the directory to print the files one per line, add the extension +s.

If you want to do both (show all the files, including the invisible files, one per line), the command is:

```
dir +as
```

(or alternatively)

```
dir +a +s
```

Arguments and Parameters

Although there is a subtle difference between the terms "arguments" and "parameters," they are often used interchangeably to refer to the additional information needed by a command. Arguments usually take on a default value if you omit them, but the system will require you to supply parameters.

For example, the command *chd* (the command to change directories) takes a directory name as its argument, and defaults to your home directory if you do not supply the argument. The command *perms*, on the other hand, prompts you with an abbreviated syntax line if you omit the parameters.

Options That Take Arguments

Some options take arguments, such as the *w* option (wait) for the *status* command. When an option takes argument (other than its default), the option that takes the argument must be the last option in the option string (the sequence of characters following the "+"). The option taking an argument must have a "=" immediately behind it, followed by the argument value. Some commands can contain only one option string, while others may have multiple strings.

For example, to display the system status you can use the *status* command. To display the system status every 30 seconds, you can use the command:

```
status +w=30
```

To display more information, you could use:

```
status +alsxw=30
```

(or one of the equivalents)

```
status +w=30 +alsx
status +al +sx +w=30
status +a +l +sxw=30
```

... etc.

MANUAL SYNTAX CONVENTIONS

Throughout this manual and the other manuals for the 4400 Series products, the following syntax conventions apply:

- Words standing alone on the command line are *keywords*. They are the words recognized by the system and should be typed exactly as shown.
- Words enclosed by angle brackets ("<" and ">") enclose descriptions that you must replace with a specific argument. If an expression is enclosed *only* in angle brackets, it is an essential part of the command line. For example, in the line:

  ```
  adduser <user_name>
  ```

  you must specify the name of the user in place of the expression <user_name>.

  In addition, specific keyboard keys for you to press are shown in angle brackets. For example, <Return> means "press the key on the keyboard marked 'Return,' while <Ctrl-C> means "hold down the key on the keyboard marked 'Ctrl', press 'C' and release them both."
- If the word "list" appears as part of a term, that term consists of one or more elements of the type described in the term, separated by spaces. For example:

  ```
  <file_name_list>
  ```

  consists of a series (one or more) of file names separated by spaces.
- Words or expressions surrounded by square brackets ("[" and "]") are optional. You may omit these words or expressions if you wish.

FILE STRUCTURE

The 4400 Series file system is a tree-structured hierarchy. Entries are files, some of which are directories. File identifiers consist of a *path name*, the sequence of files beginning at the *root*, or fundamental directory of the hierarchy, continuing through each subdirectory to the actual file name.

The file name is the identifier for a file in a particular directory. The path name is the chain of directories that enables you to find a particular file in the entire directory structure. The full name of a file is the path name of its directory with the file name after a separating "/" character.

Directory Contents — *dir*

The *dir* command lists the contents of a directory. Without an argument, it lists the contents of directory it takes as a default, or your *current working directory*; with one or more arguments, it lists the contents of the directories you give as the arguments. To identify directory entries within a directory, use the *dir* command with the option *+l*.

For example, let's look at the directory of "/", the root directory. Give the command:

```
dir / +l
```

The system shows the directory entries, one per line. Following the name field is a number (the file size in 512 byte blocks), then in some files a single character (when this character is a *d*, the file is a subdirectory), the link count, permissions associated with the file, the file's owner, and the date associated with the file.

Each directory contains two "relative" subdirectories. The first (.) refers to the directory itself. The second (..) refers to the parent of that directory. (Files beginning with a period are normally invisible; to see these files, use the *+a* option.)

You can use the "." and ".." designations from any working directory. For example, if you were down a directory tree, you could see the contents of the grandparent of your directory with the command:

```
dir ../..
```

Moving Around The Directory Tree

When you login to the system, the 4405 makes your *home directory* (this directory is defined in your entry in the password file */etc/log/password*) your current working directory. You can think of your current working directory as your location in the directory tree and changing the default directory as moving to different locations in that tree.

Your current location is always available with the command *path*. For example if you are in the directory */neat/stuff*, and type *path*, the system responds with */neat/stuff*.

To change your position, type the command *chd*, with or without an argument. Without an argument, *chd* returns you to your home directory; with an argument, it moves you to the specified directory.

In addition, *shell* maintains a directory stack. You can move about the directory stack by pushing the current directory onto the stack and moving to a new directory with the command *pushd*. You can view the contents of the directory stack with the command *dirs*, and change back to the directory on the top of the stack with the command *popd*.

Adding and Removing Files

You can create empty files with the utility *create*, and directory files with the utility *crdir*. In addition, many applications programs (such as the text editor) will automatically create files for you. To remove files or directories, you should use the utility *remove*. To remove an empty directory you can use the *+d* option; to remove a directory and all the files in it, you can use the *+k* option.

WILD CARD EXPANSION

Both *shell* and *script* perform *wild card expansion* — they will match a series of characters against *wild cards* on the command line. The most commonly used wild card is "*", which stands for any sequence of any characters. For example, the pattern *a*x* matches *ax*, *a_fox*, *amazing_clox*, or any other letter combination that begins with a and ends with x. You can match single characters with "?" as a wild card. (The operating system, and most utilities, treat upper and lower case letters as distinct. *A* is ¯Bnot the same as *a*.)

See the *4400 Series Reference Manual* entries under both *shell* and *script* for a more thorough discussion of wild card expansion.

MULTI-TASKING

The 4400 Series operating system is capable of *multi-tasking*, or seeming to do more than one job at a time. In reality, of course, it does one job, switches to another, then back — usually quickly enough to give the illusion that it is performing the tasks simultaneously.

The interactive shell and *script* control multi-tasking. To run a job in the background, you can start or end a *shell* command line with the character "&", while for *script* to do background processing, the "&" must be the last character on the command line.

USER COMMANDS BY FUNCTION

The following list summarizes the user commands by function. For details on the actual command syntax, use the *help* utility or see the *4400 Series Reference Manual*, which contains an entry for each command.

FILE MANIPULATION

The following user commands and utilities are primarily used with individual files.

copy

Make copies of directories or individual files.

create

Create new empty files.

edit

Line-oriented text editor.

link

Create a link to an existing file.

list

List the contents of text files.

move

Move files to another location or rename them.

remove

Remove files or directories.

rename

Rename a file.

FILE PROCESSING

The following commands are used for file processing.

compare

Compare and list the differences between two files.

dump

Formatted file dump in hexadecimal and ASCII.

filetype

Print out message about the type of a file.

find

Search for a pattern within a file.

info

List information about a binary file. This command usually lists copyright information and revision numbers of the file given as an argument.

touch

Update the *last-modified* time on a file.

tail

Display the last 250 (or specified number of) characters of a file.

DIRECTORY MANIPULATION

The following commands and utilities are primarily used when working with directories.

chd

Change the current working directory.

crdir

Create new directories.

dir

List the names of files in a directory.

path

Display the path name of the current working directory.

SYSTEM ACCESS AND STATUS

date

Display date and time. User logged in as *system* may modify date and time.

dperm

Adjust the default permissions for newly created files.

exit

Terminate a subshell and return to the calling shell.

help

Obtain information about commands.

login

Access the file system as a user.

owner

Change the owner of a file.

password

Set the password of a user for login.

perms

Set the permissions for files.

status

List the status of current tasks or processes. Useful to find background jobs that may be running.

stop

Gracefully shut down the system.

DISK MANAGEMENT

backup

Create archival backups of files and directories. Also useful for interchanging programs between 4400 Series machines.

diskrepair

Check and optionally repair the logical structure of a disk system. (See Section 5, *Recovery and Rebuild* for hints on using diskrepair.)

format

Format and write a file system on a floppy disk.

free

List the amount of free space on a disk.

restore

Restore files from a backup format floppy disk.

COMMAND EXECUTION

echo

Display arguments to this command.

int

Send an interrupt to a task. (Used to kill background jobs.)

jobs

List information about the user's background tasks.

script

Command language used to execute commands from a text file.

shell

Interactive system command language.

wait

Wait for background jobs to complete.

COMMUNICATIONS

commset

Examine or modify parameters for the communications port.

conset

Examine or modify parameters for the console (display and keyboard).

remote

Connect through the communications port to a host computer.

PROGRAM DEVELOPMENT

asm

The relocating assembler.

cc

The 'C' compiler.

debug

General purpose debugger.

headset

Modify task parameters in an executable file.

libgen

Generate a library file.

libinfo

List information about a library file.

load

Linking loader to produce object files.

smalltalk

Run the Smalltalk-80 interpreter and environment.

strip

Remove symbols from an executable file.

relinfo

List relocation information in an object file.

update

Execute commands based on dependencies in a makefile.

Section 4
SOFTWARE MAINTENANCE

INTRODUCTION

As in any other complex software system, the 4405 requires software maintenance. If you are the sole user of your system, you must perform these tasks, which on a larger system are done for you by a system manager. We prefer not to use the term *system manager* with the 4405 (although the term might slip in here and there), but to think of the person or persons who perform these duties as simply the user *system*. The person who is logged in under *system* has more power and much greater responsibility for keeping the system running than someone logged in under *public*.

This section discusses the responsibilities and privileges that the user logged in under *system* must be aware of.

Many utilities will allow only the *system* user to invoke them. These utilities, kept in directory */etc*, are the maintenance tools that user *system* uses to repair the disk structure, add and delete user accounts, and other system housekeeping chores. Most of these utilities are either dangerous or inappropriate for indiscriminate use.

You should be confident of your command of the operating system as a user under *public* before you log in under *system*. Many utilities and procedures require knowledge and skill to use them safely.

CAUTION

User ***system*** *has very few safeguards against accidents. As* ***system****, in directory "/", you can destroy the operating system by simply typing "remove * +k".*

THE FACTORY CONFIGURATION

As supplied from the factory, the 4405 is ready to plug in and run. Two user accounts, the general user *public* and the housekeeping user *system* are predefined. Neither account has a password. Either user can set its own password, and *system* can set a password for *public*. You need not set passwords for either account unless you feel that protection is necessary.

The general user utilities are contained in directory */bin* and the *system* utilities are contained in directory */etc*. User *public* and any other general user accounts will search only their home accounts and */bin* for command names. User *system* will also search */etc* for command names. You can change the search path by changing the shell environment variable PATH. To see the current search path, type:

```
++ echo $PATH
(the system responds with the current search path)
```

To add or remove search paths, you can change the environment variable PATH via the *set* command or use *addpath* and *rmpath*. An example of using *set* is given later in this section.

USER *public*

If you are the only person using the 4405, you should do most of your work logged into user *public*. This is the normal user mode of operation -- you are protected against causing system-wide devastation with a few accidental keystrokes. You have the full resources of the system available, and are protected only against destroying another's files or using potentially destructive utilities. If several persons are using the system, each with an individual user name, or if you are using different user names to organize your work, each of these general users will have privileges similar to those of user *public*.

File Protection and Ownership

You are the owner of all files that you create as a particular user. You can change the permissions on these files (and no others) with the *perms* command. If more than one *public* level user exists on the system, each user can allow or deny access to his or her owned files via *perms*. You can allow or deny access to your directories by allowing or denying *x* (execute) permission to a directory. User *system* can override or change these permissions.

Passwords

You can specify or change a password for your user name by using the *password* command. This inhibits anyone from logging into your account unless they know the password. User *system* can change or delete your password. You can leave the password off your account if your situation does not warrant using passwords.

If you forget your password, it can by changed by user *system* using the *password* command.

Backing up User Files

You should (depending upon your system organization) back up your own files frequently. By doing so, you reduce the possibility of losing large amounts of work. You need not back up your entire file structure each time; by using the *+t* option you can simply backup the files that are new or changed since the last backup (incremental backup).

To perform an incremental backup:

1. Move to your home directory. Type:

   ```
   chd
   ```

   (The backup process is relative. If you are not in your home directory, it will backup files under the directory in which you are currently working.)

2. Backup your files. Type:

   ```
   backup +dtA
   ```

   (See the *backup* command in the *4400 Series Operating System Reference Manual* for an explanation of the options.)

3. When prompted, type the volume name of the first backup diskette. (The volume name can be quite large and consist of more than one word.)

4. Insert a formatted diskette as prompted by the system. (If you have not formatted your diskettes, you can type *f<Return>* instead of <Return>, and the system will format the diskette before using it. You may format subsequent diskettes in the same way.)
5. Insert subsequent diskettes until the backup set is complete. If you are backing up on a previously used set of diskettes, *backup* will only write those files newer than the date of the last backup.

NOTE

If you are a smalltalk user, and have not created your own personal changes file, be sure to back up the defalt changes file. To do so, type:

```
backup + dl /smalltalk/system/*Changes*
```

backup will use all the storage space on a diskette, breaking long files across diskettes. When you *restore* files from a backup set, they will be in the same ***relative*** position they were in when they were saved. You should maintain at least two sets of backup diskettes, using them alternately. In this way, if you damage one set, you can restore some of your files from the older backup set.

You should label your backup diskettes in sequential order. If you get them out of order, you could lose some data.

It is your responsibility to back up your files. You have the safekeeping of your work in your own hands. Should the hard disk system be damaged, a current set of backup diskettes will let you recover your files up to the point of backup. You can also exchange files with other 4400 Series systems by backing up only the files you want to exchange, then *restore* the backups onto the other system.

A Suggestion

To prevent errors and reduce typing you might find it useful to put your backup procedure into a script file and keep it in a personal command directory. As an example, let's see how a user called sams (who changed his prompt to *Sams++* in his *.shellbegin* file) would do so. (The comments in parenthesis aren't to be typed in, they just explain what the commands do.)

First, user sams creates his personal command directory */sams/bin* to contain this and any other personal utilities.

```
++login sams
Password:                    (Sam types his password here)
Sams++ crdir bin             (This will be his command directory)
```

Now he creates a empty file and gives himself execute (u+x) permission.

```
Sams++ create bin/save      (The name of the command file)
Sams++ perms bin/save u+x   (Make save executable)
Sams++ dir +dl              (Let's see what we've done)

directory ".":

bin             1 d rwxrwx  2 sams  14:27 Dec 19, 1985

directory "./bin":

save            0   rwxrw-  0 sams  14:29 Dec 19, 1985
```

User sams now has two tasks left: editing his file */sams/bin/save* to put in the commands he will use, and editing the */sams/.shellbegin* file to put the directory */sams/bin* in the search path. He can use the line editor *edit*, a Smalltalk-80 file list, or the optional editor, EMACS, to do so.

In */sams/.shellbegin* the default line for his search path is:

```
PATH=.:/bin
```

He edits this line to read:

```
PATH=".:/sams/bin:/bin"
```

Now to test if PATH is right, he types:

```
Sams++ set .shellbegin
Sams++ save
Sams++ echo $PATH
 .:/sams/bin:/bin
Sams++
```

When the prompt comes back after *save* it shows that the system recognized the command (which did nothing). In response to *echo*, the system prints the command search path it recognizes.

Finally, to make the command *save* perform his backups, in the file */sams/bin/save* he puts the two lines of his command:

```
chd /sams
backup +dltAB
```

When the file is saved, the command is ready for use. Sam put in two more options: *l* to list what it is doing and *B* to not save files that end in *.bak*. You can include these if you wish, or leave them out if you prefer.

To perform his personal backups, Sam now types *save* and follows the prompts to save his new files on today's backup disk set. The next time he backs up his files, he performs the same task using his alternate set of personal backup diskettes.

RESPONSIBILITIES OF USER *system*

When more than one person uses a 4405, one person should be responsible for the system maintenance tasks. This person should **not** use the *system* account for his or her normal work, (use a *public* level account whenever possible) but use the *system* account only when necessary to perform *system* level tasks. These accounts include the following:

BACKING UP THE SYSTEM

One of the most important tasks for *system* is backing up the files on the hard disk. If users are responsible for their own files, you should do a complete system backup whenever you change any of the system files. In addition to the system backup diskettes you received with your 4405 and archived in a safe place, you should maintain at least two more system backup disk sets, alternating between them as you perform backups. If users do not maintain their own file backups, you should also do a frequent incremental backup similar to that recommended for the users earlier in this section.

Performing a *system* Backup

A system-level backup can take over an hour, depending on the number of files that exist on the system. To save time, you should have enough formatted diskettes, labeled in sequential order, to contain the full system backup. (35 - 40 diskettes will hold the system as delivered; you'll have to add diskettes as you add files.) As mentioned, you should have at least two sets of backup diskettes. Each time, use the oldest set of diskettes for the newest backup, moving the last backup to a safe place in case anything should happen to this backup set. The backup procedure is:

1. Login as user *system.*
2. Type *chd.* If you had logged in as *system* previously, this gets you back to "/".
3. Type:

```
backup +dl
```

4. Insert the backup diskettes in sequential order and press <Return> when prompted to do so by the system.
5. Store the latest set of backup diskettes in a safe place in case they may be needed for a system rebuild.
6. Logout from the *system* account. Don't use the *system* account for normal work — use a *public* level account for that.

ADDING AND DELETING USERS

Adding and deleting users on the 4405 is a simple matter. To add a user, simply type:

```
adduser <user_name>
```

The user name should be 8 or less lower-case characters. This command creates a directory under "/" with the same name as the user name, and puts a default copy of the *.login .shellbegin*, and *.shellhistory* files in that account. If you want to set a password for the new user, use the *password* command.

Deleting a user is nearly as simple. You need to decide whether to delete the user's files and directories when you delete the user. If you want to delete the user's files, type:

```
deluser <user_name>
```

Or, if you want to retain the user's files, type:

```
deluser +x <user_name>
```

CAUTION

Never delete user "system" (the user whose user number is 0). If you do so, you will have to do a complete system rebuild as described in Section 5.

INSTALLING SOFTWARE ON THE 4405

The user *system* is also responsible for installing software on the 4405. Most software for the 4405 is distributed in *backup* format as this format makes most efficient use of the floppy diskette space. To get a catalog of files on a set of *backup* format diskettes in a file you can examine, called catFile, issue the command:

```
++ restore +C >catFile
```

Then, insert the diskettes, in sequence, as the system calls for them. When finished, you can examine the file, catFile, to see the organization of the software.

Backup format files are stored as either *relative* or *absolute* named files. *Relative* files are stored relative to the current working directory (the files start at "." on your catalog), while *absolute* files start with the root directory and give an absolute pathname for each directory.

If you want to install a set of relative files on your system, *chd* to the directory you want to be the installation site for the software, then give the command:

```
++ restore +ld
```

Insert the diskettes and press <Return> as the system prompts, and the system then restores all files and subdirectories in a subtree rooted at your current working directory.

If you want to install a set of absolute files on your system (each directory name starts with "/"), you must first be sure that each directory exists. ***Restore** will not create directories in absolute mode -- they must exist for the installation to succeed.* When you have constructed a directory structure that matches your catalog file, give the command:

```
++ restore +ld
```

The files will each be placed in the correct directory, regardless of your current working directory.

If your directory structure is not correct for a set of absolute files, the system will complain. Note the directories that don't exist, create them, then *restore* again.

Exactly how you organize your system is up to you. You can put files wherever you want, but you should keep the following points in mind.

The User's Search Path

The user's default search path covers only two directories, his current working directory (.) and the */bin* directory. If you are installing software for general use, install it (or a link to it) in */bin*. If the software is for *system* use, the directory */etc* is in the search path for *system*. If you want to add another directory to the search path,

you (or the user) must modify the environment variable PATH via *addpath* or *set.*

Keep Related Programs Together

You may wish to install a set of related programs, or programs that call each other, in a directory apart from */bin* or */etc*. To call these programs from */bin* or */etc*, you can create a link within either */bin* or */etc*. See *link* in the *4400 Series Operating System Reference Manual* for an explanation.

Search Time

If you have a very large number of files that are accessible by the *public* level users, the search time for each command might get quite long. If you have this type of situation, you might wish to start a new command directory (for example */publicbin* for less common shared commands). If you do this, the *.shellbegin* file for each user must be edited to add that directory to his or her search path. Change the line that assigns the environment variable "PATH" to include your new command directory in the order you want to search and separate the directories by colons, or, alternitavely use *addpath* for these users.

Software Requirements

Some software requires that it be placed in a specific position on the directory structure. This limits your freedom where to place the software. If you want the command which invokes the software to reside elsewhere, use *link* to allow it to be addressed by both names.

Permissions and The User Bit

For a file to be executable by another user, you should set the permissions with the *o+x* option. In the discussion on *perms* in the *4400 Series Operating System Reference Manual*, mention is made of the *user bit.* This is a device which temporarily grants the user of the file the permissions that the file's owner has. This allows you to grant temporary and limited *system* privileges to a user running a particular utility. You can set this bit on or off with the *s+* and *s-* options to *perms.*

ERROR RECOVERY AND SYSTEM REBUILDING

You should be knowledgeable about the 4405 operating system before attempting to recover lost files or rebuild the system. If you are unsure as to your abilities, make a separate backup of the system and run *diskrepair* before you attempt to recover data.

If *diskrepair* found any unreferenced files, they are left in the directory */lost+found*. You should examine these files to see if they contain the missing data.

For other error recovery and instructions how to rebuild the system, see Section 5 of this manual.

Section 5
RECOVERY AND REBUILD

INTRODUCTION

PROBLEMS

If you experience problems with your 4405, the trouble can arise from either the hardware or the software. Luckily, the 4405 is an extremely reliable machine and rarely experiences any hardware problems; users can do little to solve hardware problems beyond diagnosing them. The final discussion in this section goes through how you can use your 4405's self-test facilities to determine whether your 4405 has hardware problems.

Software, on the other hand, can be damaged by many things — from a runaway program to the problem of hyperactive fingers. Since software is more prone to damage than hardware, most discussions in this section are on recovering from software errors.

REBUILDING ALTERNATIVES

This section covers several ways of recovering a damaged system. The first discussion, *Software First Aid* covers a few of the ways you can restore a partially damaged system that still boots. In addition, this discussion shows you how to remove a forgotten *system* password.

The second discussion, *Non-Destructive System Rebuild Procedure*, takes you through more radical system repair procedures. It takes you through the use of the *DISKREPAIR* diskette and *diskrepair* utilities. It shows a few of the ways you can attempt to recover a system with valuable files (hopefully) intact. Ingenuity and experience will show you additional ways to expand these recovery methods.

The third discussion, *Complete System Rebuild Procedure*, is the most radical. It covers methods of reformatting the hard disk — a procedure that completely destroys all the data on the disk. You should attempt this rebuild procedure only if you do not have important files on your disk, or the non-destructive recovery methods cannot recover the system.

These discussions do not depend on one another — information in one discussion may be repeated in another.

YOUR BACKUP DISKETTES

Your 4405 was delivered with the software installed on the hard disk and ready to run. In addition, you received a number of diskettes containing copies of the system software, and three additional diskettes that contain utilities that can be used to rebuild a damaged system.

The three additional disks have a special format and contain a special version of the operating system that will boot from the floppy disk drive. They provide you a minimal set of tools to build up a complete hard disk version of the operating system. Since these diskettes are logically formatted differently than normal 4405 file diskettes, you must copy them with the utility *fdup*. (To do so, format three blank diskettes with *format*, then use *fdup* for each of your three special diskettes.)

Let's look at each of the special diskettes.

Your *SYSREFORMAT* Disk

This disk contains the hard disk reformatting utilities. It contains a special (bootable from floppy) version of the operating system, formatting utilities and a defect list of unusable portions of the hard disk with which it is shipped.

CAUTION

Each SYSREFORMAT diskette contains the defect list for a particular hard disk. It should **NOT** *be used to format any other hard disk — it could mark good sections of the hard disk unusable and attempt to use unusable sections of the hard disk.*

In addition to the system reformatting utilities, your ***SYSREFORMAT*** diskette contains the utilities that rebuild and restore the base operating system.

Types of Hard Disk Reformatting Utilities

CAUTION

All reformatting utilities destroy data on the hard disk. Do not reformat unless it is absolutely necessary.

There are three types of hard disk reformatting utilities on your *SYSREFORMAT* diskette. They are:

logical	A logical format simply erases data from the hard disk. It is quick, simple, and does not use the hard disk defect information. The track/sector address information on the hard disk is preserved.
physical	A physical reformat erases the entire hard disk and rewrites the track/sector information on the disk. When you use a physical reformat on the hard disk, the defect list (stored as *badblks* on the *SYSREFORMAT* diskette) locks out unusable areas on the disk. Physical reformat is the most extreme reformatting method you will use.
manual	The manual reformat utility physically reformats the hard disk, but the defect list must be manually entered. You will not normally use this utility, it is an emergency measure for use only if the defect list becomes damaged or the hard disk is replaced.

Virtual Memory And Swap Space

A section of the hard disk is reserved during formatting for *swap space.* (Swap space is that portion of the disk that is used to implement the *virtual memory* of the system — memory that is directly addressable by the processor, but not necessarily in physical memory.) Disk space used for swap space is not available to the file system. Thus, a 90 Mb disk that has 32 Mb swap space has 58 Mb of space to be used by the file system, while a 45 Mb disk with 8 Mb of swap space has 37 Mb available to the system.

Names Of Reformatting Utilities

All reformatting utilities have names of the general format: aaaaFormat-x-y Where:

aaaa — is the type of formatting utility — logical, physical, or manual.

x — is the disk size in Mb (45 or 90).

y — is the swap space in Mb (16 or 32).

CAUTION

Before using "physicalFormat", verify that the serial number on your SYSREFORMAT disk matches the serial number of your MSU (located under the front cover). "physicalFormat" uses information specific to the winchester disk assigned to it at the factory. Do not attempt to use this with any other winchester disk — it could result in an extremely unreliable or even unusable system. "physicalFormat" destroys all data on the winchester disk.

CAUTION

Using "manualFormat" requires hard disk information that is not available to the user. If you use "manualFormat" without this information, it can cause unreliable system operation.

System Rebuilding Utilities

In addition to the hard disk reformatting utilities, your *SYSREFORMAT* diskette contains a utility called *copyOS.* This utility mounts the hard disk and copies the necessary skeletal directory structore for a hard-disk based operating system and the vital system files. *CopyOS* also copies to the scripts that restore the the standard system software partitions. These scripts are not useable from the *SYSREFORMAT* system — they must be used from a hard-disk based system such as the *SYSINSTALL* system.

Your *SYSINSTALL* Disk

This disk is a bootable floppy that installs a hard-disk-based operating system that runs in the structure created by the *SYSREFORMAT* system. The structure created by formatting the disk or by using *CopyOS* will support several special utilities (scripts) that restore partitions of, or the whole, operating system. These utilities are:

- *fileRestore* — a master utility to recover the six system partitions:
    - restoreOS
    - restoreK
    - restoreC
    - restoreSTI
    - restoreSTS
    - restoreSTD
- *restoreOS* — a utility to recover the operating system files from the standard system diskettes supplied with your system.
- *restoreK* — a utility to recover the operating system kernel from the standard system diskettes supplied with your system.
- *restoreC* — a utility to recover the 'C' environment files from your standard system diskettes.
- *restoreSTI* — a utility to recover the Smalltalk-80 interpreter from your standard system diskettes.
- *restoreSTS* — a utility to recover the Smalltalk-80 standard image, sources, and other system files from your standard system diskettes. RestoreSTS also overwrites the standard changes file with an empty (length 0) file and could potentially destroy any code stored in the default changes file.
- *restoreSTD* — a utility to recover the Smalltalk-80 demo system from your standard system diskettes.

Details of how to use these utilities are given in the *Non-destructive System Rebuild Procedure.*

Your *DISKREPAIR* Disk

This disk is a bootable minimum system that contains two useful utilities:

- *diskrepair* — a copy of the system *diskrepair* utility that can run from a floppy-based system. This utility can sometimes repair a system that will not boot. The *Non-destructive System Rebuild Procedure* tells how to use *diskrepair.*
- *rmpass* — this utility removes the *system* password, should it be forgotten. Details on how to use this utility are found in this section under *Software First Aid.*

Your Standard System Diskettes

Your standard system diskettes contain archival copies of the system software as delivered from the factory. You should copy these diskettes using *fdup* and store the originals in a safe place. Use the copies to rebuild your system should that become necessary.

Your standard system diskettes are partitioned into six groups of files:

1. The operating system files
2. The operating system kernel
3. The 'C' compiler and environment
4. The Smalltalk-80 interpreter
5. The Smalltalk-80 standard image and system files
6. The Smalltalk-80 demoImage and a number of demonstration files and utilities.

These diskettes are in absolute, rather than relative backup format. They cannot be restored to other than their correct places in the system. If you must restore files from these diskettes, the directories which will contain the restored files must be present on the hard disk; *restore* will not create directories in absolute format. If you get errors during an attempt to restore files from these diskettes, follow the procedure under "Non-destructive System Rebuild."

SOFTWARE FIRST AID

PREVENTIVE MEDICINE

The most effective insurance against disaster is a carefully followed plan of backing up your files. Without recent back ups, recovery of lost files is nearly impossible. With a systematic and regular backup procedure, you can minimize data loss from even the most severe system crash. Read the discussion on *Backing Up the System* in section 4 of this manual and follow either the schedule presented there, or a more rigorous backup schedule.

AUTOMATIC SYSTEM REPAIRS

Each time the 4405 boots, either from power-on or from a reset, it checks to see that it was shut down gracefully. If there is a possibility that the system was damaged or shut down in any other manner (someone pressed the reset button, or simply turned the power off), it runs the utility *diskrepair* to diagnose and repair potential or actual problems.

If you suspect that something has happened during normal operation that could damage the system structure, log in as user *system* and run *diskrepair* (full details are given later in this section). If you use the *+v* option, it will report any potential problems it finds. You can also run the disk diagnostic utilities without attempting to repair the disk structure by using the *n*, *b*, and *f* options. In some cases, this may help you to recover damaged files without doing a complete system rebuild. If your system cannot be booted, you can try booting the *DISKREPAIR* disk and run *diskrepair* from it. If *diskrepair* cannot repair your system, you will need to do a complete system rebuild.

REMOVING A FORGOTTEN *system* PASSWORD

If you lose or forget the *system* password, you can recover use of the system by using the *rmpass* utility on the DISKREPAIR disk. This utility removes the password from the user *system.*

To use the *rmpass* utility, boot the *DISKREPAIR* diskette (see the booting instructions under *Non-Destructive System Rebuild*) and type:

```
rmpass
```

This utility is an executable script that mounts the hard disk, edits the password file, then unmounts the hard disk. If this script runs into trouble (for example if the hard disk does not mount) you may need to run *diskrepair* as in the *Non-Destructive System Rebuild Procedure.*

RESTORING A USER'S FILES

If the system-wide structure seems intact, and damage is confined to one or two user accounts, you should restore only that portion of the system from the user's incremental backups. If you have followed the recommended procedure in section 4, you (or the user) can do so by following

this procedure:

1. Login under the user's name.
2. Have the latest set of incremental backup disks available.
3. Be sure you are in the user's home directory.
4. Give the command: restore +bdln
5. Insert the incremental backup disks in order as the system prompts for them.

RESTORING FILES ON A BOOTABLE SYSTEM

If the system is damaged but still bootable (some files deleted, but you are still running), you can often restore it by:

1. Login as user *system.*
2. Have the most recent set of system backup disks at hand.
3. Give the command: restore +dl
4. Insert the disks as the program prompts you for them.

If the lost files were not due to a known operator error, you should run *diskrepair* before restoring the system files.

WHEN THE SYSTEM WILL NOT BOOT

When the system boots, it tries to locate a file in the root of the winchester disk file structure called "system_4405.boot." If this file is lost or renamed, when you turn on the 4405 it displays the message:

```
Cannot Locate Boot File
Enter <Return> to continue:
```

When you press the *return* key, the system finds and displays a menu of files ending in ".boot" that it finds on the hard disk. Press the function key associated with the ".boot" file you wish to try. If you cannot boot the system from this menu, you should try the *Non-destructive System Rebuild Procedure.*

RECOVERING AN UNBOOTABLE SYSTEM

If your system will not boot, you will see an error message indicating either a hardware or software error. Hardware errors will be such things as "Device not ready." Software errors will be of the type "Cannot locate boot file" or "Wrong file type." An "I/O error" can indicate either hardware problems, or file system structural errors (run *diskrepair*).

If you have a hardware error, first check the following:

- Does the disk drive indicate activity? If not, check that the power cord is connected, that the power switch is on, and that the fuse in the back panel of the MSU is intact.
- If the MSU is running, check that the SCSI terminator is installed and secured and that the SCSI cable is securely seated.
- If you receive a "Device not ready" error, try turning the power to the MSU off, then on. Try booting again by pressing the Display/CPU reset button.

If you have software errors, or suspect that the errors could be caused by software, you should perform the following *Non-destructive System Rebuild Procedure.*

NON-DESTRUCTIVE SYSTEM REBUILD PROCEDURE

OVERVIEW

If you have important files on a non-bootable system, it is sometimes possible to rebuild all or portions of the system and recover the lost data. Before attempting this task, you should verify that the system hardware is running correctly, that the SCSI cable and terminator are completely seated. If you doubt the system's hardware integrity, run it in Continuous Self-Test overnight, as described in the final discussion in this section.

Figure 5-1 shows the 4 steps involved in the Non-destructive System Rebuild Procedure. They are:

1. Boot the *DISKREPAIR* diskette, a floppy-based system, and repair the directory structure of the hard disk if necessary. (Your system may be bootable after this step).
2. Boot the *SYSREFORMAT* diskette, another floppy-based System. Run the utility *copyOS*. This constructs the proper operating system file structure and copies the necessary files to the hard disk so that the *SYSINSTALL* system can run. This step also copies the utilities with which you can install the standard system software or any of its partitions.
3. Boot the *SYSINSTALL* diskette. This hard disk system can now run the utilities that restore the standard system software partitions.
4. Boot the repaired system from the hard disk. The operating system files and any other files you needed to restore are now in place, along with undamaged files from the previous operating system.

   The utility *copyOS* attempted to copy the old password file to the file */etc/log/password.bak*. It then replaced the password file with the default password file that contained only users *system* and *public* with no passwords. If this copy was sucessful, you can login and rename */etc/log/password.bak* to */etc/log/password*. If the file */etc/log/password.bak* was damaged, you can replace it from your incremental backup disks or rebuild it with the *adduser* utility.

   NOTE

   File ownership is determined by user number, not by name. If the /etc/log/password file is not restored, only system and public will own files. The others will only show ownership of the form : "<<nn>>" where nn is the user number who owns the file. If you add a user who is assigned that number, that user will then own the file in question.

The remainder of this discussion deals with the details of each of these steps.

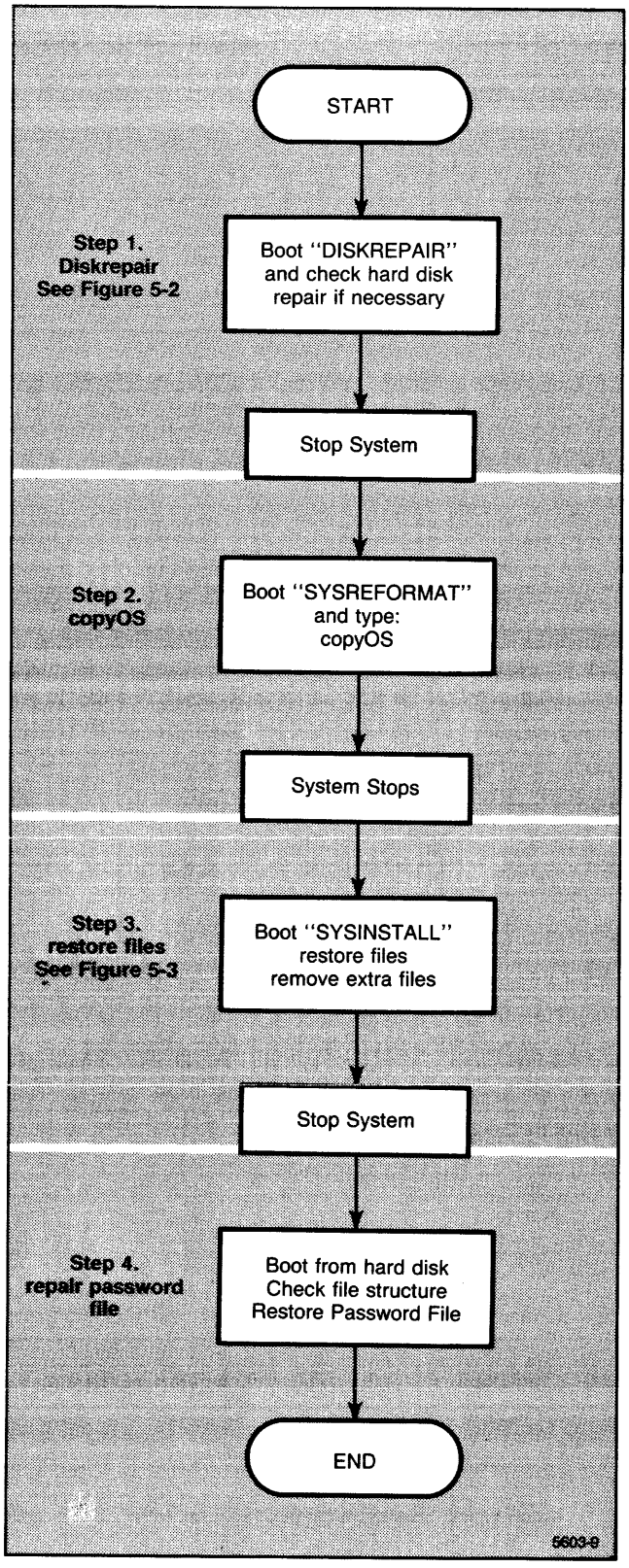

Figure 5-1. Non-Destructive System Rebuild Procedure.

STEP 1 *DISKREPAIR*

The first step in rebuilding a non-bootable system is to determine how badly it is damaged, and to repair the directory structure if it has become corrupted. Your *DISKREPAIR* diskette is a bootable diskette that contains utilities that will allow you to inspect the hard disk system and a copy of the *diskrepair* utility to rebuild a damaged directory structure. Figure 5-2 shows some of the possible choices available when you have successfully booted a *DISKREPAIR* system.

The *DISKREPAIR* system is a *floppy-based* system. It is based on the floppy, runs a more primitive shell, and looks for its utilities on the floppy. Due to space limitations, it does not contain all the functionality of the full system. When you boot the *DISKREPAIR* system, you are logged in as user *system.*

To boot your *DISKREPAIR* system:

A — Boot the *DISKREPAIR* Diskette

1. Insert the *DISKREPAIR* diskette into the floppy disk drive and power on the system.
2. Invoke the *Interactive Boot/Self-Test Menu.* You can either:
   A. Press and release the Reset Button twice. Hold the Reset Button down briefly each time. The Reset Button is located on the rear panel of the Display/CPU unit.
   B. Hold the left *Shift* key down and press the *Break* key for several seconds after the screen clears and the machine type (*Tektronix 4405*) appears.
3. When the *Interactive Boot/Self Test Menu* appears, press function key f1 (Interactive Menu).
4. When the *Interactive Boot Menu* appears, press function key f2 (Boot system_4405.boot from Floppy).
5. Wait for the system to successfully boot from the floppy disk. (This takes about a minute.)

If the *DISKREPAIR* diskette will not boot, it indicates that either the *DISKREPAIR* diskette is damaged, or that you have hardware problems with your system.

To isolate the problem further, attempt to boot your *SYSREFORMAT* diskette. If this diskette will not boot, you have hardware problems. Recheck your cables and terminator, and attempt to boot again. If the problem persists, contact your local Tektronix service representative.

If your *SYSREFORMAT* diskette boots, this indicates that your *DISKREPAIR* diskette is damaged. Contact your local Tektronix sales representative for assistance in replacing it.

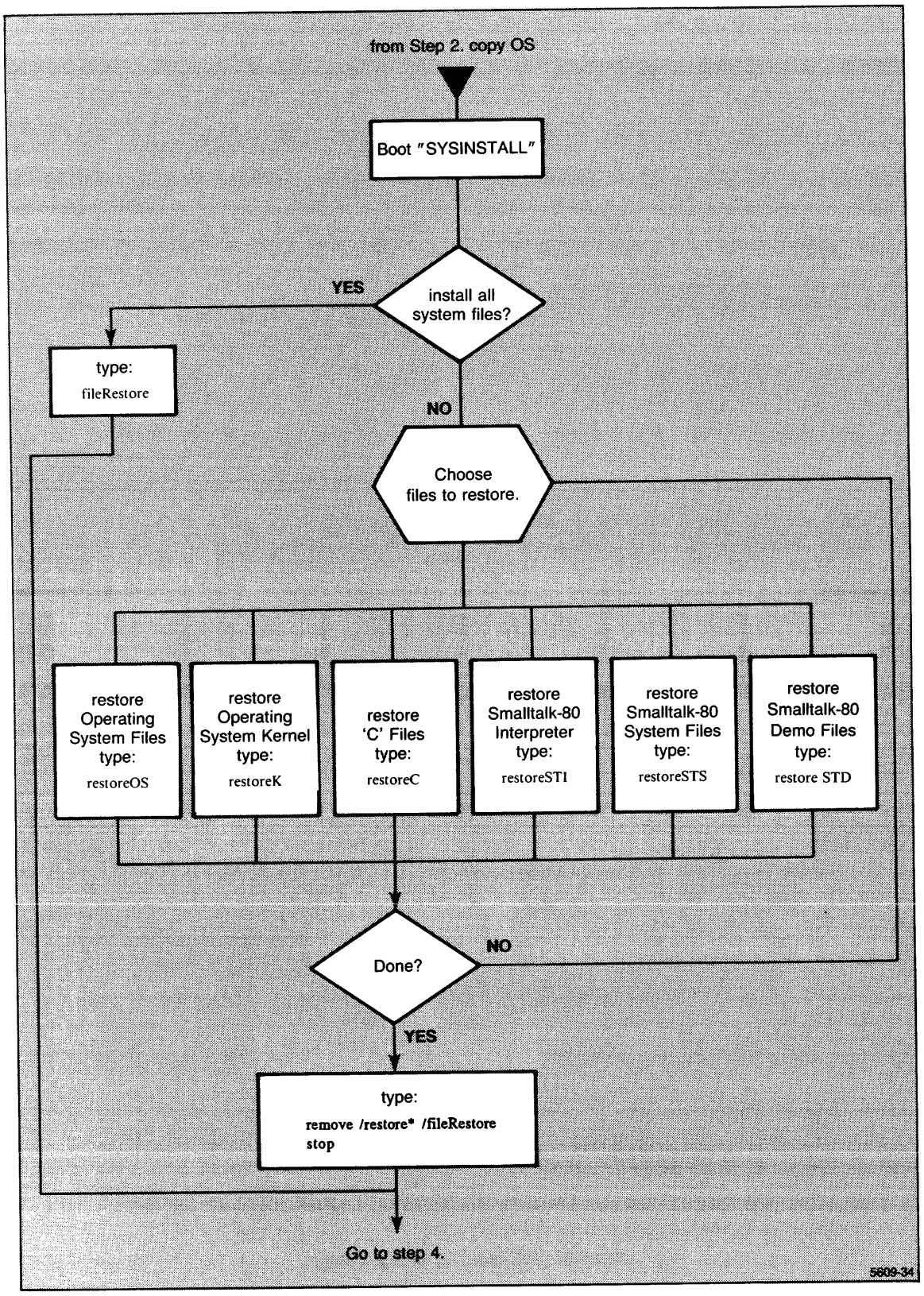

Figure 5-2. Step 1. Using *DISKREPAIR*.

B — Mount the Hard Disk

When the *DISKREPAIR* system returns with the ++ prompt, type:

```
mount /dev/disk /disk
```

If the hard disk mounts successfully, this indicates that the directory structure on the hard disk is intact. You can inspect files on the hard disk without attempting to repair it. Go to the step marked *D — Inspect Your Hard Disk Files.*

C — Run *diskrepair*

If the hard disk does not mount, the directory structure on the hard disk has been damaged. To repair it, type:

```
diskrepair /dev/disk +v
```

The +v option to *diskrepair* causes the program to inform you of its progress (see the *4400 Series Operating System Reference Manual* for details).

If *diskrepair* terminates normally (it returns you to the ++ prompt with no warning messages), go on to the next step. If *diskrepair* exits with warning messages, or shuts the system down, then you should boot the system if necessary and run *diskrepair* again. If, after several tries, *diskrepair* cannot repair the hard disk, your data is most likely lost. You must do a complete system rebuild.

D — Inspect Your Hard Disk Files

When you have successfully mounted the hard disk, you are running as *system* and have full access to any undamaged files and utilities that reside on the hard disk as well as the utilities on the *DISKREPAIR* diskette. You can use some of the hard disk utilities (but not those dependent on absolute path names such as *backup*) to inspect the hard disk directories and files. You need not inspect the hard disk files if you already know what file sets you will be restoring.

When you are running from your *DISKREPAIR* system, the root of the directory tree is on the floppy drive, while the root of the hard disk is at /disk. To access hard disk files, you must type the full path name (starting from /disk) of the utility you want to use or file you want to reference. For example, if you want to use a utility found in /bin on the standard system, you must type /disk/bin/<utility>. Alternatively, you could change your working directory to /disk/bin (to put the utility in your search path), but must still remember to type the full name of the files you reference.

The following examples show two ways to move files from one user on the hard disk to another.

```
/disk/bin/copy /disk/firstuser/filea /disk/seconduser/filea
```

```
chd /disk/bin
copy /disk/firstuser/filea /disk/seconduser/filea
```

You can sometimes diagnose the hard disk booting problem by inspecting the hard disk file structure. Example 5-1 shows the files and permissions that must be present for a disk system to boot. (Use *dir +adl* from / to find them).

```
Directory "/":
.badblocks          - - - -
act                 d rwxr-x
bin                 d rwxr-x
dev                 d rwxr-x
etc                 d rwxr-x
gen                 d rwxrwx
lost+found          d rwxr-x
system_4405.boot      rwx--
tmp                   rwxrwx

Directory "/disk/act":
utmp                  rw-rw-

Directory "/disk/bin":
remove                rwxr-x
script                rwxr-x
shell                 rwxr-x
stop                  rwxr-x

Directory "/disk/dev":
console             c rw-rw-
disk                b rw-rw-
diskc               c rw-rw-
floppy              b rw-rw-
floppyc             c rw-rw-
null                c rw-rw-
pmem                c rw-rw-
smem                c rw-rw-
swap                b rw-rw-
tty00               c rw-rw-

Directory "/disk/etc":
.init.control         rw--
init                  rwx--
log                 d rwxrwx
ttylist               rw-r-

Directory "/disk/etc/log":
password              rw-r-

Directory "/disk/gen":
errors              d rwxrwx
   system                 rwxrwx
Directory "/disk/gen/errors":
system                rw-r-


Directory "/disk/gen/system":
chkpass               rwx--

Directory "/disk/lost+found":

Directory "/disk/tmp":

Directory "/disk/public"
.shellbegin           rw-r-
.shellhistory         rw-r-
```

Example 5-1. Minimum Bootable System.

NOTE

In the directory **/disk/dev**, *the files* **/disk/dev/console** *and* **/disk/dev/tty00** *are linked together, as are /disk/dev/disk and /disk/dev/swap. In addition, the file* **/disk/public/.shellhistory** *can prevent booting if it is corrupted. (You can remove it and create a new file of 0 length if you suspect this.)*

E — Unmount the Hard Disk and Stop the System

To unmount the hard disk and stop the system, type:

```
unmount /dev/disk
stop
```

If you current working directory is on */disk*, the system will respond with the message "`device busy.`" Use *chd* to get back to "/", then stop.

STEP 2. COPY THE OPERATING SYSTEM FILE STRUCTURE

Boot your "SYSREFORMAT" diskette with the same procedure you used with your "DISKREPAIR" diskette.

When you get the ++ system prompt, type:

```
copyOS
```

The executable script file *copyOS* creates the necessary skeletal directory structure needed on the hard disk, then copies vital system files to the hard disk. These files can later be used with the "SYSINSTALL" diskette to create a hybrid system that frees the floppy drive and allows you to restore the bootable hard disk system.

When *copyOS* is finished, the script shuts the system down and prompts you to perform the next step.

STEP 3. RESTORE THE SYSTEM FILES.

Following the same boot procedure, boot your "SYSINSTALL" diskette.

The *SYSINSTALL* diskette loads a *hard disk based* operating system kernel into the system's memory. The actual scripts for restoring files came (via *copyOS*) from the *SYSREFORMAT* diskette — but could not run from the floppy based operating system, as it requires a floppy disk in the drive. A *system_4405.boot* file on the hard disk would be an image of the floppy based system on the *SYSREFORMAT* diskette, and would require a floppy in the drive at all times in order to run.

The *SYSINSTALL* diskette contains a backup/restore program in case the hard disk version is inoperable. The *restoreOS* script mounts the *SYSINSTALL* diskette and copies the backup/restore utility to the hard disk, then runs it in restore mode. If you are trying to partially restore the system and the partition restoring scripts fail, you should manually move this utility to the hard disk. To do so, you must be logged in as *system*, have the *SYSINSTALL* diskette in the floppy drive, and type:

```
++ mount /dev/floppy /floppy
++ copy /floppy/gen/system/backup /gen/system +lop
++ unmount /dev/floppy
++
```

The operating system that this boot procedure brings up allows the use of /dev/floppy — thus you can restore the standard files to bring the system up to full function. Figure 5-3 shows the choices available when restoring files.

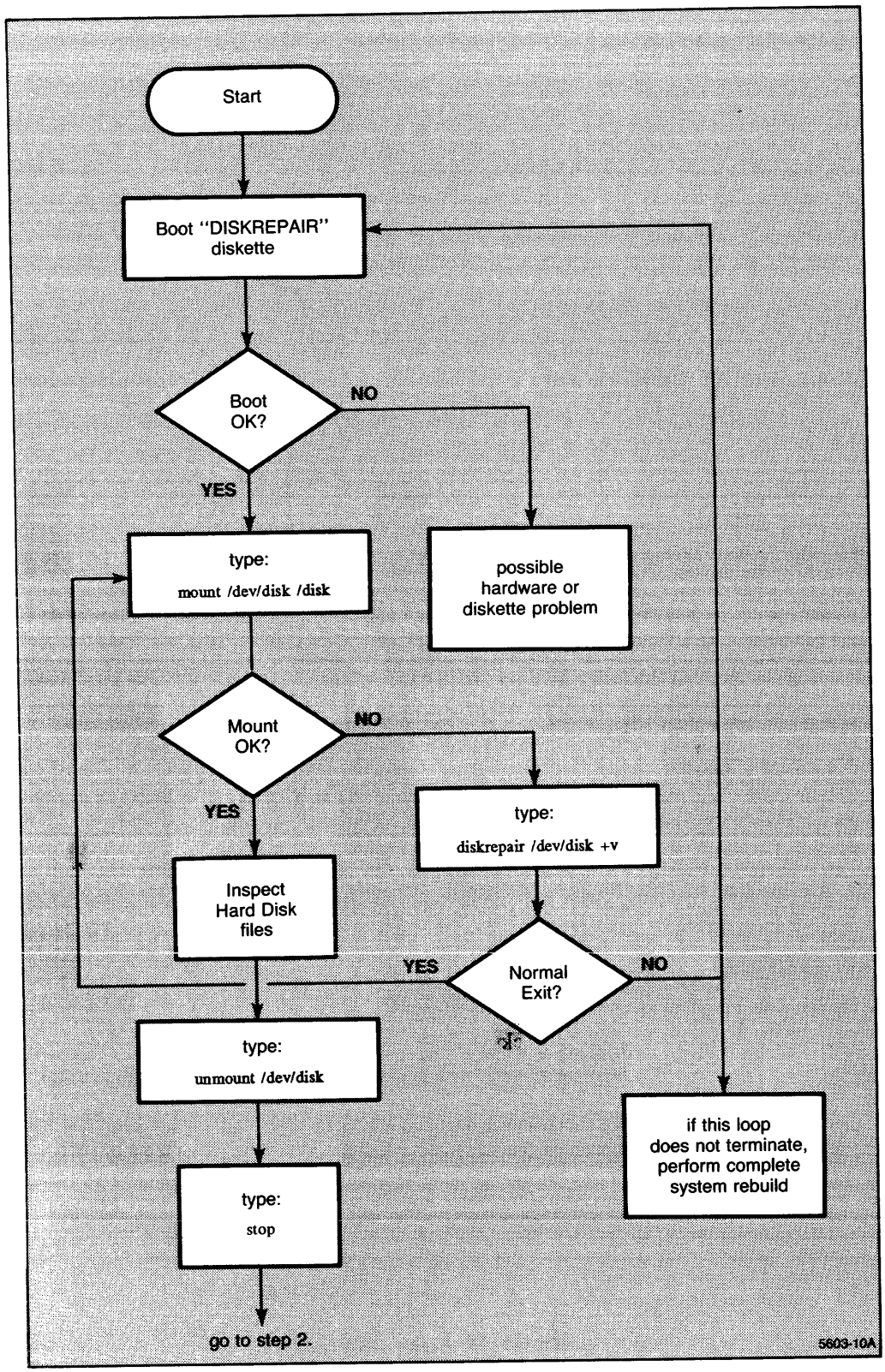

Figure 5-3. Step 3. Restore Files.

NOTE

To use the "SYSINSTALL" scripts, you should be logged in as user "system". If you want to try these scripts from a damaged, but bootable hard disk system, login as "system" and work from the root directory.

When you get the system prompt, you have several choices:

1. *If you want to restore the complete set of system files, type:*

   ```
   fileRestore
   ```

   The script, fileRestore, prompts you to insert each diskette of the six sets of standard system files. Follow the prompts, and when the script is complete your system will be bootable from the hard disk.

2. If you want to restore only the operating system files, type:

   ```
   restoreOS
   ```

   This script prompts you to insert the diskettes from the set of standard operating system utilities diskettes. When this script completes, you can either copy another group of files or stop the system.

3. If you want to restore only the operating system kernel, type:

   ```
   restoreK
   ```

   This script prompts you to insert the standard operating system kernel diskette. When this script completes, you can either copy another group of files or stop the system.

4. If you want to restore only the 'C' environment and graphics library, type:

   ```
   restoreC
   ```

   This script prompts you to insert the diskettes from the set of standard 'C' environment diskettes. When this script completes, you can either copy another group of files or stop the system.

5. If you want to restore only the Smalltalk-80 interpreter, type:

   ```
   restoreSTI
   ```

   This script prompts you to insert the standard Smalltalk-80 interpreter diskette. When this script completes, you can either copy another group of files or stop the system.

6. If you want to restore only the Smalltalk-80 standard system files, type:

   ```
   restoreSTS
   ```

   This script prompts you to insert the diskettes from the set of standard Smalltalk-80 system diskettes. When this script completes, you can either copy another group of files or stop the system.

NOTE

This script overwrites the default standard changes file. See the discussion in Section 4, Software Maintenance about backing up the default changes file.

7. If you want to restore only the Smalltalk-80 demo files, type:

```
restoreSTD
```

This script prompts you to insert the diskettes from the set of Smalltalk-80 demo diskettes. When this script completes, you can either copy another group of files or stop the system.

The partial file restore scripts leave copies of the scripts on the hard disk. To remove them, type:

```
++ remove /restore* /fileRestore
++ stop
```

(If you use the *fileRestore* script, it removes these files and stops the system for you.)

STEP 4. RESTORE THE PASSWORD FILE

Your hard disk should now have a bootable system on it. Boot the system from the hard disk by pressing the reset switch once. When the prompt appears, you are in a standard system logged in as user "public".

Login as "system" (you won't need a password) and inspect the files remaining from the last system. If you use the +l option to the *dir* command, you will notice that some retained files may not have an owner name associated with them. (The operating system obtains the owner name for files from the password file, and this file has been replaced.)

The *copyOS* and *restoreOS* utilities copy the existing password file (assuming it has not been lost or damaged) to the name *password.bak*, then copy in the distribution password file that contains only *system* and *public*. type:

```
sys++ chd /etc/log
sys++ dir
password        password.bak
sys++
list password.bak
(... contents of old password file)
sys++
```

Inspect the contents of the backup password file. If everything is fine, type:

```
sys++ rename password.bak password
sys++
```

You should now have your original password file on the system.

If the *password.bak* file is missing or damaged, you should have a copy of the current password file in your incremental system backup diskettes. To find the diskette containing this file, type:

```
restore +C
```

The system will display the files on each of the diskettes as you insert them. The password file will appear as:

```
"Directory /etc/log"
password ...(permissions, owner, etc)
```

When you locate the password file, you can abort from the *restore* utility by typing a <Ctrl-C>. To restore it, insert the disk and type:

```
restore +l /etc/log/password
```

If this diskette is not the first in your backup set, you will get a message from *restore* indicating that this is the wrong volume of the backup set, asking if you want to proceed. Answer "Y" and the password file will be restored.

CAUTION

Be sure you are restoring the file from /etc/log/password and not the utility from /bin/password. If the password file is incorrect, the system will not boot.

Alternatively, if you know the order in which the users appeared in the original password file, you can use the *adduser* utility to rebuild the file by adding users in the same order in which they were assigned. The utility *adduser* checks for files under that user name and complains (but creates the user) if it finds one.

CAUTION

File ownership is determined by user number. If you create users in a different order than they were originally created, you will have a case of massive file ownership confusion to sort out.

A final alternative is to directly edit the password file (examine a working password file before attempting this) and manually enter the fields. Either of these options requires that you know the contents of the original password file, and that users' accounts remain unprotected by passwords until either you set them while logged in as user *system*, or the user sets his or her own password again.

COMPLETE SYSTEM REBUILD PROCEDURE

OVERVIEW

Rebuilding a complex software system from scratch is not a trivial task. However, the procedure has been automated considerably. You first boot a simplified system from a flexible disk — a system that performs two tasks, it formats the winchester disk, then installs a minimum system (without the O/S kernel) on the hard disk. Next, you boot another simplified system that is a hard

disk resident system. It moves the backup/restore utility to the hard disk, then restores the entire system from the system backups. The final rebuild step is to restore any user's files that were made after the last full system backup. These steps are treated in detail for the remainder of this discussion.

STEP 1 — FORMAT THE WINCHESTER WITH *SYSREFORMAT*

CAUTION

Verify that the serial number on your SYSREFORMAT diskette matches the serial number of the MSU (located under the front cover). Each SYSREFORMAT disk contains badblock formatting information for one and only one hard disk. Using the wrong SYSREFORMAT disk can create an unreliable or unusable system. To verify that the serial number is correct:

1. *Be sure that the power is off to the MSU.*
2. *Grasp the front bezel of the MSU in both hands and pull it* ***STRAIGHT*** *forward. (This exposes the shaft of the power switch — it's very fragile and easy to break off.)*
3. *Once clear of the switch shaft, look for the serial number. Try not to remove the connector to the LED. (If you put it back wrong, it won't ruin it, it just won't glow.)*
4. *Compare the serial number on the SYSREFORMAT diskette to that of the MSU. If they don't match, don't use "physicalFormat."*

To format the hard disk, you must boot the minimal system contained on the disk marked *SYSREFORMAT*. This disk contains enough of a system to format the winchester in one of three ways:

- Logical format — a procedure that erases all the data and file structure information on the disk, but does not physically reformat the disk.
- Physical format with defect list — a procedure that physically erases and reformats the disk. Formatting information is taken from a file on the floppy disk that is specific to the winchester it was made for.
- Physical format from keyboard — a physical format that requires the user to manually enter the formatting information. This information is marked on the body of the winchester disk and is not normally available to the user. This procedure is used by a service person after replacing a winchester disk.

When you must rebuild your system, first try a logical format, then a physical format from file. If the physical format does not work, the winchester disk may be defective. If this should happen, contact your local Tektronix Field Service office for servicing.

A — Boot the *SYSREFORMAT* Diskette

1. Insert the *SYSREFORMAT* diskette into the floppy disk drive and power on the system.
2. Invoke the *Interactive Boot/Self-Test Menu.* You can either:
   A. Press and release the Reset Button twice. Hold the Reset Button down briefly each time. The Reset Button is located on the rear panel of the Display/CPU unit.
   B. Hold the left *Shift* key down and press and hold the *Break* key for several seconds after the screen clears and the machine type (*Tektronix 4405*) appears.
3. When the *Interactive Boot/Self Test Menu* appears, press function key f1 (Interactive Menu).
4. When the *Interactive Boot Menu* appears, press function key f2 (Boot system_4405.boot from Floppy).
5. Wait for the system to successfully boot from the floppy disk. (This takes about a minute.)

B — Format the Hard Disk

Logical Format. If this is your first attempt, logically format the hard disk. To do so, deterimine the disk size and swap space you are using. For example, if you are formatting a 45 Mb hard disk with a 8 Mb swap space, type:

```
++ logicalFormat-45-8
++
```

Physical Format. If you have already attempted an unsuccessful logical format, physically format the disk.

CAUTION

Be sure that your SYSREFORMAT diskette is the correct one for your hard disk drive. Remove the diskette and check the serial number against the serial number on the MSU. If they do not match, do ***NOT*** *proceed.*

Deterimine the size of the disk and swap space you are using. To format a 45 Mb disk with 8 Mb of swap space, type:

```
++ physicalFormat-45-8
++
```

The physical format will take five to ten minutes to complete. Do not disturb the system while it is formatting.

Remove the *SYSREFORMAT* disk when the following message appears.

```
"... System shutdown complete ..."
```

STEP 2 — RESTORE THE SYSTEM WITH THE *SYSINSTALL* DISK

A — Boot the *SYSINSTALL* Disk

The *SYSINSTALL* disk contains a minimal bootable operating system that can restore your system software. To boot this disk, simply:

1. Insert the *SYSINSTALL* diskette into the floppy disk drive and power on the system.
2. Press and release the Reset Button twice. Hold the Reset Button down briefly each time. The Reset Button is located on the rear panel of the Display/CPU unit.
3. When the *Interactive Boot/Self Test Menu* appears, press function key f1 (Interactive Menu).
4. When the *Interactive Boot Menu* appears, press function key f2 (Boot system_4405.boot from Floppy).
5. Wait for the system to successfully boot from the floppy. (This takes about a minute.)

B — Restore Files from Your System Backups

You now have a system that was booted from the floppy disk and is running from the hard disk. This system is capable of reading the system software, either from a complete system backup you made (as advised in Section 4) or from the set of backup diskettes you received with your system. To restore your system, type:

```
++ fileRestore
```

Now, relax and wait a few minutes. During this time, the *backup* program is moved to the hard disk, then *fileRestore* invokes *restore* to install the first partition of the system.

When the system prompts you with the message:

```
"Insert next volume -- Hit C/R to continue:"
```

Remove the *SYSINSTALL* disk and insert the first disk of your standard operating system diskettes. Press the *Return* key, and if everything is correct, the system will restore the files from that diskette.

As each diskette is restored to the system, the system will
prompt you to insert the next disk with the message:

```
"Insert next volume -- Hit C/R to continue:"
```

After each group of files has been restored, the system will prompt you to insert the next group of diskettes. When the system restoration is complete, the system will return a message to that effect.

C — Stop the System and Reboot

The *fileRestore* script automatically shuts down the system when it has completed. When the message:

```
"... System shutdown complete ..."
```

appears, you can reboot by simply pressing the Reset Button. On the first reboot, the system will go through a normal boot-up sequence, run *diskrepair* (to verify the file structure), do a CRC

check, then shut down the system again. Reboot again, and the system will run *diskrepair* and do a CRC check again, then log you in as user *public*.

STEP 3 — RESTORE USER'S FILES

The last task of the system rebuild is to restore any user files that were not copied during the last system backup. You can do this, or, if other users keep their own backups, they can restore their own files.

4405 SELFTEST

OVERVIEW

Selftest is used primarily during manufacturing and servicing of the 4405. It contains many functions that are not of general interest to the user. A complete discussion of selftest may be found in the *4405 AIS Field Service Manual.*

If you have any question whether a problem is caused by a fault in the 4405's hardware, you can run selftest to either confirm or deny that suspicion. Each time you turn on the 4405 (or press the Reset button) it executes a *power-up* selftest that is invisible to the user unless it detects an error. If you saw no error messages the probability of a hardware problem is low.

RUNNING SELF TEST

If you still suspect that you may have a hardware problem, you can invoke a more extended selftest by the following procedure:

1. If the system is on, stop it gracefully by typing stop and waiting until the system shut-down message appears. The shut-down message is:

   ```
   "...System Shutdown complete... "
   ```

   (If the system is already shut off, turn on the power to both the Display/CPU unit and the MSU and perform step 2 before the system begins to boot.)

2. Locate the Reset button (on the left rear of the Display/CPU unit as you face it) and press it twice. Hold it in briefly for each press and wait a moment between pushes.

3. The LED in the *Caps Lock* key will flash off and on until the keyboard test is done.

4. The 4405 initializes its display. Errors found at this point will be shown on the display. Upon completion of the Display initialization routine, the word "Tektronix " is printed on the screen.

5. The 4405 initializes the keyboard. If the keyboard is good, the LED will be left off, otherwise it will turn on. Errors are shown on the display. Upon completion, it prints

"4405" on the display.

6. The 4405 now prints the "Interactive Boot/Self Test Menu." This menu shows the function key selections available at this point. These are:
   - f1 — Interactive menu (discussed under *System Rebuild Procedure*)
   - f2 — Adjustment procedures (includes sound generator)
   - f3 — Adjustment menu (Disk load)
   - f9 — Continuous selftest
   - f10 — Enable/Disable cache
   - f11 — Continue selftest
   - f12 — Exit selftest

   If you make no selection, in approximately 20 seconds the 4405 defaults to the "Continue Self Test" option.

Key f1

Key f1 takes you into the interactive menu. This is the boot procedure discussed under *System Rebuild Procedure.*

Key f2

Key f2 invokes a set of adjustment procedures used during manufacturing and repair. These are of little use to most users, with the exception of the Sound Generation Menu. (This test procedure is menu-driven, so feel free to experiment.)

Key f3

Key f3 accesses the hard disk and loads the full selftest into RAM, then executes from there.

Key f9

Key f9 invokes continuous selftest. In this mode, the 4405 repeatedly goes through its selftest, halting only if it encounters an error.

Key f10

Key f10 enables and disables the use of the cache while executing selftest.

Key f11

If you select key f11, or do nothing, the 4405 goes into "Continue Self Test". The display flashes as the display memory is tested, then the 4405 begins printing messages on the screen as it tests its functions. When the selftest is complete, the 4405 will again display the "Interactive Boot/Self Test Menu."

Key f12

If you press key f12, the 4405 exits from selftest. It goes through its power-up initialization and attempts to boot from the hard disk.

FINDING INTERMITTENT ERRORS

Perhaps the most annoying problem in any mechanism, from a computer to a fishing pole, is an *intermittent* problem — a problem that disappears whenever a service person comes within a mile of the afflicted machine. If you suspect that you have one of these pernicious bugs living in your 4405 hardware, put the machine into continuous selftest and leave it overnight. The machine will run through its repertoire of tests until it encounters an error, when it will halt. If an error occurs, make a record of it and contact your local Tektronix service center for repairs.

Invoking Continuous selftest

To put your 4405 in continuous selftest:

1. Stop the system
2. Press the reset button twice, holding it briefly each time
3. When the "Interactive Boot/Self Test Menu" appears, press function key f10. The 4405 will then cycle through its ROM-based selftest.
4. To exit from continuous selftest, you must wait until the menu appears once again, then select an option from the menu before the selftest continues.

Appendix A
UNPACKING AND INSTALLATION

INSTALLATION

Before unpacking your 4405, you should select a site that will be suitable for comfortable operation of the 4405. You will also need a carton opener (a small stout-bladed knife will do) and a small, flat-bladed screwdriver.

Installing your 4405 consists of these basic steps:

1. Select an installation site.
2. Unpack the Display/CPU unit and the MSU (mass storage unit). Keep all your packing materials in case you later need to transport the 4405.
3. Assemble the mouse.
4. Connect the keyboard and mouse to the Display/CPU; connect the SCSI cable to both units.
5. Connect a power cord to both units.
6. Connect both units to a suitable power source.
7. Turn to Section 2 *The First Time* and go through the transcript there to verify that the 4405 is working properly.

SELECTING A SITE

You should choose a site for your 4405 that meets the following requirements:

- **Enough room for ventilation and cable routing.** You should allow at least three inches (75 mm) behind both the MSU and the Display/CPU for cable routing. You must also allow at least two inches between the ventilation openings and anything that might obstruct air flow.

CAUTION

Do not block the airflow or cover the 4405's air vents in any way. This could cause overheating and result in circuit damage.

- **Properly grounded power source.** Be sure that a properly grounded source of power is available for your 4405.
- **A stable environment.** Ambient temperature while the 4405 is operating should stay within 50 to 104 degrees Fahrenheit (+10 to +40 degrees Centigrade). Relative humidity should stay between 10 and 75%. The area should not be subject to rapid temperature fluctuations.

UNPACKING

NOTE

If your 4405 has been unpacked, move ahead to the appropriate step.

Before unpacking check both shipping cartons for signs of damage. Report any damage to the carrier and your Tektronix sales representative immediately.

Retain all packing materials in case the 4405 must be transported in the future.

UNPACK THE MSU

1. Open the MSU carton from the top using the carton opener.
2. Remove the box containing the software and instruction manuals.
3. Cautiously remove the MSU unit and place it where it will be installed.
4. Remove the shipping card from the floppy drive.

UNPACK THE DISPLAY/CPU

1. Open the carton from the top using the carton opener.
2. Remove the keyboard and set it aside.
3. Remove the top packing pad.
4. Remove the package containing the accessories and set them aside.
5. Lift the Display/CPU unit from the packing and set it in the installation site.
6. Remove the keyboard, mouse, and SCSI cable from the accessory packages and set them next to the 4405.

CHECK THE ACCESSORIES

In addition to the keyboard, mouse, SCSI cable, and the MSU and Display/CPU units, you should have the following accessories:

- *4405 User's Manual* (this manual).
- *An Introduction to Smalltalk-80.*
- *4400 Series Operating System Reference Manual.*

- *4400 Series 'C' Programmer's Reference Manual.*
- *4400 Series Assembly Language Reference Manual.*
- RS-232 communications port cable.
- Two power cords; 1 for MSU, 1 for Display/CPU.
- Several blank keyboard overlays (for the function keys).

ASSEMBLE THE MOUSE

1. Remove the mouse and its rubber-coated steel ball from the shipping package.
2. Turn the mouse body on its back and remove the circular retainer by turning 1/4 turn in the direction of the arrows.
3. Put the rubber-coated ball in the cavity.
4. Replace the ball retainer and turn 1/4 turn against the direction of the arrows.

The mouse is now ready for use.

CONNECT THE CABLES

CAUTION

Check that both voltage indicators on the rear of the Display/CPU and the voltage indicator on the MSU indicate the proper voltage selection for your power source.

1. Plug the keyboard cable into the rear of the Display/CPU unit in the connector marked "KEYBOARD."
2. Plug the mouse cable into the rear of the Display/CPU unit in the connector marked "MOUSE" and secure the connector by pressing the retaining clip down.
3. If you will be using the 4405 as a terminal, plug one end of the RS-232 cable into the connector marked "COMPUTER." Secure the cable with the two small screws in the connector.
4. Plug the small end of the SCSI cable into the connector on the Display/CPU marked "MASS STORAGE INTERFACE." Match the keyway on the two connectors and press the male connector into the female connector without forcing it.

   Plug the large end of the SCSI cable into the upper connector (J1000) and secure it with the two small screws in the connector.

   Plug the SCSI terminator into the lower connector.

CAUTION

Match the keyway on the connectors and the SCSI terminator and SCSI cable plug before seating the plugs. The terminator must be installed and the cable must be secure to prevent noise and errors on the SCSI bus.

5. Inspect the power switches on the MSU and Display/CPU units. Turn them **OFF** if they are not already. (A green flag shows if the power switch is ON.)
6. Attach the female ends of the power cords to the MSU unit and the Display/CPU unit.
7. Insert the male end of the power cords into a properly grounded outlet that supplies the correct power for the 4405.

CAUTION

Attempting to operate the 4405 from an incorrect or improperly grounded voltage source can seriously damage the 4405's circuitry.

READ SECTION 1

Your 4405 is now ready for operation. Read through Section 1 *Introduction* to become familiar with the controls and terminology of the 4405, then go through Section 2 *The First Time* to verify that your system is working correctly. You should also read the rest of this manual for general information pertaining to the operation of the 4405.

NOTE

The 4405 time and date have been set to factory time. To set the date to your local time, login as "system" and give the command:

date <mm>-<dd>-<yy> <hr>:<min>[:<sec>] +s

(see the "4400 Series Operating System Reference Manual" for details of the "date" command.

Appendix B
CLEANING AND MAINTENANCE

GENERAL CLEANING

Your 4405 is a delicate instrument and has no user serviceable parts. You can clean the CRT screen and the external surface of the Display/CPU unit, the MSU unit, the mouse, and the keyboard with a soft cloth dampened with a solution of mild detergent and water. Use a soft brush with natural (not synthetic) fibers to clean dust from crevices. Refer all other cleaning and routine maintenance to a qualified technician. A yearly maintenance program is recommended in the *4405 Field Service Guide.*

CAUTION

To avoid damage to the plastics used in the Display Module, the Mass Storage Unit and Keyboard, do NOT use cleaning agents that contain benzene, acetone, toluene, xylene, or similar chemicals.

CAUTION

Never use a dripping wet cloth that could allow liquid to penetrate the case of the 4405. This could cause severe circuit damage, and in some cases expose you to hazardous voltages.

CLEANING THE MOUSE

If the mouse cavity becomes dirty, clean it by removing the rubber-coated ball and blow out the cavity with a gentle stream of compressed air. Do not use liquids to clean the ball or cavity.

CLEANING SPILLS ON THE KEYBOARD

If a container of liquid such as coffee, tea, or a soft drink should spill on the keyboard:

1. Immediately turn the keyboard upside down to keep the liquid from penetrating and causing further damage.
2. Call in a qualified service person to disassemble and clean the keyboard.

Appendix C
Options

Use this section to hold documentation for 4405 options that do not have binders of their own.

Appendix D
CONNECTING PERIPHERALS

INTRODUCTION

The 4405 has several external connectors which can be used to connect peripherals. These are:

- The SCSI bus.
- The RS-232 Communications Port
- The parallel printer port.
- The speaker output.
- The (Optional) Ethernet interface.

THE SCSI BUS

LOCATION

The SCSI bus connection is located on the rear of the Display/MSU unit labeled "MASS STORAGE INTERFACE."

SOFTWARE ACCESS

SCSI bus devices, addressed through drivers built into the operating system, each have two */dev* files associated with them. Each has a descriptive name as the normally used block device, and the same name with *c* appended as a raw or character-oriented device. The standard SCSI devices recognized by the system are:

- /dev/disk — The standard winchester disk.
- /dev/diskc — The raw (character-oriented) standard winchester disk.
- /dev/disk1 ... diskn — Optional winchester disks.
- /dev/disk1c ... diskn c — Raw optional winchester disks.
- /dev/floppy — The standard flexible disk.
- /dev/floppyc — The raw (character-oriented) flexible disk.
- /dev/tapec — The raw optional (character-oriented) streaming tape drive.

In order to read or write to any SCSI device during normal operation, you must *mount* the device. (See the 4405 Reference Manual command entry for *mount* for more information.) During *backup* and *restore* the flexible disk drive should not be mounted. These commands take care of the I/O in a different manner.

THE RS-232 COMMUNICATIONS PORT

LOCATION

The RS-232 Communications Port is located on the rear of the Display/CPU unit marked "computer."

SOFTWARE CONTROL

The communications port is addressed as */dev/comm*. It is a character-oriented device with read and write permissions.

You can examine or set the communications port parameters via the *commset* utility.

'C' programmers should check the files in */lib/include/sys* (particularly *sgtty.h* and *comm.h*) if they will be working with the communications port.

THE PARALLEL PRINTER PORT

LOCATION

The parallel printer port is located on the rear of the Display/CPU unit marked "copier."

SOFTWARE ACCESS

The printer port is identified by the special file */dev/printer*. It is a character-oriented device and has write permissions only.

Sample programs written in 'C' are located in directory */samples/printer*.

THE EXTERNAL SPEAKER JACK

LOCATION

The external speaker jack is located on the rear panel of the Display/CPU unit, marked "SPEAKER."

SPECIFICATIONS

The 4405 has a low power audio output jack capable of driving a small external speaker. The audio output is optimized for an eight-ohm speaker, but other common speakers can be driven (at reduced volume) without damage to the audio circuitry.

The external speaker jack is a small, two-conductor, switching phone jack. (When you insert an external speaker plug, the internal speaker is disabled.)

The output from either an internal or external speaker can be varied with the volume control located directly above the external speaker jack.

SOFTWARE ACCESS

You can drive the speaker via the device */dev/sound.* The *4400 Series Operating System Reference Manual* contains a discussion under *THE SOUND GENERATOR* that discusses driving */dev/sound.*

THE ETHERNET INTERFACE

The ethernet interface is a separate option. It is fully described in it's own manual, which you can include in this manual in *Appendix C, OPTIONS.*

Appendix E
SPECIFICATIONS

NOTE

In this preliminary manual, these specifications are taken from early engineering models of the 4405. Production models of the 4405 may have different specifications.

Table E-1
CPU/DISPLAY UNIT PHYSICAL DIMENSIONS

Characteristic	Supplemental Information
Weight	44 lbs. (20 Kg)
Length	19.5 in (495 mm)
Width	16.5 in (419 mm)
Height	13.89 in (352 mm)
Display Size	13 in (330mm)
Display Area	9.5 x 7.0 in (241x178 mm)

*These specifications do not include the Keyboard or Mass Storage Unit.

Table E-2
MASS STORAGE UNIT PHYSICAL DIMENSIONS

Characteristic	Supplemental Information
Weight	14 lbs (6.35 Kg)
Length	17 in (433 mm)
Width	14.5 in (368 mm)
Height	5 in (128 mm)

Table E-3
CPU/DISPLAY ELECTRICAL SPECIFICATIONS

Characteristic	Supplemental Information
Nominal Input Voltages: 115 V 230 V	 87--128 V 174--250 V
Maximum Input Power	200 W
Frequency Range	48–66 Hz
Fuse	4 A Slow-blow

Table E-4
MASS STORAGE UNIT ELECTRICAL SPECIFICAITONS

Characteristic	Supplemental Information
Nominal Input Voltages: 115 V	 87--128 V
230 V	174–250 V
Maximum Input Power	140 W
Frequency	50–60 Hz
Crest Factor	1.35 minimum
Fuse	2 A Slow blow (115 V)
	1 A Slow blow (230 VOS)

Table E-5
CPU/DISPLAY ENVIRONMENTAL SPECIFICATIONS

Characteristic	Supplemental Information
Temperature:	
Operating	+50 to +104 degrees F (+10 to +40 degrees C).
Nonoperating	-40 to +149 degrees F (-40 to +65 degrees C).
Altitude:	
Operating	To 10,000 ft (3,050 m)
Nonoperating	To 40,000 ft (12,200 m)
Humidity:	
Operating	0 to 75% relative humidity (non-cond.).
Nonoperating	0 to 95% relative humidity (non-cond.).
Vibration	Withstands 0 to 0.015 in displacement, at 0 to 55 Hz (all 3 major axes).
Shock	Main cabinet withstands a 20 g shock to all faces.
Electrostatic Immunity:	
Operating	No interruption of operation, loss of data, or change of operating mode from 15 kV shock.
Nonoperating	No damage to unit from 20 kV shock.

Table E-6
MASS STORAGE UNIT ENVIRONMENTAL SPECIFICATIONS

Characteristic	Supplemental Information
Temperature:	
Operating	+50 to +104 degrees F (+10 to +40 degrees C).
Nonoperating	-40 to +149 degrees F (-40 to +65 degrees C).
Altitude:	
Operating	To 10,000 ft (3,050 m)
Nonoperating	-1000 To 40,000 ft (12,200 m)
Humidity	80%, Maximum wet bulb: 26 degrees C, noncondensing
Vibration:	
Operating	2 to 22 Hz: 0.01 in (0.254 mm) displacement 22 to 500 Hz: 0.25 g constant acceleration
Nonoperating	2 to 22 Hz: 0.01 in (1mm) displacement 22 to 500 Hz: 1 g constant acceleration
Shock: nonoperating	20 g, halfsine, 11 ms duration; Three shocks in all horizontal directions 15 g, half sine, 11 ms duration; Three shocks in all vertical directions
Shock	Main cabinet withstands a 20 g shock to all faces.
Electromagnetic Compatibility	Meets the following requirements, with respect to conducted and radiated emissions: FCC Rules, Part 15, Subpart J, Class A VDE 0871/76 class B

Table E-7
INSTALLATION REQUIREMENTS

Characteristics	Supplemental Information
Heat Dissipation	525 BTU/hour (nominal line voltage) 575 BTU/hour (max. at maximum power supply load)
Surge Current	Total System = 34A (nominal at 120vac)
Clearance	Rear: 3 inches (for cabling) Top: 3 inches (for cooling) Sides: 2 inches (for cooling)
Distance from Equipment Generating Magnetic Fields	The system's display should not be located near equipment containing fans, motors or other electro-magnetics which can distort the display.

Table E-8
GRAPHICS CHARACTERISTICS

Characteristic	Supplemental Information
Type of display:	Monochrome raster-scan
Crt size:	15 inches (diagonal measurement)
Phosphor:$P4 short-persistence	
Usable display area:	240 mm by 180 mm, plus or minus 5 mm (9.4 in by 7.1 in, plus or minus 0.2 in)
Luminance:	25 foot-lamberts for "white" color.
Moire' pattern:	A slight moire' effect may be visible under some conditions.
Anti-reflective treatment:	Dark, non-glare transmission