3

Basic Commands and Functions

Learning the Basics

Now that your system is up and running, it is time to learn about OS-9's basic features and utility commands. This chapter and the chapter on the OS-9 file system provide a "fast-track" introduction to OS-9 designed to get you started quickly.

The secret of getting up to speed quickly with OS-9 is to first identify and learn only the basic, everyday functions necessary to run application programs and programming languages. It is fairly easy to learn more as you continue to work with the system.

The general topics covered in this chapter are:

- Logging on timesharing systems
- An introduction to the shell
- Use of the keyboard and display
- The page pause feature
- help, free, and mfree utilities

Logging on to a Timesharing System

If you are using a single user system such as a personal computer, you can skip this section. Otherwise, you need to know how to log on to a multi-user system. This applies to both ***hardwire*** and ***dial-up*** terminals.

Until you press the <return> key, idle terminals on multi-user systems do nothing but beep at you. Pressing the <return> key starts the log-on program called login. login's function is to maintain system security and start each user with a personalized environment.

The system asks you for your user name and the password the system manager assigned to you. The system echoes your user name but for security purposes, your password is not echoed. You have three chances to enter a valid user name and password.

An example of the login procedure is given below:

```
OS-9/68000 V2.4 Microware Systems P32 90/11/24 14:51:12

User Name: smith
Password: [not echoed]

Process #10 logged on  90/11/24 14:51:20

Welcome!

$
```

Depending on how the system is set up, a system-wide ***message of the day*** may be displayed on your screen. You can also automatically run one or more initial programs. In addition, you are normally set up in your own main working directory.

To log off, simply press the <escape> (end-of-file) key or type logout any time your main shell is active.

For more information, see the login and tsmon utility descriptions in the ***OS-9 Utilities*** section.

An Introduction to the Shell

Every operating system has a command interpreter. A command interpreter is a translator between the commands you type in and the commands the operating system understands and executes.

> + OS-9's command interpreter is called the shell.

The shell is normally started as part of the system startup sequence on a single user system or after logging on to a timesharing system. It is the primary interface with the system. When you enter a command, it is the shell's job to translate the command into something OS-9 can understand.

The shell provides many functions and options. A chapter is exclusively devoted to an in-depth discussion of the features available. This section is intended to provide just enough familiarity with the shell for you to run basic OS-9 commands.

The shell functions in two ways:

- Accepting interactive commands from your keyboard.
- Reading a sequence of command lines from a special type of file called a ***procedure file***. The shell executes each command line in the procedure file just as if the command lines had been typed in manually from the keyboard. Procedure files are a convenient way to eliminate typing frequently used, identical sequences of commands.

When the shell is ready for command input, it displays a $ prompt. You can now enter a command line followed by a carriage return.

The first word of the command line is the name of a command. It may be in upper or lower case. The command may be the name of:

- An OS-9 utility command
- An application program or programming language
- A procedure file

Most commands require or accept additional parameters or options. These parameters and options provide the program and/or the shell with additional information such as file names and directory names to search. Almost all options are preceded by a hyphen (-) character. All parameters are separated by space characters.

The shell follows a special searching sequence to locate the command in memory or on disk. If it cannot find the command you specified, the error #000:216, "file not found" is generally reported.

Here is an example of a simple shell command line:

```
$ list myfile
```

The name of the program is list. The file name myfile is passed to the program.

Using the Keyboard

Most input to OS-9, programming languages, and application programs is line oriented. This means that as you type, the characters are collected but not sent to the program until you press the <return> key. This gives you a chance to correct typing errors before they are sent to the program.

OS-9 has several features to make data entry and error correction simple. These are called ***line editing features***. Each of these features use control keys generated by simultaneously pressing the <control> key and some other character key.

The line editing control keys are:

Key	Function
<control>A	Repeats the previous input line. The last line entered is redisplayed but not executed. The cursor is positioned at the end of the line. You may enter the line as it is or you can add more characters to it. You can edit the line by backspacing and typing over old characters.
<control>D	Redisplays the current input line. This is mainly used for hardcopy terminals that cannot erase deleted characters.
<control>H	Backspaces to erase previous characters. Most keyboards have a special <backspace> key that can be used directly without using the <control> key.
<control>Q	Resumes the input and output previously stopped by <control>S. The <control>Q function is known as X-on.
<control>S	Halts input and output until <control>Q is entered. The <control>S function is known as X-off. This is a function used by many serial I/O devices such as printers to control output speed.
<control>W	Temporarily halts output so you can read the screen before data scrolls off. Output resumes when any other key is pressed. See the section on the page pause feature.
<control>X	Deletes line; erases the entire current line.
ESCAPE or <control>[	Indicates the end-of-file: all OS-9 I/O devices, including terminals, are accessed as files. This simulates the effect of reaching the end of a disk file.

There are also two important control keys called ***interrupt*** keys. They work differently than the line editing keys because they can be used at any time, not just when a program has requested input. They are normally used to halt or alter a running program.

Key	Function
<control>C	Sends an interrupt signal to the most recent program. This functions differently from program to program. If a program does not make specific interrupt provisions, it aborts the program. If a program has provisions for interrupts, <control>C usually provides a way to stop the current function and return to a master menu or command mode. In the shell, <control>C can be used to convert the ***foreground*** program to a ***background*** program, if the program has not begun I/O to the terminal.
<control>E	Sends a ***program abort*** signal to the program presently running. In most cases, this key prematurely aborts the current program and returns you to the shell.

The control keys described above are the key assignments commonly used in most OS-9 systems. The correspondence between control keys and their functions is changeable, so your keys may be different. You can use the tmode utility to redefine the function of control keys. This command allows you to customize OS-9 to the specific computer's keyboard layout.

NOTE: For more information about tmode, see the chapter on OS-9 system management and the ***OS-9 Utilities*** section.

The Page Pause Feature

The page pause feature eliminates the annoyance of having output scroll off the screen before you can read it. OS-9 counts output lines until a full screen has been displayed. It then halts output until you press any key. This is repeated for each screen of output.

Page pause can be fooled by lines longer than the physical width of the screen. These long lines wrap around to the next line. The system does not distinguish this, and consequently does not count them properly.

tmode may be used to turn this feature on and off, or to change the number of lines per screen:

Key	Function
tmode pause	Turns the page pause mode on.
tmode nopause	Turns the page pause mode off.
tmode pag=10	Sets the page length to ten lines.

Basic Utilities

OS-9 provides over seventy standard utilities and built-in shell commands. The majority of them are used rarely, if ever, by casual users. You will frequently use less than a dozen of them and less frequently use about a dozen more.

The utilities have been broken down into three groups to give you an idea of what you should and should not bother learning immediately. You should get acquainted with the first group now, and the second group as time permits. If you plan to do advanced programming or systems-level work, you can study the third group at your convenience.

Group 1: Basic Utilities

attr	backup	build	chd	chx	copy	date
del	deldir	dir	dsave	echo	edt	format
free	help	kill	list	makdir	merge	mfree
pd	pr	procs	rename	set	setime	shell
w	wait					

Group 2: Programmer Utilities

binex	cfp	cmp	code	compress	count	dump
ex	exbin	expand	frestore	fsave	grep	load
logout	make	printenv	profile	qsort	save	setenv
tape	tee	tmode	touch	tr	unsetenv	

Group 3: System Management Utilities

break	dcheck	deiniz	devs	diskcache	events	fixmod
ident	iniz	irqs	link	login	mdir	moded
os9gen	romsplit	setpr	sleep	tapegen	tsmon	unlink
xmode						

The Help Utility and the -? Option

The most important command to learn when beginning to use the OS-9 utilities is help. The help utility is an on-line quick reference manual. To use this utility, type help, a utility name, and a carriage return. The utility function, syntax, and available options are listed. For example, if you cannot remember the function or syntax of the backup utility, you could type help backup after the $ prompt:

```
$ help backup
Syntax: backup [<opts>] [<srcpath> <dstpath>] [<opts>]
Function: backup disks
Options:
        -b=<size>       use larger buffer (default is 4k)
        -r              don’t exit if read error occurs
        -v              do not verify
$
```

The descriptions are short and precise. Try it. This is a quick way to find information without looking up the utility in the documentation.

> + Typing help by itself displays the syntax and use of the help utility.

The same information is also available by typing the utility name followed by a question mark (-?). Each utility has the -? option.

Free and Mfree

During the format procedure, a disk is divided into data sectors of a pre-defined number of bytes. These sectors, in turn, are allocated into groups called ***clusters***. The number of sectors per cluster is dependent on the storage capacity and physical characteristics of the given device. This means that small amounts of free space, given in sectors, may not be divisible into the same number of files.

free displays the amount of unused disk space in the number of sectors and in the number of bytes. It also displays the disk name, its creation date and the cluster size of the device. For example:

```
$ free
“Tazz: /H0 Wren V” created on: Oct 6, 1989
Capacity: 2347860 sectors (256-byte sectors, 8-sector clusters)
1477296 free sectors, largest block 1356000 sectors
378187776 of 601052160 bytes (360.66 of 573.20 Mb) free on media (62%)
347136000 bytes (331.05 Mb) in largest free block
```

free uses a 4K buffer by default. To increase the buffer size, use the -b option. For example, to use a 10K buffer you could type:

```
$ free -b=10
```

or

```
$ free -b10
```

mfree displays the address and size of unused memory available for allocation. For example:

```
$ mfree
Current total free RAM:  164.00 K-bytes
```

For even more information concerning the unused memory, the -e option may be used with mfree. For example:

```
mfree -e
Minimum allocation size:       4.00 K-bytes
Number of memory segments:   6
Total RAM at startup:       8192.00 K-bytes
Current total free RAM:     2084.00 K-bytes

Free memory map:

   Segment Address        Size of Segment
  -----------------   --------------------------
     $5B000         $1000         4.00 K-bytes
     $5F000         $2000         8.00 K-bytes
     $99000         $1E3000     1932.00 K-bytes
     $29C000        $3000        12.00 K-bytes
     $2A1000        $1F000      124.00 K-bytes
     $2C5000        $1000         4.00 K-bytes
```

End of Chapter 3