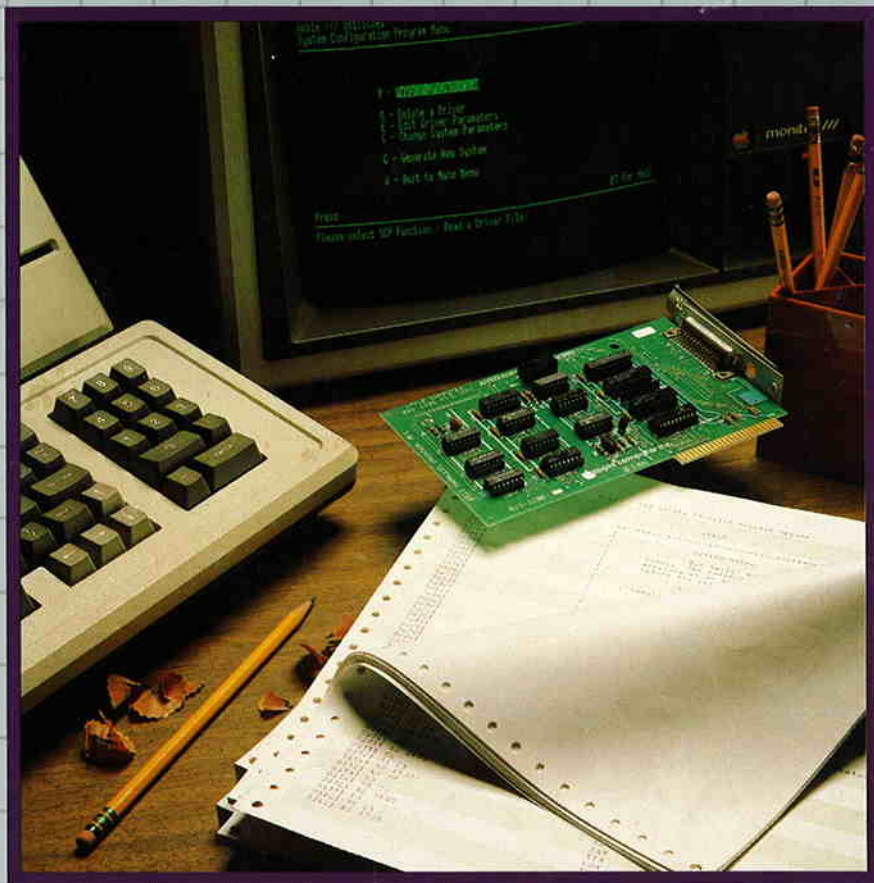


# Apple III

# SOS

## Device Driver Writer's Guide



## **Notice**

---

Apple Computer, Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

## **Disclaimer of All Warranties And Liabilities**

---

Apple Computer, Inc. makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. Apple Computer, Inc. software is sold or licensed "as is." The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer (and not Apple Computer, Inc., its distributor, or its retailer) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will Apple Computer, Inc. be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software, even if Apple Computer, Inc. has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer, Inc.

Written by Steve Hix. Special thanks to Bob Martin.

© 1982 by Apple Computer, Inc.  
20525 Mariani Avenue  
Cupertino, California 95014  
(408) 996-1010

The word Apple and the Apple logo are registered trademarks of Apple Computer, Inc. Simultaneously published in the U.S.A. and Canada

Reorder Apple Product #A3L0023

*Apple III*

*SOS Device Driver Writer's*

*Guide*



# Contents

## ***Introduction*** ***ix***

---

- x Why Device Drivers?
- x Who Uses Them?
- x How They Work
- xi Scope of this Manual
- xii Apple II Emulation Mode
- xiii Notations Used in this Manual

## ***1 Overview of SOS Device Drivers*** ***1***

---

- 4 SOS Device Classes
- 4 Character Driver Functions
- 5 DR\_INIT
- 5 DR\_OPEN
- 5 DR\_CLOSE
- 5 DR\_READ
- 5 DR\_WRITE
- 6 DR\_STATUS
- 6 DR\_CONTROL

- 6 Block Device Functions
- 6 DR\_INIT
- 6 DR\_READ
- 7 DR\_WRITE
- 7 DR\_REPEAT
- 7 DR\_STATUS
- 7 DR\_CONTROL
- 7 Conceptual Model of SOS
- 8 The Abstract Machine
- 9 SOS Data and Control Flow
- 10 Generalized Device Driver Model
- 11 Summary

## **2** ***The Physical Environment of SOS***

---

**13**

- 14 Hardware Diagram
- 14 SOS System Address Space
- 16 System Control Registers
- 16 E Register
- 17 Z Register
- 18 B Register
- 19 Memory Addressing
- 19 Bank-switched Addressing
- 19 Enhanced-Indirect Addressing
- 21 RS232 Serial Port
- 21 Receive/Transmit Data Register
- 21 Status Register
- 21 Command Register
- 22 Control Register
- 22 External Device Selection
- 22 \$C800 Selection

### **3 Request Handling** **23**

---

- 27 Driver Execution Environment
- 27 Zero- and Extended-address Page Usage
- 28 Driver Parameter Table
- 28 B Register
- 29 System Clock State
- 29 System Interrupt State
- 29 System I/O State
- 30 Internal Driver Structure
- 31 The Driver Information Block (DIB)
- 31 The DIB Header Block
- 36 The DIB Configuration Block
- 36 Storage and Communication Buffers
- 36 SOS Driver Requests
- 37 DR\_INIT
- 37 DR\_OPEN
- 38 DR\_CLOSE
- 38 DR\_READ
- 40 DR\_WRITE
- 40 DR\_REPEAT
- 41 DR\_STATUS
- 43 DR\_CONTROL

### **4 SOS-provided Services** **47**

---

- 49 System Resource Allocation
- 50 ALLOCSIR
- 51 DEALCSIR
- 51 I/O Expansion Selection
- 52 SELC800
- 52 Error Handling
- 53 SYSERR
- 53 System Errors
- 54 Event Handling
- 55 Event Queing
- 55 Event Recognition
- 56 QUEEVENT

## **5** ***Interrupt Handling*** **59**

---

- 60 Interrupt Handlers
- 61 Interrupt Handler Design
- 62 Interrupt Handler Environment
- 64 Interrupt Resources

## **6** ***Device Driver Coding Techniques*** **65**

---

- 66 General Driver Design
- 68 Writing Character Drivers
- 69 Writing Block Drivers
- 69 Writing for Interrupt-driven Devices
- 69 Creating Device Driver Code Files
- 70 Error Detection and Reporting

## **7** ***Interfacing with Apple III Peripheral Connectors*** **71**

---

- 72 Physical Description
- 73 Electrical Description
- 77 Design Techniques for Interface Cards
  - 77 Decoupling
  - 77 I/O Loading and Drive Rules
- 79 Timing Signals
- 80 Designing-in 6522s
- 82 Design Techniques for Apple III Prototyping Cards
- 83 Minimizing EMI
- 84 Safety and Testing
- 85 Programming Notes



## **Appendices**

**A** **Sample Block Driver Skeleton** **87**

**B** **Sample Character Driver Skeleton** **99**

**C** **6502B Instruction Set** **111**

**D** **Important Fixed Addresses** **121**

122 SOS Resources Available for Device Driver's Use

122 Addresses Important to Device Drivers

**Glossary** **123**

**Figures and Tables** **133**

**Index** **135**



## Introduction

The device driver is an essential and integral part of the Apple III operating system, hereafter referred to as SOS (Sophisticated Operating System). It is the part of SOS that supports all input and output (I/O) operations, regardless of the type of device being used.

In the world of SOS, everything external to the CPU and its memory address space is a file: to be opened, read, written to, and closed. Unlike many other computer systems, the type of device being used for I/O makes essentially no difference in the way that programs perceive and use them.

Device drivers write to and read from files. This manual tells you how to write device drivers and incorporate them into SOS. It assumes that you are familiar with both 6502 assembly-language programming and the information in the following four manuals:

*Apple III Owner's Guide*  
*Apple III Standard Device Drivers Manual*  
*Apple III SOS Reference Manual*  
*Apple III Pascal Program Preparation Tools*

If that assumption is not yet correct, we can resume when you return.

## **Why Device Drivers?**

---

Most of us are used to speaking with people who use and understand the same language that we do. When someone new moves into the neighborhood speaking another language, we can either learn the new language, find a translator, wait for the other person to learn your language, or else get by without communicating.

A computer system is like a neighborhood, and each different device connected to the computer “speaks differently”. If each application written to run on a computer is required to have its own routines to communicate with devices, a great amount of time (and money) is spent on needlessly duplicating effort. Rather than require users to write new interfacing programs or rewrite applications for each new device that they connect to their Apple III, SOS device drivers support uniform communication between applications and devices.

Device drivers become part of SOS and so are loaded each time the system is booted. All I/O in SOS is performed by device drivers.

## **Who Uses Them?**

Every part of the Apple III system that communicates with something or someone external to the Apple III's processor uses device drivers in SOS, and no I/O is done without them. Some device drivers are supplied with SOS, including .CONSOLE, .PRINTER, .AUDIO, and .RS232 ; they are described in the *Apple III Standard Device Drivers Manual*.

Other device drivers are supplied with the device that they serve, for example .PROFILE, supplied with the ProFile hard disk.

## **How They Work**

All SOS data flow is performed by device drivers through files. A file is a named, ordered sequence of bytes and may be used to store, transmit, or retrieve any type of information that you can put into the Apple III.

SOS recognizes two classes of files: character files and block files.

A character file is treated by SOS as an continuous stream of bytes. SOS can read or write the next byte in the stream, but it cannot reread or skip bytes in the stream.

A file sent to a character device, such as a printer, is a character device file. As far as a program running under SOS is concerned, there is no difference in the way it accesses any type of character device; all look like files to the program.

A file can also reside on a block device, such as a disk drive. A block file is composed of characters in groups called *blocks* of 512 bytes each. Blocks are numbered serially, but SOS can read from or write to any given block at will. A block file is limited to a maximum of \$FFFFFFE bytes, or 16,777,215 bytes.

A program can open, read, write, and close a character file, but cannot create, delete, or rename one. A character device file cannot be accessed as a random-access file; a block device file can be accessed randomly.

## ***Scope of this Manual***

---

This manual provides enough information for experienced assembly-language programmers to write device drivers for character and block devices to work with Apple III SOS.

This manual is not intended to be a tutorial covering basic programming or hardware-design techniques; we assume that you know them already.

Chapter 1 provides a general overview of the concepts underlying SOS device drivers.

Chapter 2 describes in general terms the underlying physical environment of SOS device drivers.

Chapter 3 describes request handling, the main “job” of device drivers.

Chapter 4 describes the services provided by SOS to aid device driver function, such as error reporting and resource allocation.

Chapter 5 describes interrupts and interrupt handling by SOS device drivers.

Chapter 6 presents techniques for developing device drivers.

Chapter 7 presents techniques for designing and building interface cards to connect with the Apple III through the backplane peripheral connectors.

Appendix A is a sample device driver skeleton that can be used as a starting point for writing drivers for block devices such as disks.

Appendix B is a sample device driver skeleton that can be used as a starting point for writing drivers for character devices such as printers.

Appendix C contains the instruction set of the 6502B, the microprocessor used by the Apple III.

Appendix D contains a list of system addresses that are important to device driver writers.

## ***Apple II Emulation Mode***

---

The Apple III also offers an Apple II Emulation mode. In this mode, the Apple III functions as a 48K Apple II or Apple II Plus with a disk controller card in slot 6, and a serial (either Communication or Serial) interface card in slot 5 or 7. There is no “slot 0”. Other limitations of Emulation mode operation are:

- No software requiring the *Language card* will run on an Apple III in Emulation mode.

- Only the built-in disk drive and the first external drive will be usable. Daisy-chaining additional drives is not supported.
- The RGB video output will only generate black and white images in HIRES graphics.
- There is no cassette port.
- DMA and interrupts are not supported.

## ***Notations Used in this Manual***

---

Three symbols appear throughout this manual to point out particularly important information:

---



A hand indicates information of an especially useful nature, which may not be very obvious at first sight.



An eye points out some characteristic of the software or hardware operation that you should be careful about.



A stop sign draws your attention to something that may have serious consequences if not used properly, such as damaging the Apple III or causing a serious error, or complete shutdown of system operation.





# Overview of SOS Device Drivers

- 4 SOS Device Classes
- 4 Character Driver Functions
  - 5 DR\_INIT
  - 5 DR\_OPEN
  - 5 DR\_CLOSE
  - 5 DR\_READ
  - 5 DR\_WRITE
  - 6 DR\_STATUS
  - 6 DR\_CONTROL
- 6 Block Device Functions
  - 6 DR\_INIT
  - 6 DR\_READ
  - 7 DR\_WRITE
  - 7 DR\_REPEAT
  - 7 DR\_STATUS
  - 7 DR\_CONTROL
- 7 Conceptual Model of SOS
- 8 The Abstract Machine
- 9 SOS Data and Control Flow
- 10 Generalized Device Driver Model
- 11 Summary

# 1

## **Overview of SOS Device Drivers**

The Apple III/SOS system deals with all input and output (I/O) in the same way: all devices connected to the system are files, communicating with SOS through device drivers.

Every device driver has one or more physical devices associated with it. For example, a block device driver has one or more block devices, a format device driver has one or more format devices, and so on.

SOS communicates to attached devices (keyboard, screen, printers, disks, and so on) by sending device requests to direct the operation of each device by its device driver. Remember that all devices connected to SOS are files.

A device driver is a memory-resident module that implements the set of SOS device requests (through request handlers) required of all devices connected to SOS. In addition to device requests, a device driver also performs interrupt handling (with interrupt handlers) for devices using interrupts.

At system startup, device drivers reside in a file called SOS.DRIVER on the boot volume. You can change the content of SOS.DRIVER with the SOS System Configuration Program (SCP) described in the *Apple III Standard Device Drivers Manual*. SCP lets you reconfigure your operating system by adding or removing device drivers. Note that SCP also checks the validity of your device driver's format.

When a device driver is called, the SOS device manager passes a request table to the device driver defining the type of operation to be done. These operations are called device requests, and each device driver has a specific set of device requests that it must perform for its own device. SOS device requests are briefly described later in this chapter, and in detail in Chapter 3.

A standard group of device drivers comes with every Apple III system to enable the operation of the Apple III's built-in devices, such as speaker, screen, keyboard, and RS232 serial port. These device drivers are described in the *Apple III Standard Device Drivers Manual*.

When you obtain an optional accessory device that can be connected to your Apple III, the device driver needed to operate it is also supplied.

Table 1-1 lists some important device drivers and the devices they serve.

Device Driver	Device(s) Served
(names as supplied)	
.CONSOLE	Screen and Keyboard
.PRINTER .RS232	Apple III serial port
.AUDIO	Apple III speaker
.GRAFIX	Apple III graphics display
.D1 through .D4	Disk III disk drives
.PROFILE	ProFile hard disk

**Table 1-1.** SOS Device Drivers and Devices

All the device drivers listed in Table 1-1 except .PROFILE and the Disk III drivers .D2 through .D4 operate built-in devices, and all except .PROFILE are supplied with the Apple III system software package. The .PROFILE driver is supplied with the ProFile hard disk, and is typical of device drivers supplied with Apple III optional devices. Its use is described in the documentation supplied with the ProFile hard disk.

### ***SOS Device Classes***

---

There are two classes of devices (and device drivers) within Apple III SOS: character devices and block devices.

Character devices, such as printers and modems, can transfer information in sequential character streams up to 64K bytes in length at one time.

Block devices, such as disks, transfer information in 512-byte blocks. Any higher orders of organization, such as files and directories, are the responsibility of SOS.

A subclass of the block device driver is the format driver, used to format a block device before use. A format device driver may either be part of a block device driver or stand alone. A format driver should be included as part of the device driver except when the format driver is very large. In such a case, memory limitations would dictate the need for a stand alone format driver.

Examples of stand alone format device drivers are .FMTD1 through .FMTD4, found on the SOS Utilities diskette and used by SCP to format diskettes.

### ***Character Driver Functions***

Character device drivers move character streams either in one direction, like .PRINTER, or bidirectionally, like .RS232. .

**Character drivers must support NEWLINE mode.** This allows the use of a single character to mark a logical end of record in a character stream. The NEWLINE character may be defined any number of times through DR\_CONTROL device requests.

The SOS device requests performed by character device drivers are described briefly below, and in greater detail in Chapter 3. Device requests are issued by the SOS device manager.

### *DR\_INIT*

DR\_INIT operates once only (during system startup) to prepare the device driver for use. The device served by the driver is not accessed and remains closed, and no resources are allocated.

### *DR\_OPEN*

DR\_OPEN is called to allocate a resource from the system: in this case, to open its device file to be either written to or read from.

### *DR\_CLOSE*

DR\_CLOSE is called to perform two operations: it shuts down its device, and it deallocates the system resources assigned to the driver and gives them back to the system.

### *DR\_READ*

DR\_READ is called to read a specified number of characters from its character device into a buffer in memory.

### *DR\_WRITE*

DR\_WRITE is called to write a specified number of characters from a buffer in memory out to the character device.

## *DR\_\_STATUS*

*DR\_\_STATUS* is called to provide information on the current status of its device. In addition to the device's status, other information specific to a given device or driver may be returned.

## *DR\_\_CONTROL*

*DR\_\_CONTROL* is called to reset the device, load control parameters, reset the NEWLINE character (described in Chapter 3), or make other changes to the device's operating parameters.

## ***Block Driver Functions***

Block devices move data in 512-byte blocks, and allow SOS to access easily any given logical block of a block device.

A block driver's device is divided into consecutively-numbered logical blocks; higher orders of organization (such as files or directories) on the device are handled outside the driver.

The SOS device requests implemented by block device drivers are briefly described below and in detail in Chapter 3.

## *DR\_\_INIT*

*DR\_\_INIT* is called during system startup to perform operations required to prepare the device for use, allocate resources needed by the driver, and open the device. A *DR\_\_INIT* request for a block device is equivalent to requesting *DR\_\_INIT* and *DR\_\_OPEN* for a character device.

## *DR\_\_READ*

*DR\_\_READ* is called to read one or more blocks from the block device, beginning at a specified logical block number.