Preface

This preface provides a brief description of the SunOS™ assembler that runs on the SPARC® operating environment and also includes a list of documents that can be used for reference.

The SunOS assembler that runs on the SPARC operating environment, referred to as the "SunOS SPARC" in this manual, translates source files that are in assembly language format into object files in linking format.

In the program development process, the assembler is a tool to use in producing program modules intended to exploit features of the SPARC architecture in ways that cannot be easily done using high level languages and their compilers.

Whether assembly language is chosen for the development of program modules depends on the extent to which and the ease with which the language allows the programmer to control the architectural features of the processor.

The assembly language described in this manual offers full direct access to the SPARC instruction set. The assembler may also be used in connection with SunOS 5.x macro preprocessors to achieve full macro-assembler capability. Furthermore, the assembler responds to directives that allow the programmer direct control over the contents of the relocatable object file.

This document describes the language in which the source files must be written. The nature of the machine mnemonics governs the way in which the program's executable portion is written. This document includes descriptions of the pseudo operations that allow control over the object file. This facilitates the development of programs that are easy to understand and maintain.

Before You Read This Book

You should also become familiar with the following:

- Manual pages: `as(1)`, `ld(1)`, `cpp(1)`, `elf(3f)`, `dis(1)`, `a.out(1)`
- *SPARC Architecture Manual* (Version 8 and Version 9)
- ELF-related sections of the *Programming Utilities Guide* manual
- SPARC Applications Binary Interface (ABI)

How This Book is Organized

This book is organized as follows:

Chapter 1, "SPARC Assembler for SunOS 5.x," discusses features of the SunOS 5.x SPARC Assembler.

Chapter 2, "Assembler Syntax," describes the syntax of the SPARC assembler that takes assembly programs and produces relocatable object files for processing by the link editor.

Chapter 3, "Executable and Linking Format," describes the relocatable ELF files that hold code and data suitable for linking with other object files.

Chapter 4, "Converting Files to the New Format," describes how to convert existing SunOS 4.1 SPARC assembly files to the SunOS 5.x assembly file format.

Chapter 5, "Instruction-Set Mapping," describes the relationship between hardware instructions of the SPARC architecture and the assembly language instruction set.

Appendix A, "Pseudo-Operations," lists the pseudo-operations supported by the SPARC assembler.

Appendix B, "Examples of Pseudo-Operations," shows some examples of ways to use various pseudo-operations.

Appendix C, "Using the Assembler Command Line," describes the available assembler command-line options.

Appendix D, "An Example Language Program," describes an example C language program with comments to show correspondence between the assembly code and the C code.

Appendix E, "SPARC-V9 Instruction Set," describes the SPARC-V9 instruction set and the changes due to the SPARC-V9 implementation.

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Table P-2 Shell Prompts

Shell	**Prompt**
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`