# Preface

This preface is a brief description of the SunOS™ assembler that runs on x86. This preface also includes a list of documents that can be used for reference.

The SunOS assembler that runs on x86, referred to as the "SunOS x86" 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 Intel® 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 x86 instruction set. The assembler may also be used in connection with SunOS 5.1 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*

Use the following documents as references:

- Intel 80386 Programmer's Reference Manual
- i486™ Microprocessor Programmer Reference Manual (1990)
- Intel 80387 Programmer's Reference Manual (1987)
- *System V Application Binary Interface Intel 386 Processor Supplement*
- *System V Application Binary Interface*
- SVID System V Interface Definition

You should also become familiar with the following:

- Man pages: `as`(1), `ld`(1), `cpp`(1), `mn(4)`,`cof2elf`(1) (elf - *Executable and Linking Format*), `elf`(3E), `dis`(1), `a.out`(5).
- ELF-related sections of the *Programming Utilities* manual.

## *How This Book Is Organized*

This document is organized into the following chapters:

**Chapter 1, "Assembler Input,"** describes the overall structure required by the assembler for input source files.

**Chapter 2, "Instruction-Set Mapping,"** describes the instruction set mappings for the SunOS x86 processor.

**Chapter 3, "Assembler Output,"** provides an overview of ELF (*Executable and Linking Format*) for the relocatable object files produced by the assembler.

**Appendix A, "Using the Assembler Command Line,"** describes the assembler command line options.

## 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.<br>Use `ls -a` to list all files.<br>`machine_name% You have mail.` |
| **`AaBbCc123`** | What you type, contrasted with on-screen computer output | `machine_name%` **`su`**<br>`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*.<br>These are called *class* options.<br>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 | `#` |