Preface

The Programming Utilities Guide provides information for developers about the special built-in programming tools available in the SunOS system.

Who Should Use This Book

This guide is intended for application programmers who are using the Solaris 2.x system.

Before You Read This Book

Readers of this guide are expected to understand the Solaris 2.x operating system, programming, and networking.

How This Book Is Organized

This guide has several chapters, each discussing a unique topic. Each chapter describes a tool that can aid you in programming. These include:

Tracing Program Execution With the TNF Utilities
The Trace Implementation Format lets you insert probe points into source code to collect data for analysis.

lex–Lexical Analysis
Generates programs to be used in simple lexical analysis of text. It is a tool that solves problems by recognizing different strings of characters.

yacc — A Compiler Compiler
A tool for generating language parsers. It is a tool that imposes structure on computer input and turns it into a C language function that examines the input stream.

make Utility
Automatically maintains, updates, and regenerates related programs and files.

SCCS Source Code Control System
SCCS allows you to control access to shared files and to keep a history of changes made to a project.

m4 Macro Processor
A macro language processor. Creates library archives and adds or extracts files.

A System V make
Describes a version of `make`(1) that is compatible with older versions of the tool.

Other tools of interest, documented more completely in the *SunOS Reference Manual*, are listed briefly here.

`ar(1)`
Creates and maintains portable libraries or archives

`cpp(1)`
The C language preprocessor

`dis(1)`
An object code disassembler for COFF

`dump(1)`
Dumps (displays) selected parts of an object file

`lorder(1)`
Finds an ordering relation for an object library or archive

`mcs(1)`
Lets you manipulate the comments section of an ELF object file

`nm(1)`
Prints a name list of an object file

`size(1)`
Displays the size of an object file

`strip(1)`
Removes symbols and relocation bits from an object file

`tsort(1)`
Performs a topological sort

`unifdef(1)`
Resolves and removes `ifdef`'ed lines from C program source.

Related Books

- *SunOS Reference Manual*

What Typographic Changes and Symbols Mean

Command names, C code, UNIX code, system calls, header files, data structures, declarations, short examples, file names, and path names are printed in `listing (constant width) font`.

User input is in `listing font` when by itself, or **`bold listing font`** when used in combination with computer output.

Items being emphasized, variable names, and parameters are printed in *italics.*

```
Screens are used to simulate what a user will see on a
video display screen or to show program source code.

Data structure contents and formats are also shown in screens.
```

Caution – The caution sign is used to show possible harm or damage to a system, an application, a process, a piece of hardware, etc.

Note – The note sign is used to emphasize points of interest, to present parenthetical information, and to cite references to other documents and commands.