Figures, Tables, and Listings

Preface

About This Book ix

Figure P-1 680x0 bit numbering xiii
Figure P-2 PowerPC bit numbering xiii
Table P-1 Sizes of memory operands xiii

Chapter 1

Introduction to PowerPC System Software 1-1

Figure 1-1 The system software for PowerPC processor-based Macintosh computers 1-5
Listing 1-1 Creating a routine descriptor 1-17
Listing 1-2 The definition of the `NewControlActionProc` routine 1-18
Listing 1-3 Creating a routine descriptor for a control action procedure 1-19
Figure 1-2 Creating imports in a fragment 1-24
Listing 1-4 Testing for unresolved soft imports 1-25
Figure 1-3 A transition vector 1-27
Figure 1-4 The structure of a PowerPC application 1-31
Figure 1-5 The structure of a 680x0 application 1-32
Listing 1-5 The Rez input for a sample `'cfrg'` resource 1-32
Figure 1-6 The structure of a fat application 1-33
Figure 1-7 The structure of an accelerated resource 1-35
Listing 1-6 Rez input for a list definition procedure stub 1-35
Figure 1-8 The structure of a private resource 1-36
Listing 1-7 Using an accelerated resource 1-37
Listing 1-8 Some acceptable global declarations in an accelerated resource 1-39
Listing 1-9 Some unacceptable global declarations and code in an accelerated resource 1-39
Listing 1-10 Using a private resource 1-40
Figure 1-9 A 680x0 stack frame 1-42
Figure 1-10 The PowerPC stack 1-44
Figure 1-11 The structure of a stack frame's linkage area 1-45
Figure 1-12 The Red Zone 1-46
Figure 1-13 The organization of the parameter area on the stack 1-49
Figure 1-14 Organization of memory when virtual memory is enabled 1-54
Figure 1-15 Organization of memory when virtual memory is not enabled 1-56
Listing 1-11 Declaring an application's QuickDraw global variables 1-59
Figure 1-16 The structure of a PowerPC application partition 1-60
Listing 1-12 A sample 680x0 VBL task definition 1-61
Listing 1-13 A conditionalized VBL task definition 1-62
Listing 1-14 Patching an Operating System trap 1-67
Figure 1-17 The Memory control panel for PowerPC processor-based Macintosh computers 1-69
Listing 1-15 Waiting to call the `WaitNextEvent` function 1-72

Chapter 2 Mixed Mode Manager 2-1

Figure 2-1 680x0 and PowerPC procedure pointers 2-5
Figure 2-2 Calling PowerPC code from a 680x0 application 2-9
Figure 2-3 The stack before a mode switch 2-10
Figure 2-4 A 680x0-to-PowerPC switch frame 2-11
Listing 2-1 Sample glue code for a 680x0 routine 2-12
Figure 2-5 A PowerPC-to-680x0 switch frame 2-13
Figure 2-6 Procedure information for a stack-based routine 2-17
Figure 2-7 Procedure information for a register-based routine 2-19
Table 3-1 Limits on the number of specifiable parameters in a procedure information 2-20
Listing 2-2 Creating global routine descriptors 2-21
Listing 2-3 Creating local routine descriptors 2-22
Listing 2-4 Creating static routine descriptors 2-23
Listing 2-5 Building a static routine descriptor 2-23
Figure 2-8 General structure of an executable code resource 2-25
Figure 2-9 General structure of a fat resource 2-26

Chapter 3 Code Fragment Manager 3-1

Listing 3-1 Pseudocode for the version-checking algorithm 3-9
Listing 3-2 Loading a resource-based fragment 3-11
Listing 3-3 Loading a disk-based fragment 3-11
Listing 3-4 The Rez input for a typical application's `'cfrg'` resource 3-12
Listing 3-5 The Rez input for a typical import library's `'cfrg'` resource 3-13
Listing 3-6 Finding symbol names 3-14
Figure 3-1 Structure of a compiled code fragment (`'cfrg'`) resource 3-29
Figure 3-2 The format of a code fragment information record 3-30

Chapter 4 Exception Manager 4-1

Listing 4-1 Installing an exception handler 4-6
Listing 4-2 A native exception handler 4-8