List of Figures

Chapter 2

Figure 2−1. Example lpp_name File. . . . . 2−19

Chapter 3

Figure 3−1. Aggregate Data Definitions. . . . . 3−17
Figure 3−2. Sample Bit-Field Layout. . . . . 3−18
Figure 3−3. Run-Time Stack. . . . . 3−24
Figure 3−4. Stack Frame Layout. . . . . 3−28
Figure 3−5. Saving LR, n GPRs, and no FPRs. . . . . 3−30
Figure 3−6. Saving LR, n GPRs, and m FPRs. . . . . 3−30
Figure 3−7. Saving GPRs and FPRs When the LR is Saved/Restored by Inline Code. . . . . 3−31
Figure 3−8. Prolog/Epilog of Function Saving m FPRs and no GPRs. . . . . 3−31
Figure 3−9. FPR and LR Save. . . . . 3−32
Figure 3−10. FPR and LR Restore. . . . . 3−33
Figure 3−11. FPR Save. . . . . 3−34
Figure 3−12. FPR Restore. . . . . 3−34
Figure 3−13. GPR and LR Save. . . . . 3−36
Figure 3−14. GPR and LR Restore When FPRs Are Not Saved. . . . . 3−37
Figure 3−15. GPR Save. . . . . 3−39
Figure 3−16. GPR Restore. . . . . 3−39
Figure 3−17. Computing a New SP and Saving the Old SP. . . . . 3−42
Figure 3−18. Computing a New SP and Saving the Old SP. . . . . 3−42
Figure 3−19. Run-Time Stack for a Function that Calls alloca. . . . . 3−44
Figure 3−20. Caller With Prototype Definition for Callee. . . . . 3−47
Figure 3−21. Caller Without Prototype Definition for Callee. . . . . 3−48
Figure 3−22. Function Invocation Using Argument Area. . . . . 3−48
Figure 3−23. Function Descriptor. . . . . 3−49
Figure 3−24. Instructions Supporting Multiple TOCs and Shared Libraries. . . . . 3−50
Figure 3−25. Load Instruction Replacing ori. . . . . 3−50
Figure 3−26. Assembly Code for glink Routine. . . . . 3−51
Figure 3−27. Assembly Code for .ptrgl Routine. . . . . 3−52
Figure 3−28. Address of Descriptor Being Passed in GPR11. . . . . 3−52
Figure 3−29. Properly-Aligned Bit-Fields in Multiples of Eight. . . . . 3−64
Figure 3−30. Canonical Representation for an Array. . . . . 3−65
Figure 3−31. Canonical Representation for an Array of Pointers. . . . . 3−65
Figure 3−32. Canonical Representation for a Structure Declaration. . . . . 3−66
Figure 3−33. Canonical Representation for a Union. . . . . 3−67
Figure 3−34. General String Constructed for precis. . . . . 3−69
Figure 3−35. Main FORTRAN Program. . . . . 3−70
Figure 3−36. Assumed FORTRAN Data Declarations. . . . . 3−70
Figure 3−37. FORTRAN Function. . . . . 3−71

Figure 3−38. Interface to Hash Function. . . . . 3−71
Figure 3−39. Source Code for hash.c. . . . . 3−73
Figure 3−40. RBYTES Data Structure with Data. . . . . 3−78
Figure 3−41. Data/Stack in the Small Model Memory Segment Layout. . . . . 3−80
Figure 3−42. Large Memory Model Segment Layout. . . . . 3−81
Figure 3−43. The <systemcfg.h> Header File. . . . . 3−83

Chapter 4

Figure 4−1. Sample XCOFF Object File Organization. . . . . 4−5
Figure 4−2. Sample XCOFF a.out File Structure. . . . . 4−6
Figure 4−3. Loader Section Structure. . . . . 4−19
Figure 4−4. Sample Symbol Table Organization. . . . . 4−38

Chapter 5

Figure 5−1. Loader Section Structure . . . . . 5−4

Chapter 7

Figure 7−1. Archive File Organization. . . . . 7−4
Figure 7−2. Fixed-Length Header Format. . . . . 7−4
Figure 7−3. Member Header. . . . . 7−5

Chapter 9

Figure 9−1. Root File System Hierarchy. . . . . 9−4