Systems Programming: Reading

John Stavrakakis

The course textbook is:

Title: Computer Systems: A Programmer's Perspective
Author/s: Randal E. Bryant and David R. O'Hallaron
ISBN: 9781292101767
Edition: 3
Publisher: Pearson Education
Publish Year: 2016
Publish Location: Boston
Note: It is the Global, 3rd edition. 1st and 2nd edition also cover course contents very well.

1 List of weekly readings from the textbook

Week 1

- 1.1, 1.2
- Optional 1.3 - 1.6

Week 2

- 1.7.4
- 2.1
- 3.8.1 - 3.8.2
- 3.10.1
- 10.1
- 10.9

Week 3

- 3.9.1 - 3.10.2
- 10.1 - 10.4
- 10.10
- Useful tips in 10.11

Week 4

- 3.4.4 (assembly not covered, but useful to know about these primitives)
- 3.7 - 3.7.1
- 3.10.3
- 9.9 - 9.9.2
- 9.11 - very useful to new C programmers

Week 5

- Function pointers

  3.10 page 312-315
- Signals 8.5

  The book covers signals as part of processes. This is logical organisation. We will be discussing processes later, but we emphasise that signals can be sent/received within only one process. Example: to trigger system events such as File I/O, or a user based control mechanism.
- Error handling Appendix A (page 1077)
- Files revisited

  10.1 - 10.4

  10.5 (good I/O style with signal handling)

  10.10 - 10.12

Week 6

- The reading for week 6 is about the Preprocessor, Compilation and Linking of C programs.

  I recommend from Sections 7.1 - 7.5.

  (TLDR) We do want you to grasp the idea of a symbol table as a necessary component for linking.

  The entirety of Chapter 7 of the textbook serves as an excellent reference for linking. We will not assess your ability to implement the structure of the symbol table, ELF layout, linking algorithm, PIC etc. The lecture will only cover a subset of this process.

  The book does not cover preprocessor, and we recommend:
  Brian W. Kernighan and Dennis M. Ritchie. The C Programming Language, 1988 Chapter 4, section 11.

  The USYD library has an online version. Brian W. Kernighan and Dennis M. Ritchie. The C Programming Language, 1988

  Most textbooks will have a section on preprocessor. If you are looking for a more comprehensive guide on the topic, please see GNU docs on preprocessor: `https://gcc.gnu.org/onlinedocs/cpp/` Just be aware there are specific preprocessor macros for GNU that are not part of the C standard.

Week 7

- Processes parts of chapter 8
  8.2 - 8.4

Week 8

- IPC
- Memory mapping 9.8
- Basic concurrency and IPC (pipes) 12.1, 12.2
- Debugging
  3.10.2 page 315 - 320

  $ man assert

  Warning: Print statements are OLD style of debugging. There are reference books that advocate macros and such to debug code. This is not the only way to discover, and resolve software defects.

Week 9

- We could say that the remainder of the course on parallelism and concurrency is on chapter 12

  Concurrent programming with threads
  12.3 (Optional 12.1-12.2)

Week 10

- From the textbook 12.4 - 12.7 sharing
- Semaphores on pg1038

Week 11 and 12

- Recursive procedures
  3.7.6 pg289-291
- Performance and memory
- Chapter 5 Optimising program performance
  As much as you can bear
- Chapter 6 The memory hierarchy
  6.2, 6.5, 6.6
- Chapter 9 Virtual memory
  9 - 9.2, 9.8