The Book of the Class for cpe 453  
Spring 2019

Phillip L. Nico  
Department of Computer Science  
California Polytechnic State University  
San Luis Obispo, CA 93407  
pnico@calpoly.edu

DRAFT(June 5, 2019 at 12:56)

Abstract

These are my lecture notes and nothing more. That means that they are full of omissions and errors and were meant only as a means of helping me remember what I intended to talk about in class. They may or may not bear any resemblance to what was actually said. They may, however, be useful in providing some small insight into the author’s state of mind and approach to the course material.
## Contents

0 Syllabus 1

1 Lecture: Introduction and Background 11
1.1 Robust Programming 11
1.2 Background: The Ballad of the Unknown Stuntman 12
1.3 Syllabus 12
1.4 How to succeed in this course 16
1.5 The Last Page 16
1.6 Foundations 16
1.6.1 What an operating system is 16
1.6.2 Virtualization and Transparency 17
1.6.3 What an operating system is not 17
1.7 Major subjects for the quarter 17
1.8 This quarter 18

2 Lecture: History and Definition of an O.S. 19
2.1 Announcements 19
2.2 About the lab 20
2.3 About the assignment 20
2.4 Everything you wanted to know about C development 20
2.5 Aside: Review of 357 20
2.5.1 Review of Unix IO 20
2.5.2 Processes, etc. 21
2.5.3 Compilation 21
2.5.4 Tool of the week: Make(1) 21
2.6 Defining of an Operating System: The System Calls 22

3 Lecture: The Process Model 23
3.1 Announcements 23
3.2 Two stories 24
3.3 Defining of an Operating System: The System Calls 24
3.4 System Calls Again 24
3.5 System Call Mechanisms 24
3.5.1 How to do it 24
3.6 OS Pre-history: The boot process 25
3.6.1 How it all begins (on an Intel PC with a floppy) 25
3.6.2 How it continues 25
3.6.3 And onwards: How does the OS get control back? 25
3.7 OS History 29
3.7.1 Ancient times 29
3.7.2 The middle ages 29
3.7.3 The renaissance: families 29
3.7.4 Modern Times 30
4 Lecture: Operating System Structures
4.1 Announcements ........................................... 31
4.2 From last time: History .................................. 32
  4.2.1 The middle ages .................................... 32
  4.2.2 The renaissance: families .......................... 32
  4.2.3 Modern Times ....................................... 33
4.3 From last time: The Process (Users' View) ............... 33
4.4 The Operating System's view: The context ............... 34
4.5 Example of a context switch ............................ 34
4.6 System Calls Again ...................................... 36
4.7 System Call Mechanisms ................................ 36
  4.7.1 How to do it ........................................ 36
  4.7.2 And onwards ........................................ 36

5 Lecture: Intro. to Concurrency .............................. 37
5.1 Announcements ........................................... 37
5.2 Onwards .................................................. 37
5.3 The Process Model: a little deeper ....................... 37
5.4 Pseudoparallelism and nondeterminism ................... 38
5.5 Possible process states .................................. 38
5.6 Scheduling ............................................... 39
5.7 What about IPC? ......................................... 39
5.8 Operating System Structures ............................. 39
  5.8.1 Monolithic Systems .................................. 40
  5.8.2 Layered Systems ..................................... 40
  5.8.3 Virtual Machines .................................... 41
  5.8.4 Client-server model ................................. 41
5.9 The Layered Architecture of Minix ....................... 41
5.10 Example: Description a MINIX disk interrupt .......... 42

6 Lecture: Lightweight Processes ............................. 43
6.1 Announcements ........................................... 43
  6.1.1 From last time: Client-server model ................. 44
6.2 The Layered Architecture of Minix ....................... 44
6.3 Example: Description a MINIX disk interrupt .......... 44
6.4 Once again: Processes .................................... 45
6.5 Lightweight Processes: Threads ......................... 45
6.6 Introduction to Asgn2 ................................. 46
  6.6.1 Nine little functions ............................... 46
  6.6.2 Demonstration: threading in action ................. 46
  6.6.3 Review: context switch ............................. 47
  6.6.4 A thread's context: stack and registers .......... 47
6.7 What's this about libraries, then? ....................... 51
<table>
<thead>
<tr>
<th>Lecture: Concurrency and Synchronization</th>
<th>52</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Announcements</td>
<td>52</td>
</tr>
<tr>
<td>7.2 LWP</td>
<td>52</td>
</tr>
<tr>
<td>7.3 Problems with parallelism: Race Conditions</td>
<td>53</td>
</tr>
<tr>
<td>7.3.1 Race condition, defined</td>
<td>53</td>
</tr>
<tr>
<td>7.4 Critical Sections</td>
<td>53</td>
</tr>
<tr>
<td>7.5 Mutual Exclusion</td>
<td>54</td>
</tr>
<tr>
<td>7.6 Busy Waiting: Software Only</td>
<td>54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lecture: More Synchronization</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Announcements</td>
<td>55</td>
</tr>
<tr>
<td>8.2 LWP?</td>
<td>55</td>
</tr>
<tr>
<td>8.3 From Last time: Busy waiting</td>
<td>56</td>
</tr>
<tr>
<td>8.4 Onwards: Busy Waiting: With Hardware Support</td>
<td>56</td>
</tr>
<tr>
<td>8.5 Reflection</td>
<td>57</td>
</tr>
<tr>
<td>8.5.1 sleep() and wakeup()</td>
<td>57</td>
</tr>
<tr>
<td>8.6 Synchronization without busy waiting</td>
<td>58</td>
</tr>
<tr>
<td>8.7 Semaphores</td>
<td>58</td>
</tr>
<tr>
<td>8.7.1 Monitors: (Hoare 1974, Brinch Hansen 1975)</td>
<td>59</td>
</tr>
<tr>
<td>8.8 More Interprocess Communication</td>
<td>59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lecture: Even More Concurrency</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 Announcements</td>
<td>61</td>
</tr>
<tr>
<td>9.2 Synchronization without busy waiting</td>
<td>61</td>
</tr>
<tr>
<td>9.3 Sleep and Wakeup</td>
<td>61</td>
</tr>
<tr>
<td>9.4 Semaphores</td>
<td>62</td>
</tr>
<tr>
<td>9.4.1 Monitors: (Hoare 1974, Brinch Hansen 1975)</td>
<td>62</td>
</tr>
<tr>
<td>9.5 More Interprocess Communication</td>
<td>63</td>
</tr>
<tr>
<td>9.5.1 Further generalization: message passing</td>
<td>63</td>
</tr>
<tr>
<td>9.6 Classic IPC Problems</td>
<td>65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lecture: Scheduling</th>
<th>66</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1 Announcements</td>
<td>66</td>
</tr>
<tr>
<td>10.2 We talked a lot about LWP</td>
<td>66</td>
</tr>
<tr>
<td>10.2.1 From Last Time: Further generalization: message passing</td>
<td>67</td>
</tr>
<tr>
<td>10.3 Classic IPC Problems</td>
<td>67</td>
</tr>
<tr>
<td>10.4 So what: Scheduling</td>
<td>68</td>
</tr>
<tr>
<td>10.4.1 Process States</td>
<td>68</td>
</tr>
<tr>
<td>10.4.2 Policy vs. Mechanism</td>
<td>68</td>
</tr>
<tr>
<td>10.4.3 Process types</td>
<td>69</td>
</tr>
<tr>
<td>10.4.4 When to schedule</td>
<td>69</td>
</tr>
<tr>
<td>10.4.5 Evaluation Criteria</td>
<td>69</td>
</tr>
<tr>
<td>10.4.6 Non-preemptive shedding: run-to-completion</td>
<td>70</td>
</tr>
<tr>
<td>10.4.7 Preemptive shedding</td>
<td>70</td>
</tr>
</tbody>
</table>
11 Lecture: Scheduling, this time for real

11.1 Announcements ............................................. 71
11.2 Classic IPC Problems ........................................ 71
   11.2.1 From Last time: Process types .......................... 72
   11.2.2 When to schedule .................................... 72
   11.2.3 Evaluation Criteria .................................. 72
   11.2.4 Non-preemptive scheduling: run-to-completion ........ 73
   11.2.5 Preemptive scheduling ................................ 73

12 Lecture: MINIX IO Architecture ............................ 74

12.1 Announcements ............................................. 74
12.2 Aside: Reentrancy ......................................... 74
   12.2.1 From Last time: Process types .......................... 75
   12.2.2 When to schedule .................................... 75
   12.2.3 Evaluation Criteria .................................. 75
   12.2.4 Non-preemptive scheduling: run-to-completion ........ 76
   12.2.5 Preemptive scheduling ................................ 76
12.3 Minix architecture revisited ............................... 76
12.4 Minix 2 Scheduling ........................................ 77
12.5 Minix 3 Scheduling ........................................ 77
   12.5.1 Three-level scheduling ................................ 78
12.6 Scheduling Example ........................................ 78
12.7 Minix IPC .................................................... 78
12.8 Minix structures .......................................... 79
12.9 Where do we go from here? ................................ 79
12.10 Input/Output .............................................. 79
12.11 Devices ..................................................... 79
12.12 Deadlock and its avoidance ............................... 79
   12.12.1 Deadlock Avoidance Methods ......................... 81

13 Lecture: Managing Multiple Resources ..................... 82

13.1 Announcements ............................................. 82
13.2 Where do we go from here? ................................ 82
13.3 Input/Output .............................................. 82
13.4 Devices ..................................................... 83
13.5 Deadlock and its avoidance ............................... 83
   13.5.1 Deadlock Avoidance Methods ......................... 84
13.6 Banker’s Algorithm (Dijkstra, 1965) ....................... 84
   13.6.1 Single Resource ...................................... 85
13.7 Managing Multiple Resources ............................. 85
   13.7.1 Resource Trajectories ................................ 87
13.8 Managing Multiple Resources, cont. ..................... 87
   13.8.1 Multi-way bankers’ ................................... 87
13.9 Wrapping up deadlock avoidance .......................... 88
# List of Figures

1. A Linux `open()` implementation ........................................ 26
2. Minix `open()` implementation ........................................... 27
3. Minix `syscall()` ............................................................. 27
4. Minix `sendrec()` abstracted .............................................. 28
5. The process of a context switch .......................................... 35
6. Possible states for a process .............................................. 38
7. The lowest layer of the operating system .............................. 39
8. Layout of the Minix system ................................................ 41
9. Layout of the Minix system ................................................ 44
10. Stack development (Remember that the real stack is upside-down) 49
11. Peterson's solution for mutual exclusion ............................. 56
12. A producer-consumer implementation with a race condition .... 57
13. A semaphore-based producer-consumer implementation .......... 58
14. A monitor-based solution to the producer-consumer problem .... 60
15. A semaphore-based producer-consumer implementation .......... 62
16. A monitor-based solution to the producer-consumer problem .... 64
17. A message-passing solution to the producer-consumer problem.. 65
18. A message-passing solution to the producer-consumer problem.. 68
19. Possible states for a process .............................................. 68
20. Layout of the Minix system ................................................ 77
21. How a deadlock forms ...................................................... 80
22. How a deadlock forms ...................................................... 83
23. One dimensional example of Dijkstra's Bankers Algorithm ....... 85
24. Another example, but one that doesn't wedge immediately ...... 86
25. Resource trajectory example from Tanenbaum ....................... 87
26. Multi-dimensional Bankers Algorithm ................................ 89
27. Resource trajectory example from Tanenbaum ....................... 91
28. Multi-dimensional Bankers Algorithm ................................ 92
29. Minix IO ........................................................................ 94
30. Minix IO ........................................................................ 105
31. Histogram of scores for the midterm .................................... 115
32. Virtual Memory translation in the MMU (4k pages) ............... 125
33. Virtual Memory translation in the MMU (4k pages) ............... 129
34. Partition table entry .......................................................... 143
35. Anatomy of a disk ............................................................. 143
36. Anatomy of a filesystem ...................................................... 143
37. Minix superblock .............................................................. 144
38. Minix inode ..................................................................... 145
39. Minix directory ent ........................................................... 145
40. useful Minix numbers ....................................................... 146
41. A minix file with 16-byte blocks and 4-byte zone numbers ...... 149
42. Partition table entry .......................................................... 150
43. Anatomy of a disk ............................................................. 151
44. Anatomy of a filesystem ...................................................... 151
45. Minix superblock .............................................................. 152
46. Minix inode ..................................................................... 152
47 Minix directory ent .................................................. 153
48 useful Minix numbers .................................................. 153
49 Partition table entry .................................................... 156
50 Anatomy of a disk ....................................................... 156
51 Anatomy of a filesystem ............................................... 156
52 Minix superblock ....................................................... 157
53 Minix inode ............................................................... 158
54 Minix directory ent ..................................................... 158
55 useful Minix numbers ................................................... 159
56 The (metadata) entries of a Master File Table ..................... 166
57 The NTFS Attributes ................................................... 166
58 Bobby Tables ........................................................... 174

List of Tables

1 Integer registers of the x86 CPU ..................................... 47