The Importance of Safety

CS439: Principles of Computer Systems
February 20, 2019
Bringing It All Together

• Processes
  – Abstraction for protection
  – Define address space

• Threads
  – Share (and communicate) through global and static data, share the heap, each has its own stack and full use of the registers
  – Race conditions may be a problem!

• CPU Schedulers
  – May pre-empt a process or thread at any time

• Ensuring correctness (OR eliminating race conditions and deadlock)
  – Safety and liveness
  – Atomic instructions
  – Synchronization: mutual exclusion, counted resources...
  – Locks, semaphores, monitors, transactions, conservative two-phase locking
  – The Six Commandments of multi-threaded programming
  – Common patterns: Bounded Buffer, Dining Philosophers, Readers/Writers
Today’s Additions

- The Importance of Safety (Therac-25)
- Pemberley!
Therac-25
or The Importance of Safety
What is the Therac-25?

• Linear accelerator
• Used to treat patients ...
## Modes of Operation

<table>
<thead>
<tr>
<th></th>
<th>Beam Energy</th>
<th>Beam Current</th>
<th>Beam Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>For electron therapy</td>
<td>5-25 MeV</td>
<td>low</td>
<td>magnets</td>
</tr>
<tr>
<td>For X-ray therapy, photo mode</td>
<td>25 MeV</td>
<td>high (100x)</td>
<td>flattener</td>
</tr>
<tr>
<td>For field light mode</td>
<td>0</td>
<td>0</td>
<td>none</td>
</tr>
</tbody>
</table>
What Went Wrong?

• Two (major) software problems
• Tons of bad software design/human failures that might have prevented this:
  – False alarms
  – Errors reported by number only and there was no documentation!
  – No clearinghouse for mistakes and company hid failures from other users
  – No end-to-end consistency checks
  – No quality control
  – Don’t trust software---hardware should have prevented this, too
What about more recent disasters?

- We don’t know for sure
- Possibly software lost treatment plan and defaulted to “all leaves open”

- Software should have sensible defaults!
Lessons

• Complex systems fail for complex reasons
• Be tolerant of inputs
• Be strict on outputs

• Assume buggy software and protect against it!
Pemberley!
Summary

• Please Think!
• Safety first!
  – Coarse-grained locking is the easiest to get right, so do that
  – Don’t worry about performance at first
  – In fact, don’t even worry about liveness at first
• Follow the thread coding standards
  – If you don’t, it is wrong!
Announcements

• Exam 1 is next Wednesday at 7p FAC 21
  – If you have a conflict, you should have already told me
    Show up ON TIME
• “Class” on Wednesday is shortened and optional
  – 10:15a-11:45a in GDC 6.302 (probably)
  – Review sessions (driven by your questions!)
  – Any student may attend either section
  – My regular office hours will be canceled
• Sample exam materials and a topics list will be posted Thursday
  – Solutions will be posted on Monday
• Project 1 due next Friday at 5:59p/11:59p