

P3 CK1 (et al.)

Dave Eckhardt  
[de0u@andrew.cmu.edu](mailto:de0u@andrew.cmu.edu)

# A Word on “Design”

Design

≠

# A Word on “Design”

Design

≠

Hitting it with a stick until  
the twitching stops

# A Word on “Design”

- Issue

```
/*caller beware: turns screen blue!*/  
void putbyte(char c)
```

- “Solution”

```
...  
save_screen_color();  
putbyte(b);  
restore_screen_color();  
...
```

# Checkpoint Issue Summary

- What is an OS kernel?
  - VM
  - Context switch
  - Synchronization
- Each one contains *hard stuff*
  - Not just typing!

# Virtual Memory

- Confusion about the hardware
  - You don't get to define your own PT format
  - See text, pp. 309 – 312
- “copy address space to child”
  - This is *not easy*
  - More than one way to do it
    - Think about tradeoffs

# Context Switch

- *Where* do registers get saved?
  - The hardware strongly encourages an approach
  - Look at x86 trap/exception actions
  - Look at your P1 interrupt handlers
- “Create child PCB and kernel stack”
  - This is *not easy*
  - This is a kind of context switch

# Context Switch

- “Copy argv[] to child”
  - Where?
  - How does it get used?



# minclone() vs. fork()

- Some similarity
- Some differences
- *Please don't* “press start on the code photocopier”
- Design is an iterative process
  - (requires iteration)

# Synchronization

- There are real issues here
  - `readline()` vs. `getchar()`
  - `readline()` vs. `readline()`
  - `wait()` vs. `exit()`
  - `deschedule()` vs. `make_runnable()`

# Synchronization

- Please don't try to paper over them
  - “magic mutex dust”
    - *There is no mutex* unless you build it
    - ...which requires thinking about the mutex issues
    - ...which are *different* in a kernel than in user space.
  - “yield() loop considered harmful”
    - Maybe consult Lecture 9 again?
    - Don't need that *interface*, do need those *thoughts*

# Synchronization

- `yield(-1) ≠ deschedule()`
  - Except at a high level of abstraction

# Synchronization

- Scanning a queue of disabled processes is bad
  - ...especially if you are the timer *interrupt handler*
    - Should not be  $O(N)$  in common case!
    - “Scanning all PCBs considered harmful”
- Generally, sleep/wakeup requires thought

# Miscellaneous

- Managing free frames
  - Don't get bogged down on this
  - But do try to think about it for a while
    - There are “wrong” answers
- Defend static buffer size choices

```
char x[64]; /* Why 64?? */
```
- If you are still uncomfortable in assembly...
  - “cc -S” may be your friend

# Checkpoint Issue Summary

- Design *loop*
  - Outline
  - Think
  - Try
  - Repeat
- What is an OS kernel?
  - VM, context switch, synchronization
  - Each requires design & implementation

# Exam Scores

<i>Range</i>	<i>Count</i>	<i>Percent</i>
<b>90 - 100</b>	<b>21</b>	<b>24</b>
<b>80 - 90</b>	<b>36</b>	<b>42</b>
<b>70 - 80</b>	<b>17</b>	<b>20</b>
<b>60 - 70</b>	<b>10</b>	<b>12</b>
<b>&lt; 60</b>	<b>2</b>	<b>2</b>