First half:

- basic program, HW, + OS structure
  user/super stacks

- context switching

- synchronization
  - HW basics - memory ordering and locking
  - interrupt disable
  - lock free data structures - circular buffer
  - Programming support for sync.
    - mutex + - HW mutex + scheduling
    - monitors (semaphores)

- Queuing and Markov models
  - M/M/1 queue \( \lambda, \mu, \sigma = \frac{\lambda}{\mu} \)
  - total wait and service time \( \mu \)
  - markov model balance eq.

- Virtual Memory
  - page table, TLB
    (size and bounds)
  - page fault handling
    crash / demand alloc / COW / demand paging
  - page replacement
    FIFO / LRU / OPT
- clock
- page fault walk-through

→ Virtualization (No problems in the mid-term)

Questions for mid-term:
1. Context switching
2. Marker 1 Queuing
3. Write a monitor
4. Virtual memory.

Marker 1 Queuing

\[ \lambda_1 \]

\[ S_1 \rightarrow S_2 \]

\[ \lambda_2 \]

\[ S_3 \]

Stay in \( S_1 \) for avg. \( \frac{1}{\lambda_1 + \lambda_2} \)

\( \rightarrow \) go to \( S_2 \) queuing \( \frac{\lambda_1}{\lambda_1 + \lambda_2} \)

\( \rightarrow \) go to \( S_2 \) exp \( \frac{\lambda_1}{\lambda_1 + \lambda_2} \)

\( \lambda_1 + \lambda_2 \)

\( \lambda_2 \)

Queue \( \frac{\lambda_1}{\lambda_1 + \lambda_2} \)

Queue \( M/M/1 \)
\[ \omega = \frac{\lambda}{\mu} \]
\[ \overline{N} = \frac{\sigma}{1-\epsilon} \]
\[ \overline{W} = 1 - \frac{\sigma}{1-\epsilon} = \frac{1}{1-\epsilon} \] (or more)

We can cut the model in two separate subgraphs which are only connected between two states. Then you can simplify it and solve for just the transitions across them and the state probabilities, and worry about the detailed within them separately.

**Write a model**

- Precise
- Draw diagrams.

\[ \text{#1} \] Draw a flowchart

\[ \text{#2} \] Check your pseudocode

Each var: where set?
- Cleared!
- Checked!
# 3 Avoid states

if condition

\[ \text{signal} \Rightarrow \text{condition} = \text{false} \]

another scheduled thread

if a get scheduled then

it too will think

in the fast thread.

Courting semaphore using monitor: Semaphore

\[
\begin{align*}
\text{lock:} & \quad n = \text{value} \\
& \quad \text{cond } \leq 0 \\
& \quad \text{wait } (c) \\
\text{unlock:} & \quad \text{if } n < 0 \\
& \quad \text{signal } (c)
\end{align*}
\]