

# Project 3 Roadmap

**Scheduling**  
**Semaphores**

# Multilevel Feedback Scheduling

- `struct Kernel_Thread` in `kthread.h`
  - **current queue:** `int currentReadyQueue`
  - **blocked state:** `bool blocked`
- **there are 4 run queues now:**
  - `s_runQueue[MAX_QUEUE_LEVEL]` in `kthread.c`
- **a thread is initially in `s_runQueue[0]`**
  - except IDLE thread that enters at 3 when switching to MLF mode
- **check the scenarios**
  - `Find_Best()` automatically pick the highest priority thread in a queue
  - `Get_Next_Runnable()` will call `Find_Best()` for each queue until it finds a ready process

# Multilevel Feedback Scheduling

- end of quantum ?
  - if blocked promote
  - if has just run demote (check TODO in `Timer_Interrupt_Handler()` defined in `timer.c`)
- make sure your `Sys_SetSchedulingPolicy()` works before stuff gets complicated
- A thread becomes blocked in `Wait()` .

# Semaphores

- rough guideline
  - add two files `sem.h/sem.c`
  - create a semaphore structure
  - each has a thread queue, count, name, etc
  - P/V semantics
    - P decrease, block if needed
    - V increase, never blocks
    - section 7.4.2 in the textbook, pp 203

# Testing Your Code

- `workload.exe [rr/mys]`
- `ping.exe & pong.exe`

# GetTimeOfDay()

- `Sys_GetTimeOfDay` - **trivial**
  - **(return** `g_numTicks`)