15-410
"...misbehave(7)..."

Project 2
Sep. 22, 2004

Dave Eckhardt

Bruce Maggs

Synchronization

P2 (et seq.) partners

- "Partner Registration Page" on web site
- 48 people have already registered - Thanks!
- If you know, *please register today*
  - This will help people still looking for partners

Good things to talk about

- How many late days?
- Project *schedule* in other classes
  - *Write down* a joint project schedule
- Auditing or pass/fail? Target 410 grade?
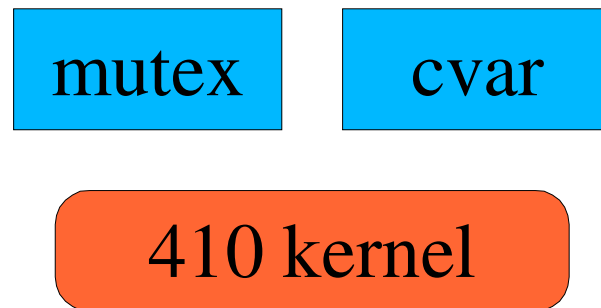- Prior experience

Outline

What you'll build

- Mutex, condition variable
- Thread library
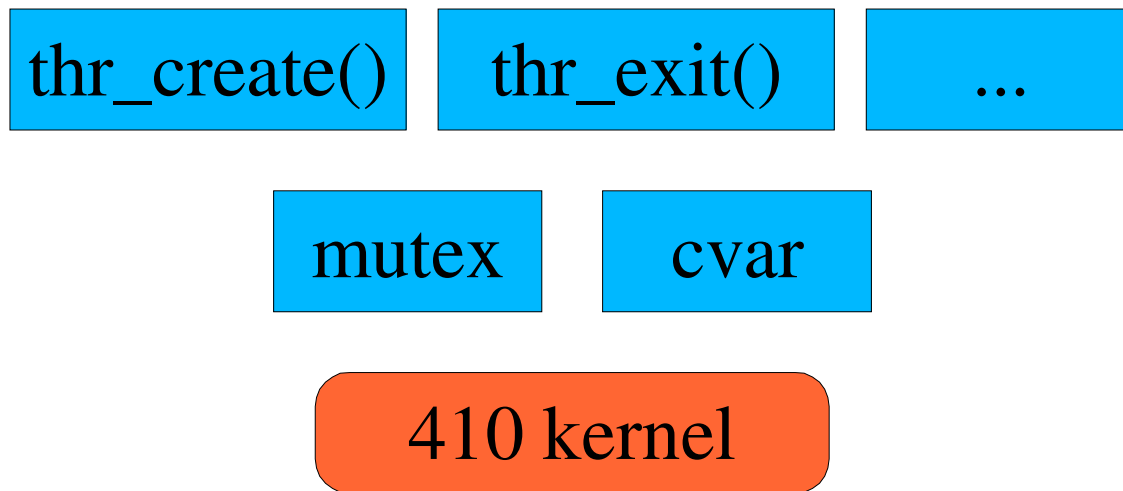- Supplemental library routines
- Tests

How the pieces fit together

- A picture is worth 1000 words
- You'll need to read the handouts too
  - (two, each >1000 words)
  - kspec – specifies our kernel for P2, your kernel for P3
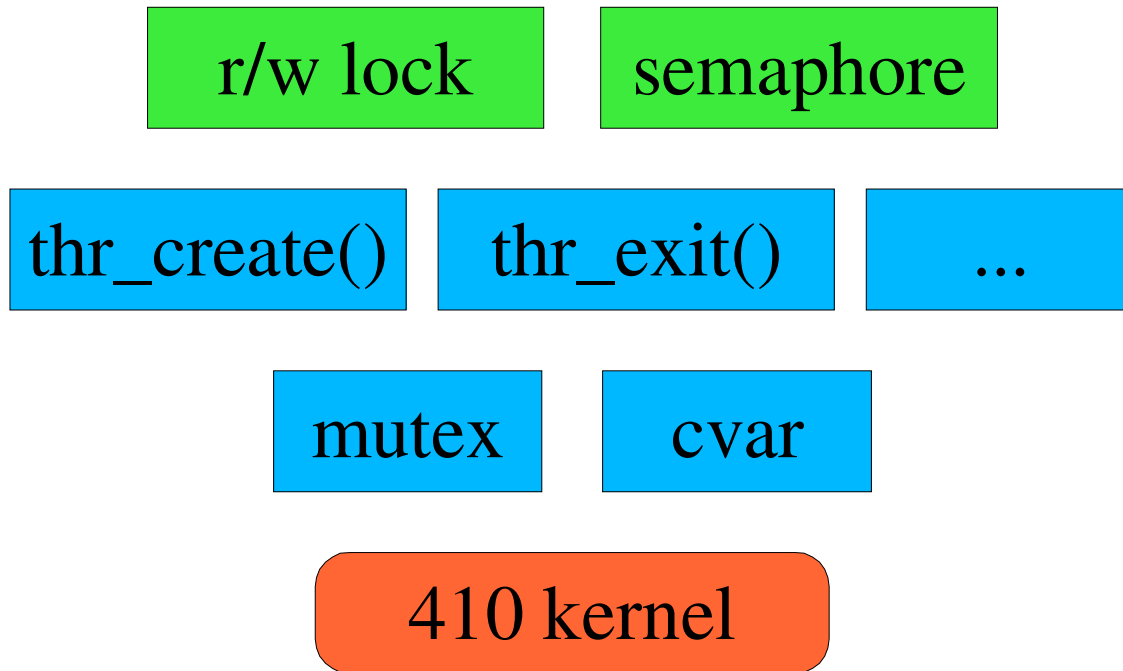 - thr_lib – specifies thread library
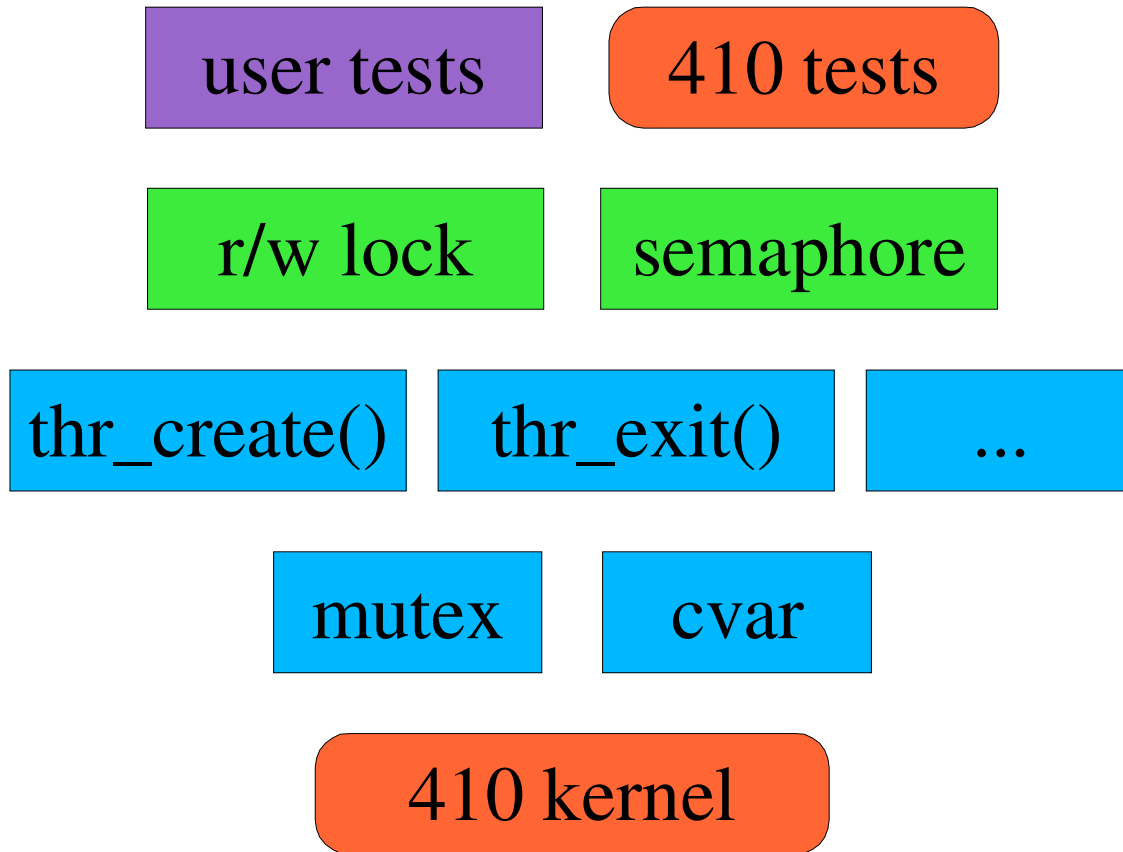
Mutex & Condition Variable

mutex   cvar

410 kernel
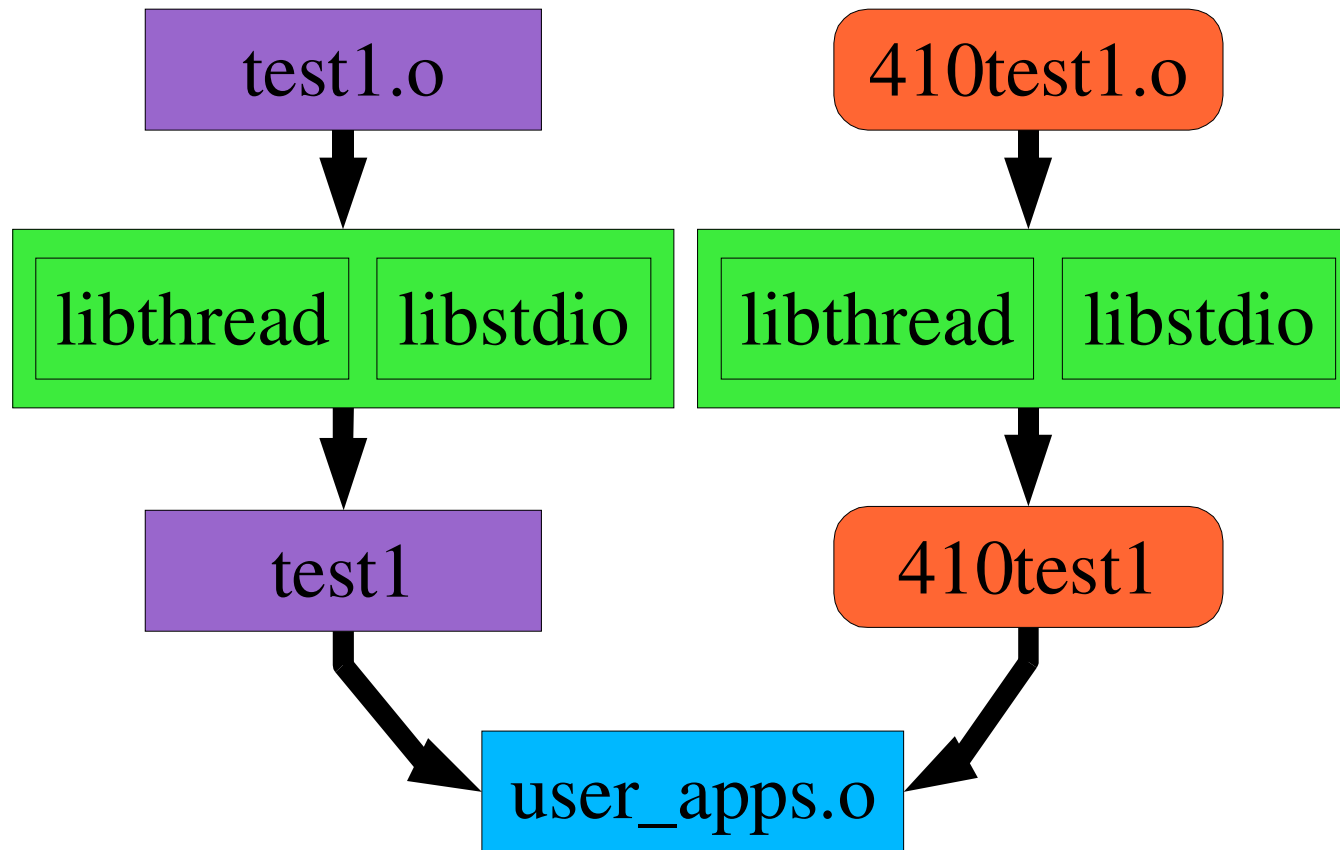
Remainder of Thread Library

thr_create()	thr_exit()	...

mutex	cvar

410 kernel

Supplemental Library Routines

| r/w lock | semaphore |

| thr_create() | thr_exit() | ... |

| mutex | cvar |

410 kernel

Tests (Yours & Ours)

user tests

410 tests

r/w lock

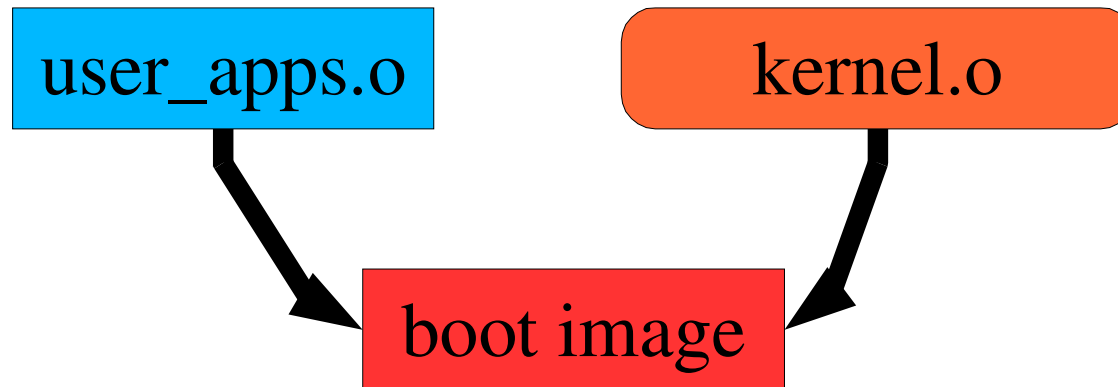semaphore

thr_create()

thr_exit()

...

mutex

cvar

410 kernel

Building a "RAM disk" image

Linking "RAM disk" to kernel

The "Misbehave" System Call

misbehave(int mode)

- Special debugging-support system call in our 410 kernel
- Adjusts "behavior" of system
  - Multiple legal behaviors (you will feel this during P3)
  - Each mode selects a particular mix
    - We will not document these
    - We expect you to not "document" them to classmates either
- Debug your thread library with one mode, then the next...
  - A dazzling array of flavors
    - 0...15
    - maybe even more
    - -1
- You will not be required to implement misbehave() in P3

Plea

This code is *tricky*

- Most of you have already written multi-threaded code
  - That can be tricky enough
- Writing the internals is harder
  - Get a part 99% done
  - Discover a "bug"...
  - ...which is really a misconception
  - *Totally new design* to fix it

Make sure core parts are solid

- Better to skip readers/writers locks if not

The first 90% will take the first 90% of the time

- The last 10% will take the *second 90% of the time*