

Preliminaries

Operating Systems

SAELEE:CS450-S12,#1

Agenda

- prerequisites
- website, textbooks & other class resources
- assignments, exams & grading
- OS topics mindmap
- class overview

§ Prerequisites

Operating system abstractions:

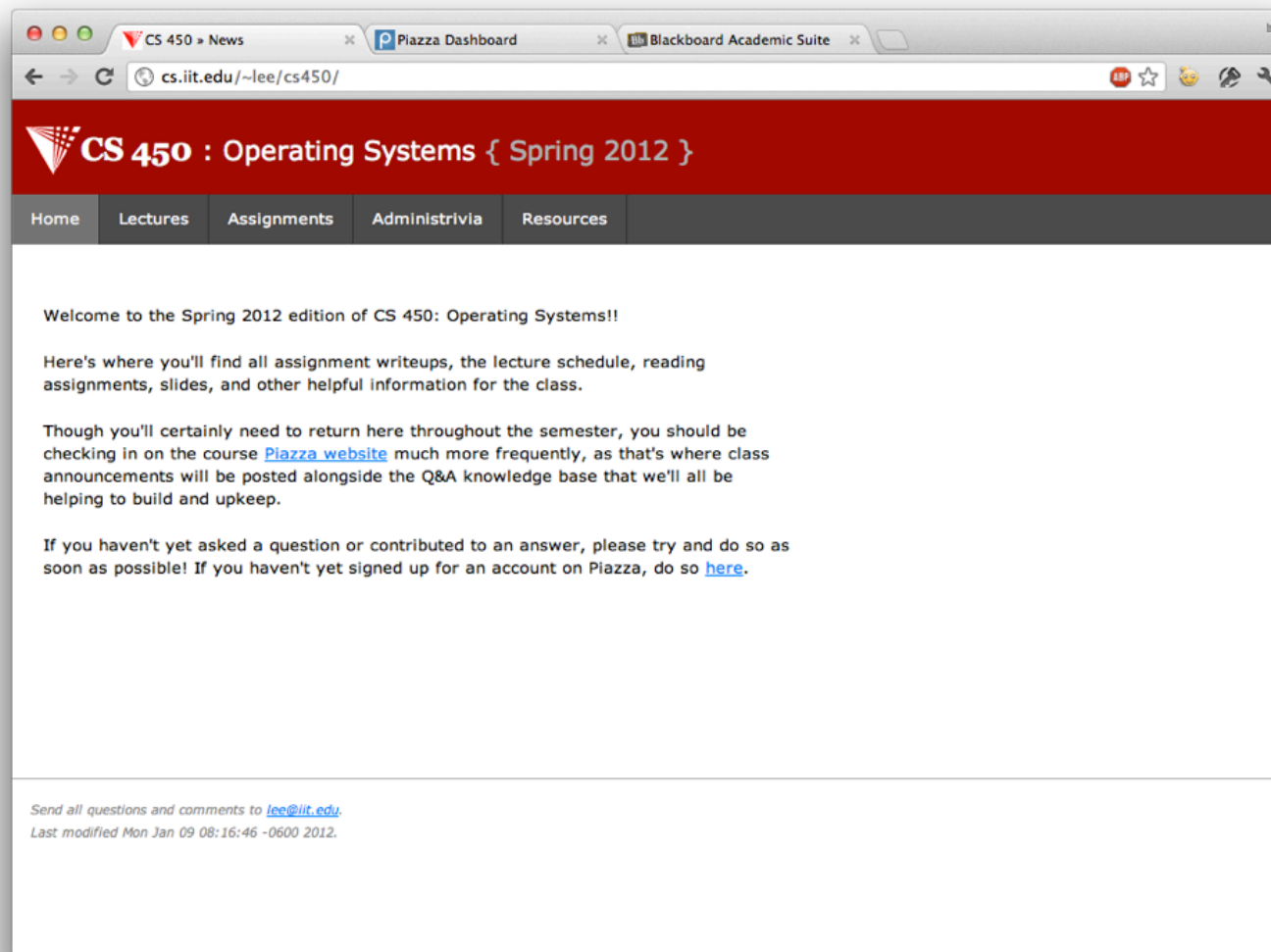
- the process
- concurrency & exceptional control flow
- memory hierarchy (caching)
- virtual memory
- files and I/O structures (e.g., file descriptor)

System level APIs (i.e., syscalls) for:

- process management
- exceptional control flow
- input/output
- interprocess communication

- C
- Some form of assembly (e.g., x86, ARM)
- Stack usage (in procedure call/return)

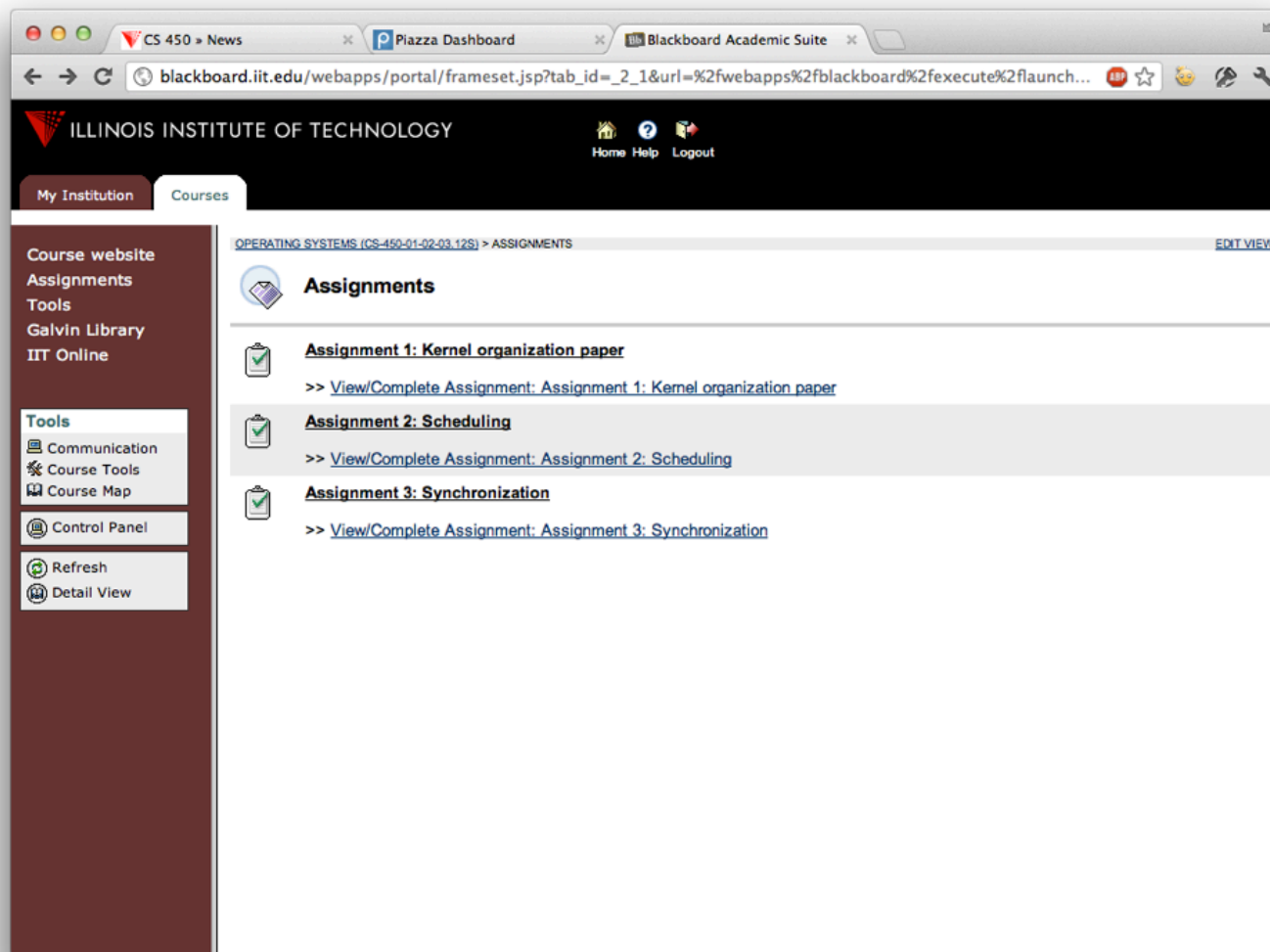
§ Class resources



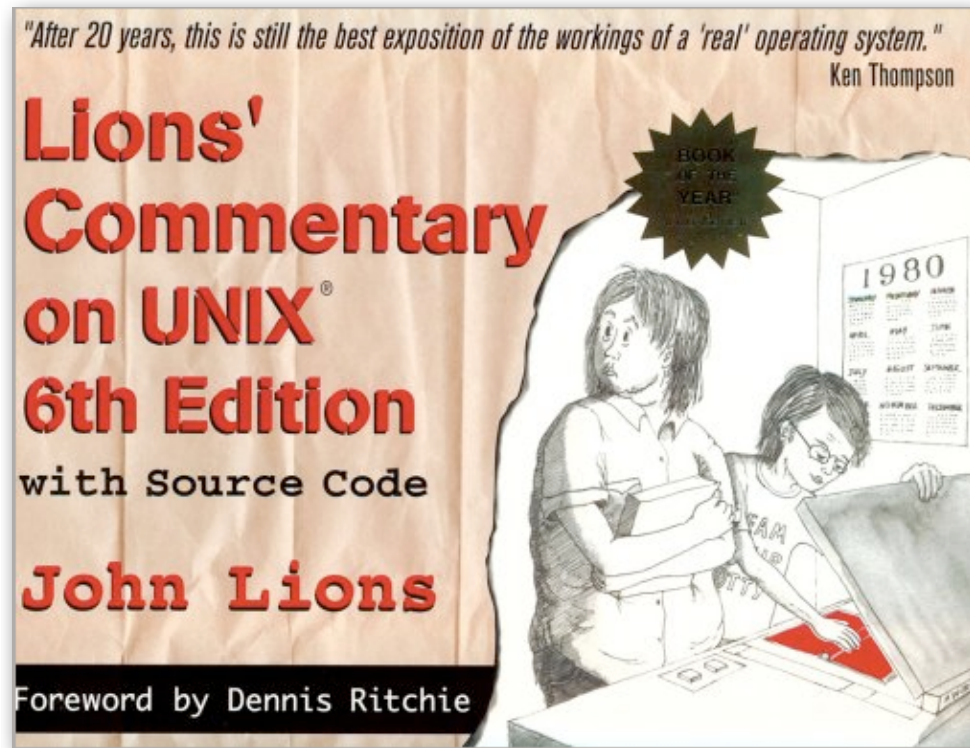
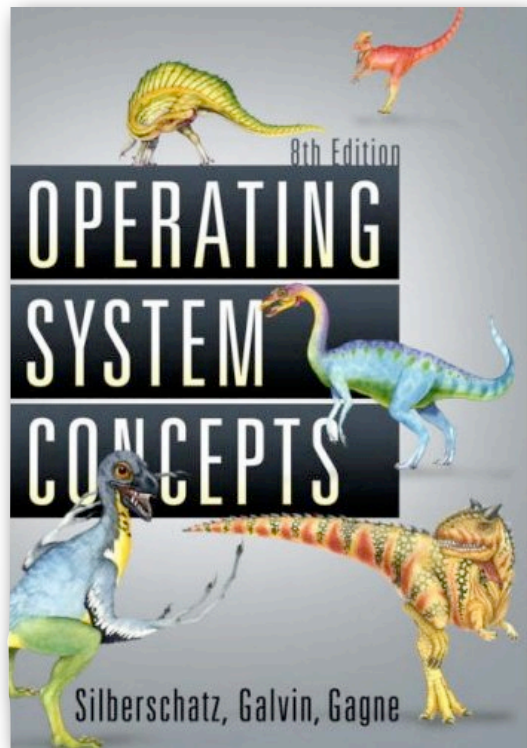
Class website: <http://cs.iit.edu/~lee/cs450>

The screenshot shows a web browser window with the URL piazza.com/class#spring2012/cs450/0. The page header includes the 'piazza' logo, 'CS 450' with a settings icon, a search bar, and a '+ New Post' button. The user profile 'Michael Saelee' is visible in the top right. The main content area features a 'note' titled 'Welcome to Piazza!' with 1 view. The note text reads: 'Piazza is a question-and-answer platform designed to get you great answers from classmates and instructors fast. We put together this list of tips you might find handy as you get started: 1. Ask questions! Ask questions on Piazza rather than emailing your teaching staff so everyone can benefit from a response. 2. Edit questions and answers Wikipedia-style. Every question has a single Students' Response that students can edit collectively (and a single Instructors' Response for instructors). 3. Ask Followups when answers aren't clear. If the Students' and Instructors' Responses aren't clear, ask a Followup below the responses. You can comment on Followups, too, or start a new Followup thread for a different topic. 4. Go anonymous. Shy about asking a question? Select an "Anonymous" option before you post. 5. Tag your posts. Tag your posts so classmates can easily filter questions of the same topic. Type a "#" before a key word to tag.' The footer shows 'Average Response Time: N/A', 'Special Mentions: There are no special mentions at this time.', and 'Online Now: 1 | This Week: 1'. Copyright information for Piazza Technologies, Inc. is at the bottom.

Class Q/A forum: <http://piazza.com>



Blackboard: <http://blackboard.iit.edu>



+ *The Little Book of Semaphores* by A. Downey

Required Textbooks

§ Assignment, Exams & Grading

5-6 assignments — 50% of grade:

- written paper
- quantitative analysis
- simulation
- coding

two exams (midterm & final) @ 25% each:

- final is *not* comprehensive
- *no* curving, scores normalized to 70%
- score $\geq 50\%$ on both exams to pass

A: $\geq 90\%$

B: 80-89%

C: 70-79%

D: 60-69%

E: $< 60\%$

§ OS Topics

§ Class Overview

... the best way to prepare [to be a programmer] is to write programs, and to study great programs that other people have written.

In my case, I went to the garbage cans at the Computer Science Center and fished out listings of their operating system.

- Bill Gates

*Those who do not understand Unix
are condemned to reinvent it, poorly*

- Henry Spencer

"After 20 years, this is still the best exposition of the workings of a 'real' operating system."

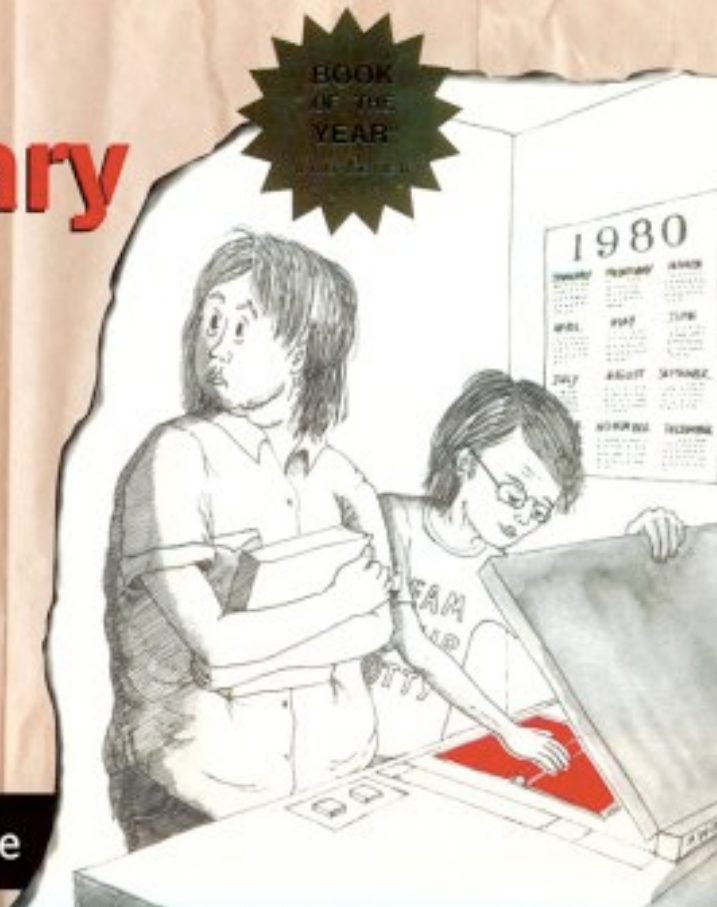
Ken Thompson

Lions' Commentary on UNIX[®] 6th Edition

with Source Code

John Lions

Foreword by Dennis Ritchie



```

2500 #
2501 /*
2502 */
2503
2504 /*
2505 * Structure of the coremap and swapmap
2506 * arrays. Consists of non-zero count
2507 * and base address of that many
2508 * contiguous units.
2509 * (The coremap unit is 64 bytes,
2510 * the swapmap unit is 512 bytes)
2511 * The addresses are increasing and
2512 * the list is terminated with the
2513 * first zero count.
2514 */
2515 struct map
2516 {
2517     char *m_size;
2518     char *m_addr;
2519 };
2520 /* ----- */
2521
2522 /*
2523 * Allocate size units from the given
2524 * map. Return the base of the allocated
2525 * space.
2526 * Algorithm is first fit.
2527 */
2528 malloc(mp, size)
2529 struct map *mp;
2530 {
2531     register int a;
2532     register struct map *bp;
2533
2534     for (bp = mp; bp->m_size; bp++) {
2535         if (bp->m_size >= size) {
2536             a = bp->m_addr;
2537             bp->m_addr += size;
2538             if ((bp->m_size -= size) == 0)
2539                 do {
2540                     bp++;
2541                     (bp-1)->m_addr = bp->m_addr;
2542                 } while((bp-1)->m_size = bp->m_size);
2543             return(a);
2544         }
2545     }
2546     return(0);
2547 }
2548 /*----- */
2549

```

Reproduced under license from the Western Electric Company, NY
Copyright, J. Lions, 1976

```

2550 /*
2551 * Free the previously allocated space aa
2552 * of size units into the specified map.
2553 * Sort aa into map and combine on
2554 * one or both ends if possible.
2555 */
2556 mfree(mp, size, aa)
2557 struct map *mp;
2558 {
2559     register struct map *bp;
2560     register int t;
2561     register int a;
2562
2563     a = aa;
2564     for (bp = mp; bp->m_addr <= a && bp->m_size != 0; bp++)
2565         if (bp->m_size && (bp-1)->m_addr + (bp-1)->m_size == a) {
2566             (bp-1)->m_size += size;
2567             if (a+size == bp->m_addr) {
2568                 (bp-1)->m_size += bp->m_size;
2569                 while (bp->m_size) {
2570                     bp++;
2571                     (bp-1)->m_addr = bp->m_addr;
2572                     (bp-1)->m_size = bp->m_size;
2573                 }
2574             }
2575         } else {
2576             if (a+size == bp->m_addr && bp->m_size) {
2577                 bp->m_addr -= size;
2578                 bp->m_size += size;
2579             } else if (size) do {
2580                 t = bp->m_addr;
2581                 bp->m_addr = a;
2582                 a = t;
2583                 t = bp->m_size;
2584                 bp->m_size = size;
2585                 bp++;
2586             } while (size = t);
2587         }
2588     }
2589 /*----- */
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599

```

Reproduced under license from the Western Electric Company, NY
Copyright, J. Lions, 1976

< 10,000 lines of code

compared to > 10,000,000 for modern kernels

- multi-user

- preemptively multitasked

- a UNIX