Overview of My Research

Contributions
- Multithreaded Issues
- Deterministic Execution
- Memory Security
- Architecture Support

Our Recent Publications
- Memory Security
  - CCS’17, Security’18: secure memory allocator
  - ICSE’16, PLDI’18: detecting memory vulnerabilities, such as overflows, use-after-frees
  - Micro’18: Detecting vulnerabilities using PMU hardware
- Performance of Parallel Programs
  - EuroSys’17: synchronization performance issue
  - CGO’16: false sharing issue
- Reliability of Parallel Programs
  - ASE’17: deadlock detection and prevention

General Information
- Self introduction
  - Research interests: security, reliability and performance of Big Data systems and parallel systems (Software Research Group)
- Class Web
  - Syllabus, class schedule and slides/handouts
- Prerequisites:
  - CS 3733: Operating Systems or equivalent
  - CS 4753: Architecture or equivalent
  - Working knowledge of C/C++/Java
General Information (cont.)

- **Required textbook:**
  - *Operating System Concepts*, by Silberschatz, Galvin and Gagne (SGG), 9th edition (older versions work)

- **Recommend Reading:**

Contact Information

- **Office:** NPB 3.328
- **Office hours:**
  - TuTh: 3:50pm – 5:00pm
  - Or by appointment
- **Email:** Tongping.Liu @ utsa.edu
  - Best way to reach me!
  - Common questions should be posted at Blackboard forum.

Grade Distribution

- **Homework and Programming projects** (30%)
  - Few writing homework (2 or 3)
  - Three projects
  - Discussions are allowed but no code copying/cheating
- **Quizzes** (10%)
- **Two Mid-Term Exams** (15% each)
  - Closed books, closed notes
- **One Final Exam** (30%)
  - Dec 11 (6pm – 8:30pm): fixed date and time
  - Comprehensive, closed books, closed notes
Grading Policy

- Final letter grade:
- Late assignment submission is allowed, but with 10% off each day without prior consent.
- No early/makeup exam without university sanctioned excuse or prior consent.
- Zero tolerance on cheating!!
  - A direct fail on the plagiarism on homework or project.

Course Objectives

- Better understanding of basic OS concepts;
- Learn the principles behind the design of operating systems, both centralized and distributed;
- Discuss on “solved” and “open” problems in OS design and recent OS trends;
- Gain hands-on programming experiences
  - Multithreaded/network programming
  - Distributed system design and implementation

Topics to be covered (and schedule)

- Operating system overview: history, components, design principles and different types of OS
- Process and memory management
- Threads, concurrency and synchronization
- Inter-Process Communication (IPC) and networks
- Distributed/Remote Objects and RMI
- Name and Directory Services
- Security/Protection and Fault Tolerance
- Distributed File Systems
- Advanced OS topics: depending on time

Expectation for OS Course

- Difficult course
  - Significant amount of workload
  - Many abstractions and concepts
  - No much coding examples of the OS implementation (CS4853/CS5463 instead)
- Important course
  - Introducing many important concepts, such as concurrency, scheduling, memory management, replication management
  - Help understand the performance and scalability
  - Teaching the design of computer systems even if you never touch a line of kernel code
Course Design

- The course was designed by the whole OS team at UTSA, which involves the effort from Dr. Dakai Zhu, Tongping Liu, Steve Robins, Lama Palden, and Turgay Korkmaz
- We designed the slides and projects by combining our wisdom together

1st Homework

- Change email on Blackboard to your favorite email:
  - For future important notices etc;
  - Get used to Blackboard system