EECS 750: Advanced Operating Systems

01/21/2015

Heechul Yun
About Instructor

• Assistant Prof., Dept. of EECS, University of Kansas (Aug.’13 ~ )
  – Office: 3040 Eaton, 236 Nichols
  – Email: heechul.yun@ku.edu

• Educations
  – Ph.D. (CS), University of Illinois at Urbana-Champaign
  – M.S. (CS) and B.S (CS), KAIST

• Professional Experiences
  – Senior software engineer, Samsung Electronics, ...

• Research Areas
  – Operating systems, embedded/real-time systems

• More Information
  – http://ittc.ku.edu/~heechul
About This Class

• Lecture: M/W/F 1:00 – 1:50 LEA 2115
• Office hour: M/F 1:50 - 2:50 @ 3040 Eaton
• Advanced topics in OS and system software
• Prerequisite: Undergraduate OS
• No textbook !
  – Optional: Linux Kernel Development (3rd edition) by Robert Rove
• Lecture slides and papers
• Seminar style: I and YOU will present (more on later)
• Course website: [http://ittc.ku.edu/~heechul/courses/eecs750/S15](http://ittc.ku.edu/~heechul/courses/eecs750/S15)
• Audience: Grad students (senior undergraduate) who are interested in research
About This Class

• Goals
  – Learn advanced topics in operating systems
  – Improve your research skills

• Research skills
  – Learning skills
    • Quickly learn from papers, books, and the internet!
  – Communication skills
    • Written form = paper, oral form = presentation
  – Programming skills
    • Need to build some “interesting” things
  – ...
About This Class

• Academic research cycle
  – Read papers
  – Develop “novel” ideas
  – Write a paper
  – Present your ideas at a conference
Operating System

• What are operating systems?

• Provide abstractions
  – Easy to use (hide details), functionality
    • E.g., filesystem, process (virtual memory), ...

• Manage resources
  – Performance, efficiency
    • E.g., CPU scheduling, I/O scheduling, ...
Topics

• Performance
  – How to manage Multicore, cache, DRAM, SSD, and GPU for good throughput/fairness/QoS?

• Power/Energy
  – How to save power/energy while still getting enough performance?

• Reliability
  – How to make system more predictable, less buggy?
  – If you have bugs, how to find them, automatically?
Reading Papers

• We will cover 2~3 papers per week. You need to summarize one of them (of your choice)
• Reading paper well is an important skill
  – A good reference: “How to Read a Paper”
• You are required to email me your summary
  – by 11:59 p.m. the day before the class
  – by 7:00 a.m. is the hard deadline
  – 7:00 a.m. ~ before the class → 50% reduction
  – After the class → zero
Written Summary

• Must be in ASCII text format (no words, pdf)
  – Include “[EECS750]” in the subject line

• Each summary should include:
  – Summary of main ideas
  – What you liked
  – What you disliked
Written Summary: Example

• [Summary] This paper presents a kernel level page allocator which is DRAM Bank-Aware. This allocator is able to allocate pages across cores in a way that causes banks to be shared or partitioned depending on user configuration. This can be used to provide more predictable memory access to multicore software. The authors implemented their memory allocator in a recent version of the Linux Kernel and compared its performance with the existing buddy allocator.

• [The good] This paper is well written. The issue of DRAM banks was not familiar to me at the time of reading but was well explained which motivated the rest of the paper well. The algorithm used is quite straightforward and the explanation is easy to follow.

• [The bad] While the authors acknowledge that the approach they take bears similarity to multi-core page coloring[1,2,3,4] the novelty of their work is not well established. This work appears to be a relatively straightforward application of rudimentary page coloring techniques. The related work section touches on these similarities but does not establish any particular novelty aside from the fact that this paper is addressing the problem of shared DRAM banks for the sake of isolation and not shared caches.
Lecture Organization

• Three phases
  – I will introduce the paper
  – YOU (or I) will present the paper
  – We will discuss the paper

• Each student is required to present two papers per semester
  – May change depending on the class size
Reading List

• Posted on the class website
  – Subject to change
  – Mostly recent papers and some classic ones

• Sign-up process
  – Email me two papers in the list you want to present
  – I will update the schedule on a First Come First Serve basis
Paper Presentation & Discussion

• Suggested structure (30min)
  – Motivation & Background
    • Ask why the authors write this paper?
  – Explain the main ideas
    • From your perspective. Careful about their assumptions
  – Discussion topics
    • Questions: “I don’t understand XXX.”
    • Critiques: “This approach seems bad because ...”

• Send your slides (draft) to me by 5:00 p.m. the day before your presentation
Project

• A team work
  – Ideally two
• A good project
  – A publishable research project
  – A comprehensive evaluation of certain system behaviors
    • E.g., “I want to know when my smartphone’s battery drains fast”
  – A re-implementation of a published paper
    • E.g., “I will re-implement paper XXX on my Linux machine” ⇒ example.
Project Ideas

• Some ideas will be posted on the class website
• Your own idea is the best
  – Feel free to discuss with me
Project Schedule

• Feb. 4: Email me about your group
  – Member names
• Feb. 22: Proposal due
  – 1 page: include what you will build/evaluate
• Apr. 5: Midterm progress report due
  – 3 pages: include intro, related work, progress and plan
• May. 7 (3h): Project presentation
  – No class on Apr. 28 and 30
• May. 15: Final report due
  – 5~8 pages: include results and conclusion
  – a complete paper in two-column IEEE/ACM format
Latex

• Everybody in CS uses it to write papers
• Proposal/midterm/final report should be written with Latex
  – Paper template is on the course website
• Ubuntu
  – $ sudo apt-get install texlive-full
• Window
  – Install MikTex.
  – Google “latex editor”
No Exams !!!
Grading

• Class participation (10%)
• Paper summaries (20%)
• Student presentations (20%)
• Project (50%)
  – Proposal: 5%
  – Midterm report: 10%
  – Final presentation: 15%
  – Final report: 20%
Office Hours

• M/F 4:00 – 5:00 at 3040 Eaton
• By appoint at 236 Nichols
  – heechul.yun@ku.com
Introduce Yourselves

• Name
• Status: grad/undergrad, year
• Relevant background
• Research interests