Spring 2025 – CS Special Topics

1. **CS 594 – Yan – Energy-Efficient Deep Learning**
2. **CS 594 – Tizpaz-Niari – Responsible AI Engineering**
3. **CS 594* – Wang – Advanced Linux Kernel Programming**
4. **CS 594 – Lan – Emerging Trends in Large-Scale Computer Systems**

* CS 594 taught by Xiaoguang Wang in Spring 2025 will be converted to CS 561. So, the class will count as a regular CS 5xx coursework and not as a special topics CS 594 for graduation requirements. Students cannot repeat CS 561 in future semesters for credit.

Back to List

Spring 2025 – CS Special Topics

CS 594 – Energy-Efficient Deep Learning

- Instructor: Yan Yan
- Meeting time: MW 11-12:15pm
- CRN: 33648

Course Description

This course focused on optimizing deep learning and systems for energy efficiency. With deep neural networks requiring substantial computational resources, their deployment on edge devices can be challenging and puts a strain on cloud infrastructure. The course covers advanced AI computing methods designed to enable robust deep learning applications on devices with limited resources. Topics include model compression, pruning, quantization, neural architecture search, and on-device fine-tuning. Additionally, it explores application-specific acceleration techniques for large language models and diffusion models. Students will gain practical experience by implementing model compression strategies and deploying large language models on a laptop.

Course Outline

Week 1: Introduction: Basics of Deep Learning
Week 2: Efficient Inference: Pruning and Sparisity
Week 3: Efficient Inference: Knowledge Distillation
Week 4: Efficient Inference: Quantization
Week 5: Efficient Inference: Neural Architecture Search
Week 6: Domain-Specific Optimization: Transformer and LLM
Week 7: Domain-Specific Optimization: Efficient Transformer Deployment
Week 8: Domain-Specific Optimization: Post Training strategy
Week 9: Domain-Specific Optimization: Post Training on Diffusion Models
Week 10: Domain-Specific Optimization: Post Training on Diffusion Transformer
Week 11: Efficient Training: Distributed Training
Week 12: Efficient Training: On device training and transfer learning
Week 13: Future
Week 14: Final Project Presentation
Week 15: Final Project Presentation

Course Work

The course work will include lectures based on the reading material and student presentations. Students will complete a course project on a topic of their choice.

Course Grading:

- Project and presentation: 50%
- Paper review and presentation: 50%

Book and Readings

- Efficient deep learning, by Gaurav Menghani and Naresh Singh
- Research papers from recent machine learning and computer vision conferences.

Prerequisite
Grade B or better in CS 412 or CS 512.

SPRING 2025 – CS SPECIAL TOPICS

CS 594 – Responsible AI Engineering

- Instructor: Saeid Tizpaz-Niari
- Meeting time: MW 3:30-4:45
- CRN: 48277

Course Goals

The objective of this course is to familiarize students with the state-of-the-art artificial intelligence engineering techniques that incorporate deep neural network (DNN) and Large Language Models (LLMs) as well as their safety, security, and accountability implications. The course first overviews the fundamentals of engineering AI systems that includes requirements, architecture design, validation, and operation of AI-enabled software. Then, the course covers responsible AI-Software development that include topics such as security, privacy, fairness, interpretability, explainability, transparency, and trust, etc. Upon completion of this course, students can answer the following questions:

- How to reliably deploy and update AI models in production? How can we test the entire machine learning pipeline? How can MLOps tools help to automate and scale the deployment process?
- How do we scale production ML systems? How do we design a system to process huge amounts of training data, telemetry data, and user requests? Should we use stream processing, batch processing, lambda architecture, or data lakes?
- How to test, debug, and repair production ML systems? How can we evaluate the quality of a model's predictions in production? How can we test the entire AI-enabled system, not just the model? What lessons can we learn from software testing, automated test case generation, simulation, and continuous integration for testing for production machine learning?
- Which qualities matter beyond a model's prediction accuracy? How can we identify and measure important quality requirements, including learning and inference latency, operating cost, scalability, explainablity, fairness, privacy, robustness, and safety?
- What does it take to build responsible products? How to think about fairness of a production system at the model and system level? How to mitigate safety and security concerns? How can we communicate the reasons of an automated decision or explain uncertainty to users?

Recommended Reading

The course will follow the topics as covered in *Machine Learning in Production: From Models to Products*, by Christian Kästner (2024), *Trustworthy Machine Learning*, by Kush R. Varshney (2022), and *Building Intelligent Systems: A Guide to Machine Learning Engineering*, by Geoff Hulten (2018). In doing so, the course will require reading of academic papers from the top-tier CS venues on these topics.

Prerequisite

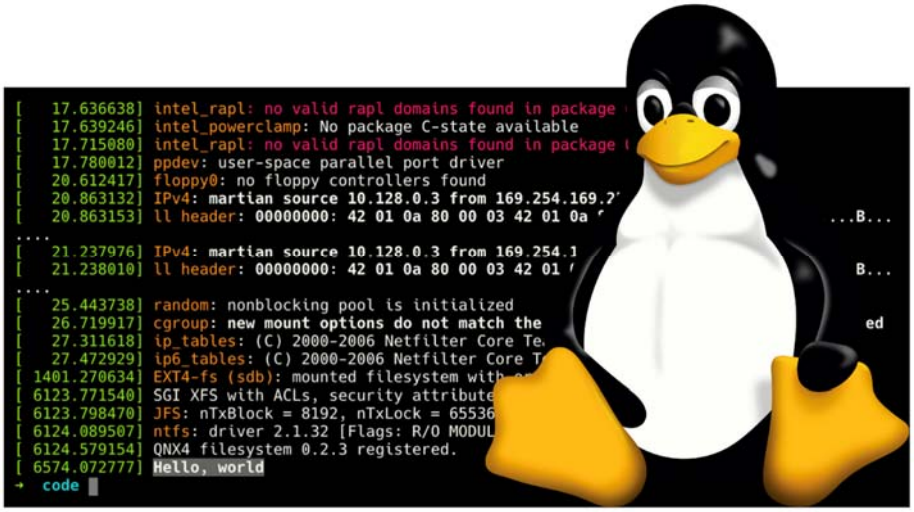
The students are recommended to have some backgrounds on AI/ML courses, e.g., by taking CS 411 (AI 1), CS 412 (Intro to ML), CS 418 (Introduction to Data Science), CS 440 (SE I), or CS 442 (SE II) with a Grade of B or better (or consent of instructor).

Spring 2025 – CS Special Topics

Back to List

CS 594/561 – Advanced Linux Kernel Programming

- Instructor: Xiaoguang Wang - xgwang9@uic.edu
- Meeting time: TR 9:30-10:45am
- CRN: 34724



Course Description:

The Linux kernel is one of the most commonly used and heavily optimized operating system kernels, widely accepted across the industry. It is used on a broad spectrum of computer hardware, from embedded devices to servers, and from portable devices to HPC platforms. Given its versatile design, many industrial systems (e.g., Google Android) and academic systems software research projects rely on Linux. As a result, Linux kernel skills are highly valuable for software engineers - especially those working with systems software - and are also beneficial for hardware engineers who need to test new features or devices.

Students exposed to this course will learn how to program the Linux kernel, implement new or modify existing kernel subsystems, and performance-optimize kernel modules and subsystems by exploiting various time/space tradeoffs and building experience working with a large-scale open-source project. In addition, students will learn the differences between designing, implementing, and debugging application-level and system-level software. These skills are highly desirable for developing operating systems, embedded systems, virtualization infrastructures, and even application software development.

BACK TO LIST

SPRING 2025 – CS SPECIAL TOPICS

Textbook:

Love, Robert. Linux Kernel Development (3rd Edition). Pearson Education, 2010.

Prerequisites:

Systems programming (e.g., CS 361) or equivalent, or consent of the instructor.

Recommended:

Operating Systems Design and Implementation (e.g., CS 461)

Programming Requirements:

A solid understanding of C programming and proficiency with the Linux command line are essential prerequisites for this course. Familiarity with algorithms, data structures, and computer architecture is also recommended to support more advanced concepts and hands-on exercises.

Course Outline and Topics (Tentative):

The course will combine lectures on Linux Kernel programming and reading discussions. Students will also be encouraged to read papers on the state-of-the-art of systems research.

Week	Topics
W01	Intro to Linux kernel Programming, Toolchain for Kernel Development, Isolation, and System Calls
W02	Linux Kernel Data Structures
W03	Kernel Debugging, and Tracing Techniques
W04	Process Management and Process Scheduling
W05	Interrupt Handling: Top Half, Bottom Half
W06	Kernel Synchronization
W07	Timer and Time Management, Device Drivers
W08	Memory Management, Address Space
W09	Virtual File System, Page Cache, and Page Fault
W10	Filesystem and Block IO
W11	Operating System Virtualization
W12	Paper Reading and Discussion
W13	Paper Reading and Discussion
W14	Paper Reading and Discussion
W15	Research Project Presentation

Spring 2025 – CS Special Topics

Course Work:

The course will have 3-4 programming projects and one final project (no final exam). The final project will be structured as a mini research project, including a research proposal, the design and implementation of a prototype, and an evaluation of the prototype's effectiveness.

SPRING 2025 – CS SPECIAL TOPICS

CS 594 – Emerging Trends in Large-Scale Computer Systems

- Instructor: Zhiling Lan – zlan@uic.edu
- Meeting time: TR 2-3:15pm
- CRN: 33792

Course Goals

Large-scale computer systems play a pivotal role in both high-performance com- puting and cloud computing. This class is designed to enable students to keep up with the latest developments in modern computing platforms, with hard- ware and software working in concert to deliver good levels of performance and efficiency. The lectures cover a broad array of topics including heterogeneous architecture, networking, storage, power management, availability and reliability, resource management, and emerging topics such as machine learning for systems. These concepts are reinforced with research paper readings and hands-on projects that involve computer system design and analysis. By the end of this course, students will:

- Gain a deep understanding of the core principles and technologies behind large-scale computer systems that power high-performance computing and cloud computing.
- Develop practical skills in concerted hardware and software development for massive computing.
- Dive into emerging areas of interest such as machine learning applications for system optimization.
- Apply the knowledge to solve real-world problems through hands-on projects and case studies.

Course Materials

A recommended textbook is "Datacenter as a Computer: An Introduction to the Design of Warehouse–scale Machines" by L. Barroso, Jimmy Clidaras, U. Hoelzle (BCH), 3rd edition, Morgan & Claypool Publishers, 2019. A number of research papers taken from premier cloud computing conferences (e.g., USENIX conferences) and high-performance computing conferences (e.g., SC, HPDC, IPDPS, Cluster) will be used for class reading. All the reading material will be provided to students as either PDF files or pointers to online resources.

SPRING 2025 – CS SPECIAL TOPICS

Course Outline (Tentative)

Week	Topic	Readings
Core concepts		
Week 1	Software infrastructure	BCH ch. 1,2
Week 2	Hardware architectures	BCH ch. 3,6
Week 3	Power management	BCH ch. 4,5
Week 4	Availability and reliability	BCH ch. 7
Week 5	Storage and file systems	Research
Week 6	Networking	Research
Week 7	Resource management	Research
Week 8	Monitoring and analysis	Research
Emerging topics		
Week 9	Emerging architectures	Research
Week 10	Predictive analysis	Research
Week 11	ML for resource	Research
Week 12	ML for memory/storage	Research
Week 13	ML for compiler	Research
Week 14	ML for crosscutting	Research
Week 15	LLMs for systems	Research

Course Work

We expect students to have a basic understanding of computer systems, high-performance computing, and machine learning knowledge. The course is a com- bination of lectures and paper reading, including reading, presenting, and a semester-long research project.

Summary

This course focuses exclusively on the **emerging hardware and software trends in large-scale computer systems** deployed for cloud and high-performance computing. Upon completion, students will be well-prepared to tackle the challenges and opportunities presented by modern systems in these domains.