Graduate Operating Systems COP5614

Spring 2023

1

Course Overview

- Instructor:
  - Christian Poellabauer (cpoellab@fiu.edu)
- Course Meetings
  - MW 9:30 – 10:45
  - CASE 135
  - Zoom (live/recordings)
- Office Hours
  - Monday 12-1 & Tuesday 1-2 or **by appointment**
  - Zoom office hours (link on website)
  - Course web site, announcements

2

Course Overview & Goals

- (Operating) systems research
- Reading, reviewing, critiquing research literature
- Conduct (a bit of) operating systems research (including paper writing process)
- Satisfy core requirement & pass the qualifying exam
- Learn about “life as a grad student & researcher”

3

OS Research Literature

- How has the role of the OS evolved?
- What are key principles for OS?
- How did the past influence the present?
- What are current trends and what will the future bring?

4

Reading/Critiquing Papers

- Read lots of papers
- Discuss papers, methodology, problems they address, solutions they propose, etc.
- Determine what makes a good research paper
- Typically discuss 1-2 papers per lecture

5

Papers and Discussions

- Classical/important/recent papers
- Papers that demonstrate excellence in research
- Papers that demonstrate how the field is changing
- Be willing to question the paper
- Be willing to take a position
- Be willing to be wrong
- Understand that there is not always a “right” or “wrong” answer

6

Scientific/Research Papers?

- What is "research"?
  - "**Creative** and **systematic** work undertaken to increase the stock of knowledge" [Wikipedia]
- Engineering vs Research
  - "I want to build a mouse trap"
    - This is **not** research!
  - **Research requires a question!**

7

Engineering vs Research

- "I want to build a mouse trap"
  - "Is my new mouse trap better at trapping mice than a conventional mouse trap?" (Why?)
  - "Are there common traits among the mice that are being captured (and the ones not being captured)?"
  - Are there characteristics (materials) that make better traps?
  - "Does habituation occur and how?"
  - If my mousetrap were invisible, would it be better?
  - How can we build an invisible mousetrap?
  - **Research requires a question!**
  - **Research often requires engineering!**

8

Engineering vs Research

- Engineering helps you answer the question
  - Create a prototype mousetrap
  - Build a framework in which to evaluate the efficacy of mousetraps
  - Designing experiments combines engineering and research
  - Conducting experiments is often engineering
  - Analyzing and interpreting the results is research

9

Rigor & Reproducibility

- **Rigor:** strict application of the scientific method to ensure unbiased and well-controlled experimental design, methodology, analysis, interpretation and reporting of results
- **Reproducibility:** ability of a study or experiment to be reproduced (by somebody else)

10

Diligent Research

- Write down everything
- Understand the data
- Question yourself constantly
- Remind yourself of the question you are asking
- Keep (publish) the data
- Avoid bias
  - Introduction of systematic error
- Be careful (ethical) using & interpreting data
- Discuss your work with others (share data, paper drafts, etc.)
- **Know the literature!**

11

Research

- There are not necessarily any right answers
- No one can tell you with certainty that you are right
- You are never really done
- Understanding (and working with) large systems is difficult

12

Examples of Research Approaches

- Form a hypothesis
- Measure a real system (trace data)
- Instrument existing systems (and measure again)
- Run simulations
- Analytical investigation of collected data
- Micro vs. macro investigations
- Draw conclusions
- Compare results against others' results
- Use results to form new hypotheses

13

Research Papers

- Big idea papers, unifying themes, small ideas with evaluation, measurements, comparisons, retrospective or experience papers, …
- Keep track of important/relevant/good papers in your field (bibliography, bib file, etc.)

14

Research Papers: LaTeX

- High-quality typesetting system
- De facto standard for the communication and publication of scientific documents
- www.overleaf.com

15

Research Papers: bibtex

```
@article{Smith1987,
    AUTHOR = {Smith, Hal},
     TITLE = {Oscillations and multiple steady states in a cyclic gene
model with repression},
   JOURNAL = {J. Math. Biol.},
  FJOURNAL = {Journal of Mathematical Biology},
    VOLUME = {25},
      YEAR = {1987},
    NUMBER = {2},
     PAGES = {169--190},
      ISSN = {0303-6812},
     CODEN = {JMBLAJ},
   MRCLASS = {92A09 (34K15)},
  MRNUMBER = {896432 (89f:92026)},
MRREVIEWER = {S. J. Merrill},
       DOI = {10.1007/BF00276388},
       URL = {http://dx.doi.org/10.1007/BF00276388},
}
```


16

Research Papers: Critiquing

- Is the problem well described/motivated?
- Does the idea make sense?
- Does the paper make a difference?
- What is being measured/proven/demonstrated?
- Are the measurements (experimental setup) meaningful?
- Are the results meaningful?

17

Research Papers: Critiquing

- Summarize paper in a few sentences
- Put papers in categories (e.g., classic, important, useless, …)
- Is the paper well-written?
- What did you learn from the paper?
- How would you have conducted the research?
- Does the paper suggest any future work?

18

Writing a Paper

- **Abstract:** introduce area, state problem, explain approach, summarize conclusions
- **Introduction:** describe problem, importance, approach and contributions, road map
- **Background:** anything reader needs to know
- **Approach/Solution:** what you did
- **Results:** experimental setup, explain expected results, surprising results
- **Related Work:** relate your work to prior efforts
- **Conclusions** (and future work)

19

Cells: A Virtual Mobile Smartphone Architecture

Jeremy Andrus, Christoffer Dall, Alexander Van't Hof, Oren Laadan, and Jason Nieh
{jeremya, cdall, alexvh, orenl, nieh}@cs.columbia.edu
Department of Computer Science
Columbia University

ABSTRACT
Smartphones are increasingly ubiquitous, and many users carry multiple phones to accommodate work, personal, and geographic mobility needs. We present *Cells*, a virtualization architecture for enabling multiple virtual smartphones to run simultaneously on the same physical cellphone in an isolated, secure manner. Cells introduces a usage model of having one foreground virtual phone and multiple background virtual phones. This model enables a new device namespace mechanism and novel device proxies that integrate with lightweight operating system virtualization to multiplex phone hardware across multiple virtual phones while providing native hardware device performance. *Cells* virtual phone features include fully accelerated 3D graphics, complete power management features, and full telephony functionality with separately assignable telephone numbers and caller ID support. We have implemented a prototype of Cells that supports multiple Android virtual phones on the same phone. Our performance results demonstrate that *Cells* imposes only modest runtime and memory overhead, works seamlessly across multiple hardware devices including Google Nexus 1 and Nexus S phones, and transparently runs Android applications at native speed without any modifications.

Categories and Subject Descriptors
C.0 [**Computer Systems Organization**]: General–System architectures; D.4.6 [**Operating Systems**]: Security and Protection; D.4.7 [**Operating Systems**]: Organization and Design; D.4.8 [**Operating Systems**]: Performance; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces–User-centered design; I.3.4 [**Computer Graphics**]: Graphics Utilities–Virtual device interfaces

General Terms
Design, Experimentation, Measurement, Performance, Security

SOSP'11, October 23-26, 2011, Cascais, Portugal.

Keywords
Android, Smartphones, Virtualization

1. INTRODUCTION
The preferred platform for a user's everyday computing needs is shifting from traditional desktop and laptop computers toward mobile smartphone and tablet devices [4]. Smartphones are becoming an increasingly important work tool for professionals who rely on them for telephone, text messaging, email, Web browsing, contact and calendar management, news, and location-specific information. These same functions as well as the ability to play music, movies, and games also make smartphones a useful personal tool. In fact, hundreds of thousands of smartphone applications are available for users to download and try through various online application stores. The ease of downloading new software imposes a risk on users as malicious software can easily access sensitive data with the risk of corrupting it or even leaking it to third parties [35]. For this reason, smartphones given to employees for work use are often locked down resulting in many users having to carry separate work and personal phones. Application developers also carry additional phones for development to avoid having a misbehaving application prototype corrupt their primary phone. Parents sometimes wish they had additional phones when their children use the parent's smartphone for entertainment and end up with unexpected charges due to accidental phone calls or unintended in-app purchases.

Virtual machine (VM) mechanisms have been proposed that enable two separate and isolated instances of a smartphone software stack to run on the same ARM hardware [2, 5, 13, 22]. These approaches require substantial modifications to both user and kernel levels of the software stack. Paravirtualization is used in all cases since ARM is not virtualizable and proposed ARM virtualization extensions are not yet available in hardware. While VMs are useful for desktop and server computers, applying these hardware virtualization techniques to smartphones has two crucial drawbacks. First, smartphones are more resource constrained, and running an entire additional operating system (OS) and user space environment in a VM imposes high overhead and limits the number of instances that can run. Slow system responsiveness is less acceptable on a smartphone than on a desktop computer since smartphones are often used for just a few minutes or even seconds at a time. Second, smartphones incorporate a plethora of devices that applications expect to be able to use, such as GPS, cameras, and GPUs.

20

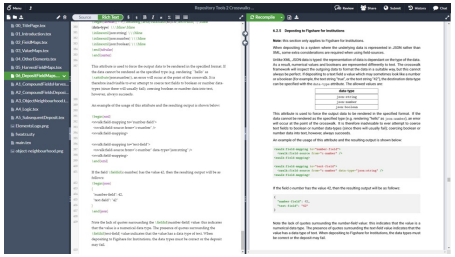
Writing a Paper

- SPELL CHECK!!!!!!!!!!!!!!!
- Learn grammar, style, etc., adapt to your field/advisor/community/…
- Read and critique your own work!!! Are you satisfied? If you know there is a problem, a reviewer will find it too
- Write while you do the work; keep track of all you do; safely store data!

21

Writing a Paper

- LaTeX
  - Recommendation: Overleaf/Sharelatex
- Microsoft Word
  - Recommendation: Google Docs



22

"Entrance Exam"

- On the following slides, you'll find the "entrance exam"
- Try to answer questions on your own by next time (**you do not have to submit anything and this is NOT graded!**)
- Based on how you perform:
  - May want to wait a year and take **ugrad OS** first if you have no background whatsoever in operating systems or a related field
  - **Be willing to learn (some) basic OS concepts in parallel** (will require proactiveness, independence, time, …)

23

"Entrance Exam"

- What is a **multi-threaded** process?
- What is the purpose of **mutual exclusion**?
- What does it mean to say an operation is **atomic**?
- Use a brief example to describe what a **deadlock** is or how it can be caused.
- What is the difference between **deadlock** and **starvation**?

24

"Entrance Exam"

- What is the purpose of an **interrupt**?
- What is **priority inversion**?
- What does a **page table** do?
- What does **thrashing** mean?
- What is a **symbolic link**?
- What is a **parity bit**?
- What is an **i-node** (or **file control block**)?

25

"Entrance Exam"

- What does it mean to **fork** a process?
- What is the danger of **caching a write**?
- What is a **page fault**?
- What is the difference between **kernel space** and **user space**?
- What is **disk fragmentation**?
- What is a **critical section**?

26

"Entrance Exam"

- What is a **runqueue** (or **ready queue**)?
- What is a **binary semaphore**?
- What is the difference between a **direct pointer** and an **indirect pointer** in a file system such as EXT?
- Can you name and very briefly describe a **scheduling algorithm** that would be **fair** to all tasks awaiting execution?

27

"Entrance Exam"

- Can you name and very briefly describe a **scheduling algorithm** that might be a good choice in a **real-time system**?
- What is a **system call**?
- What does it mean for a system call to **block**?

28

Summary

- Course website; “resources” link
- 1-2 papers per lecture
- Typical course structure:
  - Introduction into subject (not each lecture)
  - Discussion of paper(s)
  - Discussion of “grad student life” topics (time permitting)
- First papers: next week Monday
- Start thinking about **annotated bibliography** & **project proposal**

29

Next Lecture & Next Week

- Wednesday:
  - Introduction
  - Revisit ”entrance exam”
- Next 2 weeks (no class Monday!):
  - **OS History and Architecture**
    - **[1]** P. Brinch Hansen, "The Nucleus of a Multiprogramming System", Communications of the ACM, 238-242, April 1970.
    - **[2]** Dennis M. Ritchie and Ken Thompson, "The UNIX Time-Sharing System", Communications of the ACM, volume 17, number 7, July 1974.
    - **[3]** Dawson R. Engler, M. Frans Kaashoek, and James O'Toole Jr., "Exokernel: An Operating System Architecture for Application-Level Resource Management", Proc. of the 15th Symposium on Operating Systems Principles, December 1996.

30