CS 5204
Operating Systems
Fall 2011

Lecture 1: Introduction
Ali R. Butt
The essentials

• Meets T Th 5:00-6:15pm McBryde 329

• Office hours
  – By appointment

• Class website
  http://courses.cs.vt.edu/~cs5204/fall11-butt/
Prerequisites

• Understanding of undergraduate-level operating systems concepts and principles

• Programming skills especially in C/C++

• Some familiarity with *NIX systems
Force-add:
If you are not officially enrolled

• Information needed: Name, student ID

• Email this info to me ASAP!!
  butta@cs.vt.edu
  – Use the words “CS5204 force-add request” in the subject field
About this class:
Graduate level operating systems

• Understand advance operating systems concepts

• Prepare for systems research
  – Read and evaluate research papers
  – Learn from experienced researchers and actual systems

• Build actual systems
  – An open-ended term-long project

• Make presentations
  – Discuss published research papers and your own projects
Reading material

• Primary reading is the assigned research papers
  – Look at the reading list on class website
  – What would you like to present?

• Textbooks (primarily for background)
  – Tanenbaum & van Steen: 
    Distributed Systems: Principles and Paradigms (2nd Ed.)
  – Silberschatz, Galvin, Gagne: 
    Operating Systems Concepts
Course format

• Discussions + lectures

• Two student presentations
  – one for assigned research paper
    (may increase to 2 depending on class size)
  – one for term project

• Project:
  – Term project on unstructured problems
Discussions

- Everybody reads assigned papers before class

- Submit brief evaluation form
  - Proves you’ve read the paper
  - Enables you to contribute to discussion

- Evaluation form submission deadlines:
  - Paper evaluation: start of class
  - Speaker evaluation: end of class
Late policy

• No late submissions will be accepted

• Instead, you have six wildcards:
  – Six dates on which you can skip evaluations without penalty
  – Need not be announced beforehand

• Contact instructor for exceptions in severe circumstances only
Paper evaluation form

• What problem does the paper attack? How does it relate to and improve upon previous work in its domain?

• What are the key contributions of the paper?

• Briefly describe how the paper’s experimental methodology supports the paper’s conclusions

• Write down one question you plan to bring up in the discussion
Your presentation

• Present research as if it were your own
  – Give background if necessary

• Evaluate research from your perspective
  – Add insights, criticism, etc.

• Help lead subsequent discussion
Preparing your presentation

• Guidelines for presentations are posted on class website
  – Strongly recommend you read them

• Every student must meet with instructor to discuss slides.
  – Tentative Time:
    • Thursday 1pm for Tuesday presentation
    • Tuesday 1pm for Thursday presentation
  – You must have your slides ready by that time
Speaker evaluation form:
Getting feedback on your presentation

• Content
  – Did the speaker extract and emphasize the paper’s main contributions?
  – Did the speaker put the presented work in context?

• Form
  – Slides: Were the slides readable and concise?
  – Presentation: Was the presentation understandable and clear?
  – Other comments you wish to provide, if any
Class participation

• Your participation is very important
• Usually proportional to preparation

• Does not mean I will count the number of times you speak in class!!

• That said, lack of participation may lead to a loss of as much as a letter grade
Midterm Exam

• One oral exam

• Tentative midterm date
  – Week after October break

• Covers material from lectures and discussion
Overview of Topics

• Concurrency
  – Race detection in multithreads programs
  – Race detection in the kernel

• Threads & Events
  – The principal of Duality
  – Using threads at scale
  – Using events

• Scheduling
  – Proportional share scheduling
  – QoS-aware scheduling
• Kernel Structures
  – Exokernels
  – Flexible system calls
  – Kernel for massive-core machines
• Virtual Machines
  – Supporting VMs in the OS
  – Understanding XEN
  – Memory management in VMs
• File Systems
  – Modern file systems
  – Journaling and failure recovery
  – Distributed storage systems
• Distributed Systems
  – The concept of time
  – MapReduce
  – End-to-end argument
  – Distributed transactions and notifications
  – Highly available key-value store

• Robustness & Reliability
  – Improving reliability of Oss
  – Highly available cluster services
Term project

• Addresses some unstructured problem

• Milestones
  – Project proposal (9/16)
  – Reports via email/meeting every 3 weeks

• Final presentation/demo
  – To teaching staff during or before final’s week
Programming project

• Done in teams of 3-4 students (larger if project size warrants)

• Many options:
  – Build small distributed system
    • E.g., small P2P system; distributed web cache
    • Distribute existing system
  – Perform experiments
    • E.g., characterize Linux workloads
  – Modify or improve existing system
    • E.g., add failure report facility to Linux

• ... your own idea
Grading (tentative)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>2.5</td>
</tr>
<tr>
<td>Paper evaluations and class Participation</td>
<td>1.0</td>
</tr>
<tr>
<td>Research Paper Presentation</td>
<td>2.0</td>
</tr>
<tr>
<td>Term project</td>
<td>3.5</td>
</tr>
<tr>
<td>Final presentation</td>
<td>1.0</td>
</tr>
<tr>
<td>Total</td>
<td>10</td>
</tr>
</tbody>
</table>
Honor Code

• Will be strictly enforced in this class

• Do not cheat
  – Observe collaboration policy outlined in syllabus

• Do not plagiarize
  – Use proper citations

• Read the policies posted on the website
  – Note reference to “codes of ethics used by professional societies in the United States”

• If in doubt, ask!