CSE 451: Operating Systems
Winter 2013

Prolegomena

Gary Kimura

Introduction

- Instructor
  - A bit about myself
  - Gary Kimura CSE 476
- Sources of Information
  - Lectures
  - Reading
  - Projects / Source code
  - All are important
- Lectures
  - Supplement rather than recapitulate text
  - Lots of historical/developmental info
  - Lots of “why was it done this way” info
  - ASK QUESTIONS!

Class tools

- Primary textbook
  - Anderson and Dahlin, *Operating Systems Principle and Practices*
  - Keep up with the reading.
  - Far better for you to read the chapters BEFORE the class
- Ancillary C and Windows reference books
  - Kernighan and Ritchie, *The C Programming Language*
  - Russinovich and Solomon, *Microsoft Windows Internals*
- Class email list
- Class discussion board

Class projects

- Projects based on Windows Research Kernel (Windows 2003 Server) sources
- 4 projects
  - Two individual projects and two group projects
  - You Will Write Code. You Will Read Lots of Code
  - You are either very familiar with C or will become so quickly
- Form 3-person teams during the first week of class
  - Pick one person from your team to communicate to the TAs the names of the members of your team
  - If this is not accomplished by January 17 you will be randomly assigned to a team

Grading

- Goal is to determine what YOU have learned and can express
  - Scores available via Catalyst
- Grading scale (subject to change)
  - 40% Projects (5%, 5%, 15%, 15%)
  - 20% Friday Quizzes
    - Expect 7-8, dropping the lowest or missed quiz. No make up for missed quizzes. Plan accordingly.
  - 30% Final Exam
  - 10% Participation
- Policies
  - Collaboration vs Cheating
  - Late projects

Course Objectives

- Two views of an OS
  - The OS user's (i.e., application programmer's) view
  - The OS implementer's view
- In this class we will learn:
  - Historical motivations
  - What are the major parts of an O.S.
  - How is the O.S. and each sub-part structured
  - What are the important common interfaces
  - What are the important policies
  - What algorithms are typically used
  - What engineering/practicality tradeoffs were used

Philosophy

- You may not ever build an OS
  - But as a computer scientist or computer engineer you need to understand the foundations
  - Most importantly, operating systems exemplify the sorts of engineering design tradeoffs that you'll need to make throughout your careers – compromises among and within cost, performance, functionality, complexity, schedule …
- A good OS should be easily usable by everyone

IT'S NOT SUPPOSED TO BE ERGONOMIC.
E-mail: BobThaves@aol.com ©2002 Thaves / Dist. by NEA, Inc. www.comics.com
THAVES 9-30