Welcome to CPS 210

• Goals: balance breadth (accessibility of recent system research papers / talks) & depth (details of one implementation -- Linux)
• Graduate level OS knowledge
  – readings, discussions
  – programming projects
• Systems Quals course
  – midterm and final exams
• Gateway to systems research
  – E-track term project

Logistics

/www.cs.duke.edu/education/courses/spring04/cps210/

• Programming projects in Linux -- practical hands-on experience with a “real” OS. Book describing kernel design.
• Readings from the literature -- research topics.
• Background: any undergraduate Introduction to OS textbook – Tanenbaum recommended
• Discussion, in class / collaboration, outside of class.
E- and G- Tracks

E-track project:
• Project of your choice
• Mini-conference during reading period.
• Milestones:
  – March 2 - 1 page proposal.

What is an OS?
Traditional Definitions

• Resource Manager of physical (HW) devices ...
• Abstract machine environment. The OS defines a set of logical resources (objects) and operations on those objects (an interface on the use of those objects).
• Allows sharing of resources. Controls interactions among different users.
What is an OS?
Traditional Definitions

• Birthplace of system design principles: e.g., Separation of Policy and Mechanism.
• Supporting role - to provide services for the target workload, not an end product itself.
• Privileged, protected software - the kernel. Different kind relationship between OS and user code (entry via system calls, interrupts).

What is an OS?
Traditional Definitions

• Resource Manager of physical (HW) devices
  ...
  – CPU (computation cycles)
  – Primary memory
  – Secondary memory devices (disk, tapes)
  – Networks
  – Input devices (keyboard, mouse, camera)
  – Output devices (printers, display, speakers)
What is an OS?
Traditional Definitions

• **Resource Manager** of physical (HW) devices …
  – Working simultaneously (source of ||ism).
  – Shared among tasks.
  – Relative performance, capacity, & cost constantly changing.

What is an OS?
Traditional Definitions

• **Resource Manager** of physical (HW) devices …
• **Abstract machine** environment…
  – Threads or Processes (Fork)
  – Address spaces (Allocate)
  – Files (Open, Close, Read, Write)
  – Messages (Send, Receive)
What is an OS?  
Traditional Definitions

- **Resource Manager** of physical (HW) devices ...
- **Abstract machine** environment...
- Allows **sharing** of resources. Controls interactions among different users.

What is an OS?  
Traditional Definitions

- Birthplace of **system design principles**:  
  - Separation of Policy and Mechanism.  
  - End-to-end argument.  
  - Need-to-know principle.  
  - Cache it!
What is an OS? 
Traditional Definitions

• Birthplace of *system design principles*…

• Supporting role - to provide services for the target workload, not an end product itself.
  – Implications on design (build for the common case of the workload as you know it)
  – Implications on performance evaluation
    • *Everything* the OS does is overhead.
    • Must have a good workload model.

• Privileged, protected software - the *kernel*. Different kind relationship between OS and user code (entry via system calls, interrupts).
  – OS *structure* is always an issue
Trends

- Non-performance goals: *-abilities: adaptability, availability, reliability.
- Use of remote resources (harvesting cycles, memory, storage, power, etc).
- Growth areas: wide area (Internet), clusters, grid, multimedia, mobility, ubiquitous computing.
- Security!
- Challenges / opportunities of HW advances.

Influences in OS Design

Traditional
Influences in OS Design

Traditional

Services & API
Internal Structure
Policies / Mechanisms

Workload
Scientific computations
Database operations
Multi-user

Metrics
Performance as
Bandwidth and Latency.

Hardware Resources
Processor, Memory, Disks, Network,
Keyboard, Display, Multiprocessors

Changing

Services & API
Internal Structure
Policies / Mechanisms

Workload
Productivity applications
Games, Multimedia, Web
Process control
Personal (PDAs), Embedded.
E-Commerce

Metrics
Performance as
Bandwidth and Latency.

Hardware Resources
Processor, Memory, Disks, Network,
Keyboard, Display, Multiprocessors
Influences in OS Design

- **Workload**
  - Productivity applications
  - Games, Multimedia, Web
  - Process control
  - Personal (PDAs), Embedded
  - E-Commerce

- **Services & API**
  - Internal Structure
  - Policies / Mechanisms

- **Metrics**
  - Performance as Bandwidth and Latency.

- **Hardware Resources**
  - Processor, Memory, Disks (?), Wireless & IR,
    - Keyboard(?), Display(?), Mic & Speaker,
    - Motors & Sensors, GPS, Camera, Batteries

- **Changing**

  - Productivity applications
  - Games, Multimedia, Web
  - Process control
  - Personal (PDAs), Embedded
  - E-Commerce

- **Metrics**
  - Accessibility, Reliability, No-futz-ness
  - Energy efficiency, Security

- **Hardware Resources**
  - Processor, Memory, Disks (?), Wireless & IR,
    - Keyboard(?), Display(?), Mic & Speaker,
    - Motors & Sensors, GPS, Camera, Batteries