Welcome to CPS 210

Theme for Spring 2001:
Energy-aware OS for mobile/embedded computing

• Graduate Level Operating Systems
  – readings, discussions, and programming projects
• Systems Quals course
  – midterm and final exams
• Gateway to systems research
  – E-track term project

Logistics

–www.cs.duke.edu/education/courses/spring01/cps210/

• What, no book?
The course will be based on readings from the literature.
Background: any undergraduate introduction to OS textbook.
• Discussion, in class / collaboration, outside of class.

E- and G- Tracks

E-track project:
• Possibility 1:
  SOSP contribution March 26
• Possibility 2: Project of your choice
• Mini-conference during reading period.
• Milestones:
  – February 1 - 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 …
  – 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.

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

- Workload
- Metrics
- Hardware Resources
- Services & API
  - Internal Structure
  - Policies / Mechanisms

Performance as Bandwidth and Latency.

HP Processor, Memory, Disks, Network, Keyboard, Display, Multiprocessors

Scientific computations, Database operations, Multi-user

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

Games, Multimedia, Web, Process control, Personal (PDAs), Embedded

Processor, Memory, Disks (?), Wireless & IR, Keyboard(?), Display(?), Mic & Speaker, Motors & Sensors, GPS, Camera, Batteries
Influences in OS Design

- Workload
  - Productivity applications: Games, Multimedia, Web
  - Process control: Personal (PDAs), Embedded

- Services & API
  - Accessibility, Reliability, No-futz-ness
  - Energy efficiency, Security

- Internal Structure
  - Policies / Mechanisms

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

What is Impact?

Rethink OS Design

- Workload
  - Productivity applications: Games, Multimedia, Web
  - Process control: Personal (PDAs), Embedded

- Services & API
  - Accessibility, Reliability, No-futz-ness
  - Energy efficiency, Security

- Internal Structure
  - Policies / Mechanisms

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

Gee Toto, I don't think we're in Unix* anymore...

*substitute Windows NT, Intel Pentium, Computing...