

## COMPSCI 110 Operating Systems

- Who - Introductions
- How - Policies and Administrative Details
- Why - Objectives and Expectations
- What - Our Topic: Operating Systems

1

## How COMPSCI 110 will work

- It's all explained on the web  
<http://www.cs.duke.edu/education/courses/fall01/cps110/>  
Don't expect handouts regularly
- New feature: Discussion sections
  - Goals: provide opportunity for interaction, questions answered, exploration of details that can't be covered in lecture, problem-solving experiences.
  - Initially, based on problems assigned from textbook

2

## How COMPSCI 110 will work

- Immediate ToDo's:
  - Form project groups - email me
    - carla@cs.duke.edu subject: 110 groups
  - Begin reading textbook:
    - Today's lecture - Chapters 1
    - Next lecture - Review of CPS 104
    - Next big topic, Process Mgt and Concurrency - Chapter 2
  - Fill out and leave "Who's who" questionnaire
  - Take pictures of each other
    - Signup sheet, associating your picture with frame number on camera.
    - Tips for "good" photos.

3

## Objectives/Expectations

- What we want to accomplish *today*.
- What I want you to learn in this class ...
- What you can expect from *me*.
- What I expect from *you*.

4

## What you will learn

- What an OS *does*. What services are provided, what functions are performed, what resources are managed, and what interfaces and abstractions are supported.
- How the OS is *implemented*. How the code is structured. What algorithms are used.
- Techniques, skills, and "systems intuition" (e.g., concurrent programming).
- Peaks at current research topics.

5

## What is an OS?

6

## What is an OS?

- *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.
- Privileged, protected software - the *kernel*. Different kind relationship between OS and user code (entry via system calls, interrupts).

7

## What is an OS?

- 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.
- Not the command interpreter and not a library of utility functions that can be linked into user programs.

8

## HW Resources to be Managed

- CPU (computation cycles)
  - Primary memory
  - Secondary memory devices (disk, tapes)
  - Networks
  - Input devices (keyboard, mouse, camera)
  - Output devices (printers, display, speakers)
- Working simultaneously. Shared among tasks.  
||ism - concurrent demands from all directions.

9

## HW Resources to be Managed

- CPU (computation cycles)
- Primary memory
- Secondary memory devices (disk, tapes)
- Networks - bandwidth for web transactions
- Input devices (keyboard, mouse, camera)
- Output devices (printers, display, speakers)
- Energy / battery -life (for mobile devices)

10

## Examples of Abstractions

- Threads or Processes (Fork)
- Address spaces (Allocate)
- Files (Open, Close, Read, Write)
- Messages (Send, Receive)

11

## Main Issues in OS

- Structure
- Concurrency and Synchronization
- Extensibility, Compatibility
- Communication
- Sharing
- Naming
- Performance
- Protection, Access control, Security
- Reliability, Fault Tolerance
- Persistence, Longevity
- Scalability, Distribution
- Accounting - \$\$

12