

Fundamental OS Issues

Structure

How is an operating system organized?

Sharing

How are resources shared among users?

Naming

How are resources named (by users and programs)?

Protection

How are users/programs protected from each other?

Security

How can information access/flow be restricted?

Communication

How to exchange data?

Reliability and fault tolerance

How to mask failures?

Extensibility

How to add new features?

Concurrency

How to control parallel activities?

Performance

How to make efficient use of resources, reduce OS overhead?

Scale and growth

How to handle increased demand?

Compatibility

Can we ever do anything new?

Distribution

How to coordinate remote operations?

Accountability

How to change for/restrict use of resources?

The **principles** in this course are the design **methods, approaches, and solutions** to these issues

What is an operating system?

The OS is the software layer between user applications and the hardware

The OS and Hardware

The OS *abstracts/controls/mediates* access to hardware resources:

Computation (CPUs)

Volatile storage (memory) and persistent storage (disk, etc.)

Communication (network, modem, etc.)

Input/output devices (keyboard, display, printer, camera, etc.)

The OS defines a set of logical resources (*objects*) and a set of well-defined operations on those objects (*interfaces*):

Physical resources (CPU and memory)

Logical resources (files, programs, names)

Benefits to applications:

Simpler (no tweaking device registers)

Device independent (all network cards look the same)

Portable (across Windows95/98/ME/NT/2000/XP/VISTA/...)

Transportable (same program across different OSes (Java))

