Operating Systems
(Honor Track)

Lecture 2: Introduction to OS

Yao Guo (郭耀)

Peking University

Fall 2021

Acknowledgements: Prof. Xiangqun Chen & Tao Wang at PKU and Prof. Yuanyuan Zhou at UCSD

Buzz Words

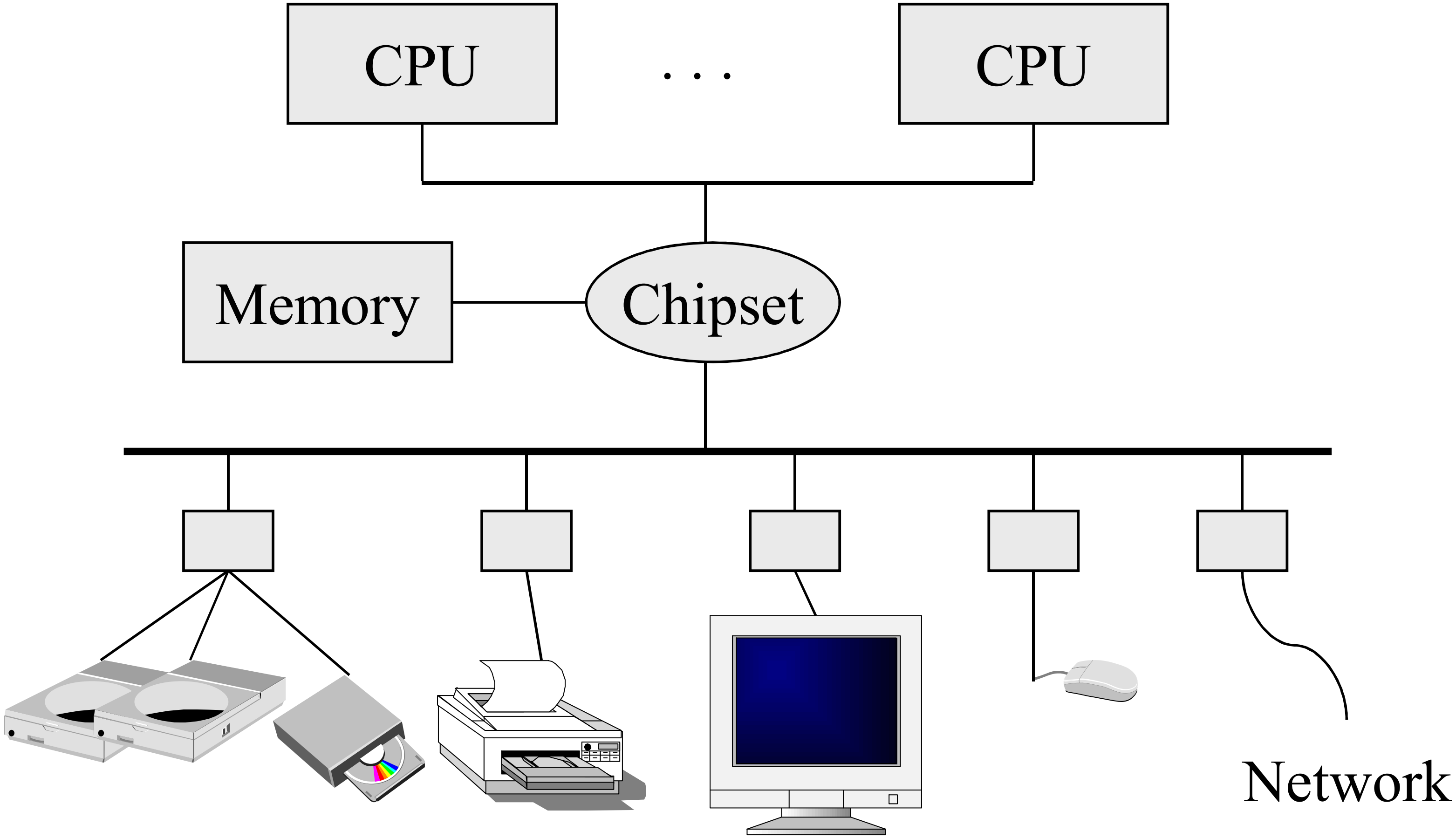
Allocation

Protection

Reclamation

Virtualization

A Typical Computer from a Hardware Point of View



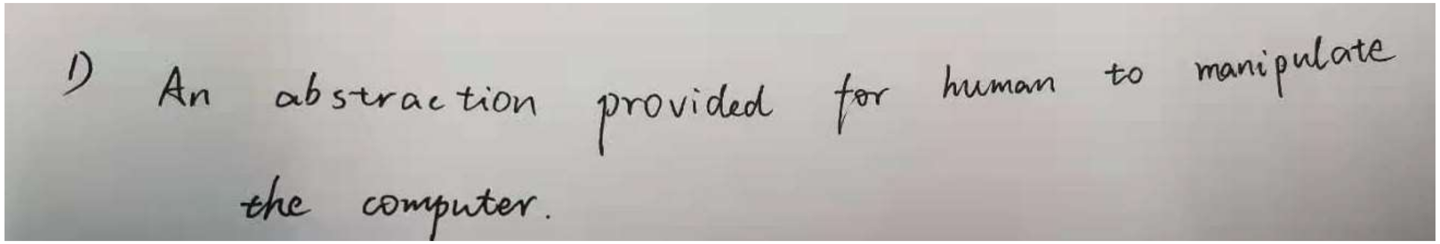
Why Do We Need an OS?

A modern computer consists of:

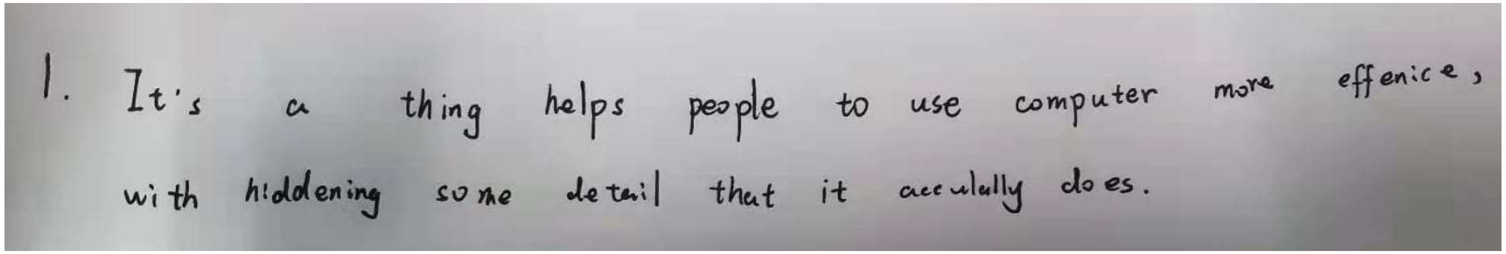
- One or more processors
- Main memory
- Disks
- Screen
- Printers
- Various input/output devices

Managing all these components requires a layer of software – the **operating system**

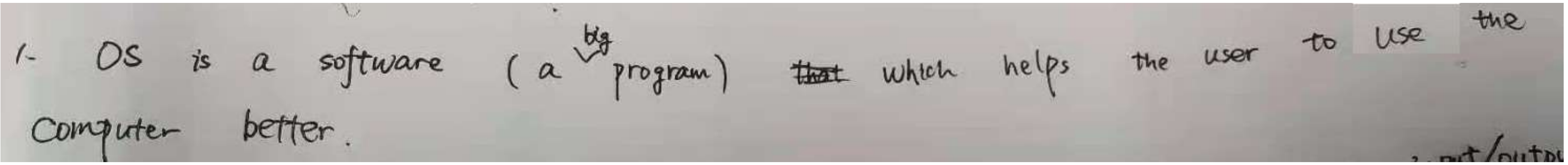
What is an OS? : Your opinions



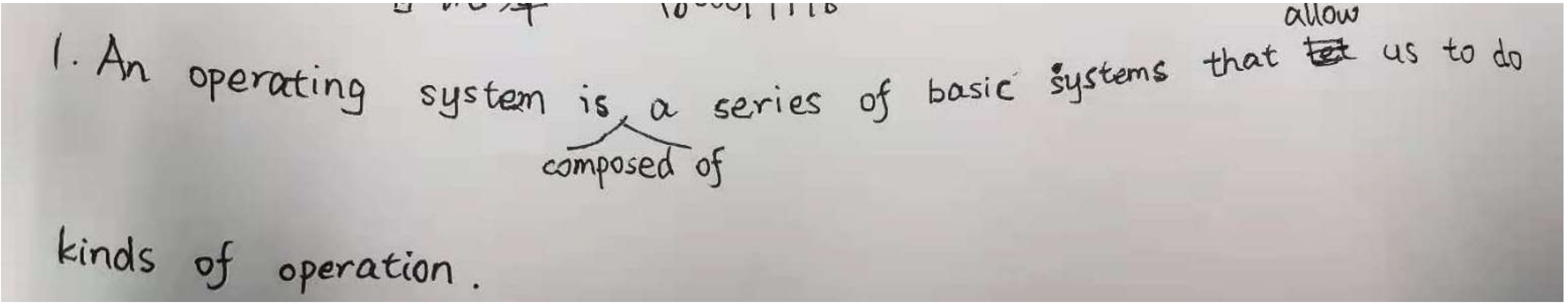
1) An abstraction provided for human to manipulate the computer.



1. It's a thing helps people to use computer more effenice, with hiddening some detail that it acculally does.

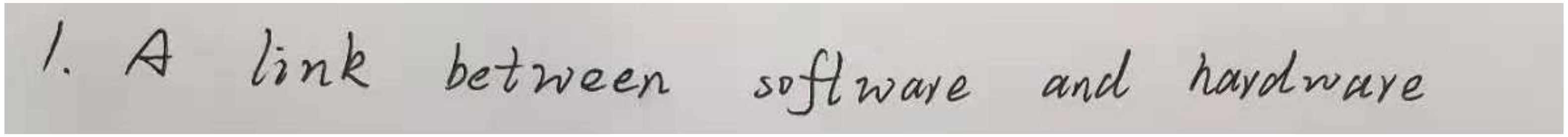


1. OS is a software (a big program) ~~that~~ which helps the user to use the computer better.

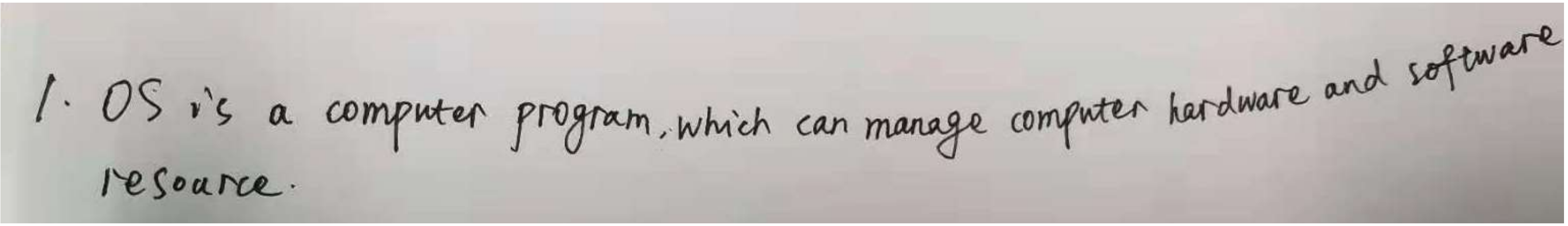


1. An operating system is composed of a series of basic systems that ~~let~~ allow us to do kinds of operation.

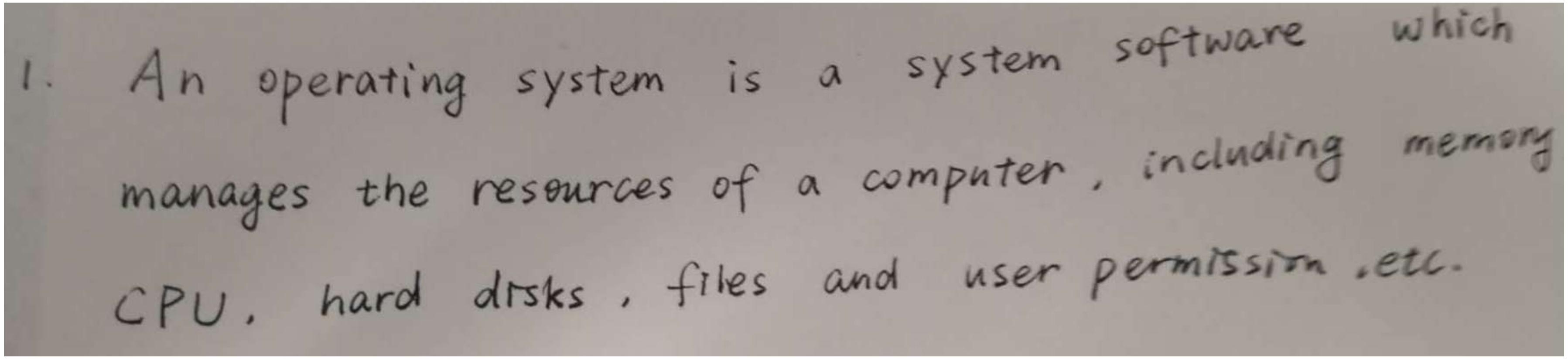
What is an OS? : Your opinions



1. A link between software and hardware

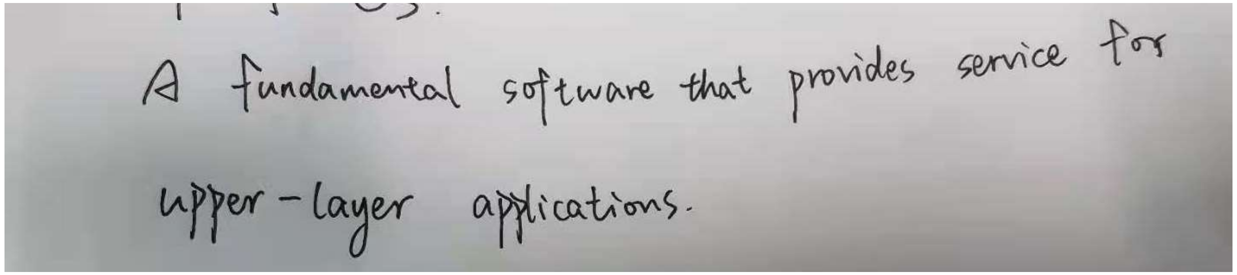


1. OS is a computer program, which can manage computer hardware and software resource.

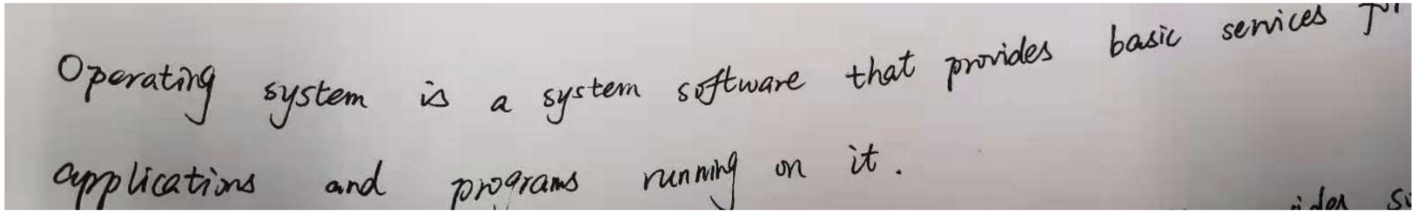


1. An operating system is a system software which manages the resources of a computer, including memory CPU, hard disks, files and user permission, etc.

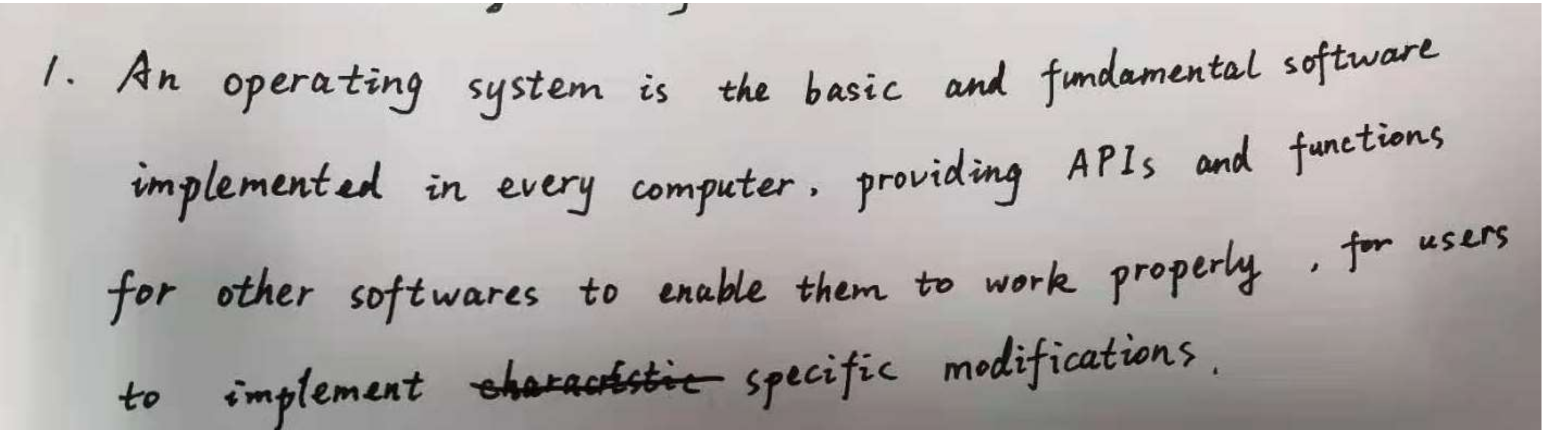
What is an OS? : Your opinions



A fundamental software that provides service for upper-layer applications.



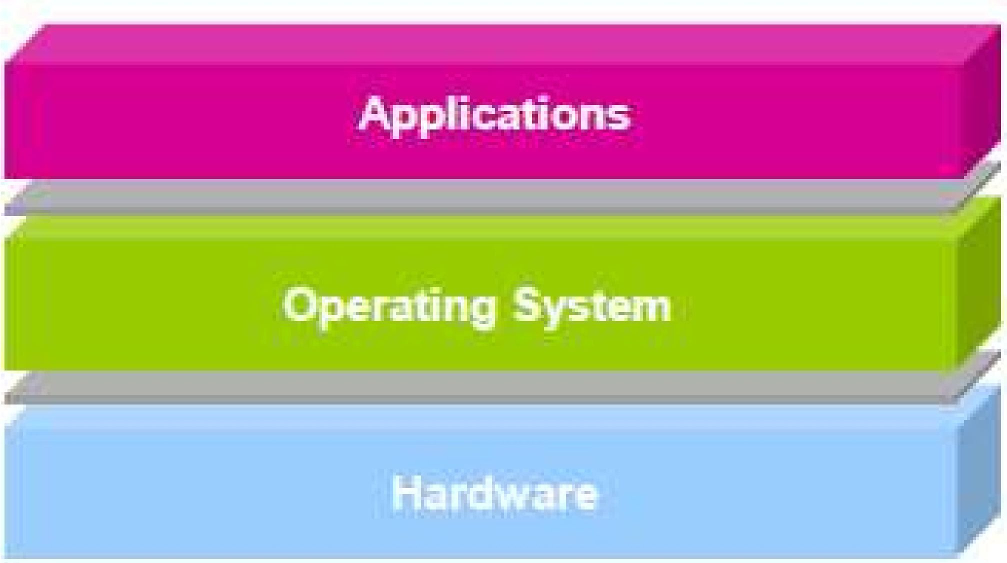
Operating system is a system software that provides basic services f applications and programs running on it .



1. An operating system is the basic and fundamental software implemented in every computer , providing APIs and functions for other softwares to enable them to work properly , for users to implement ~~characteristic~~ specific modifications .

OS Is…

- □ The operating system is the software layer between user applications and the hardware



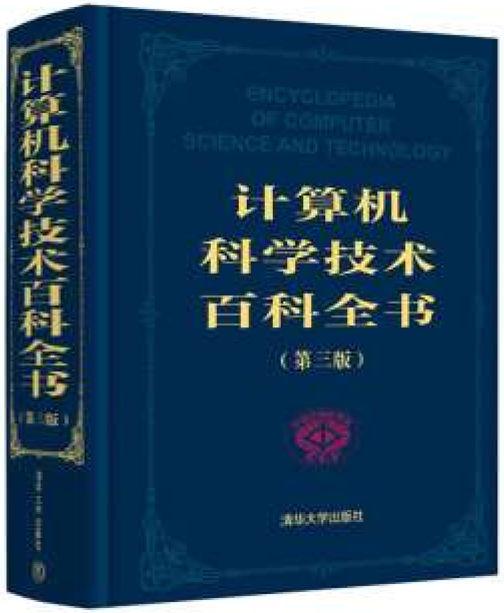
- □ The OS is “all the code that you didn’t have to write” to implement your applications

Some (somewhat) official definitions

- **Wikipedia**: An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.
  - Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.
- **Tanenbaum (OS Concept textbook):** operating system, whose job is to provide user programs with a better, simpler, cleaner, model of the computer and to handle managing all the resources just mentioned.

A more official definition

- 操作系统是管理系统资源、控制程序执行、改善人机交互，为其它软件系统提供支持的一种系统软件，是软件系统的核心。
  - 计算机科学技术百科全书(第3版)



What Is an OS?

"Code" that:

- □ Sits between programs & hardware
- □ Sits between different programs
- □ Sits between different users

But what does it do?

Role 1: as a resource manager

- □ Provides an orderly and controlled allocation of the processors, memories and I/O devices among the various programs competing for them

Real life analogy:

- ■ Government?

As a Resource Manager

Resources

□ Allocation

□ Protection

□ Reclamation

□ Virtualization

As a Resource Manager

Resources

- □ Allocation
- □ Protection
- □ Reclamation
- □ Virtualization

Finite resources

Competing demands

Examples:

- □ **CPU**
- □ **Memory**
- □ **Disk**
- □ **Network**

Government

Limited budget,

Land,

Oil,

Gas,

…

As a Resource Manager

Resources

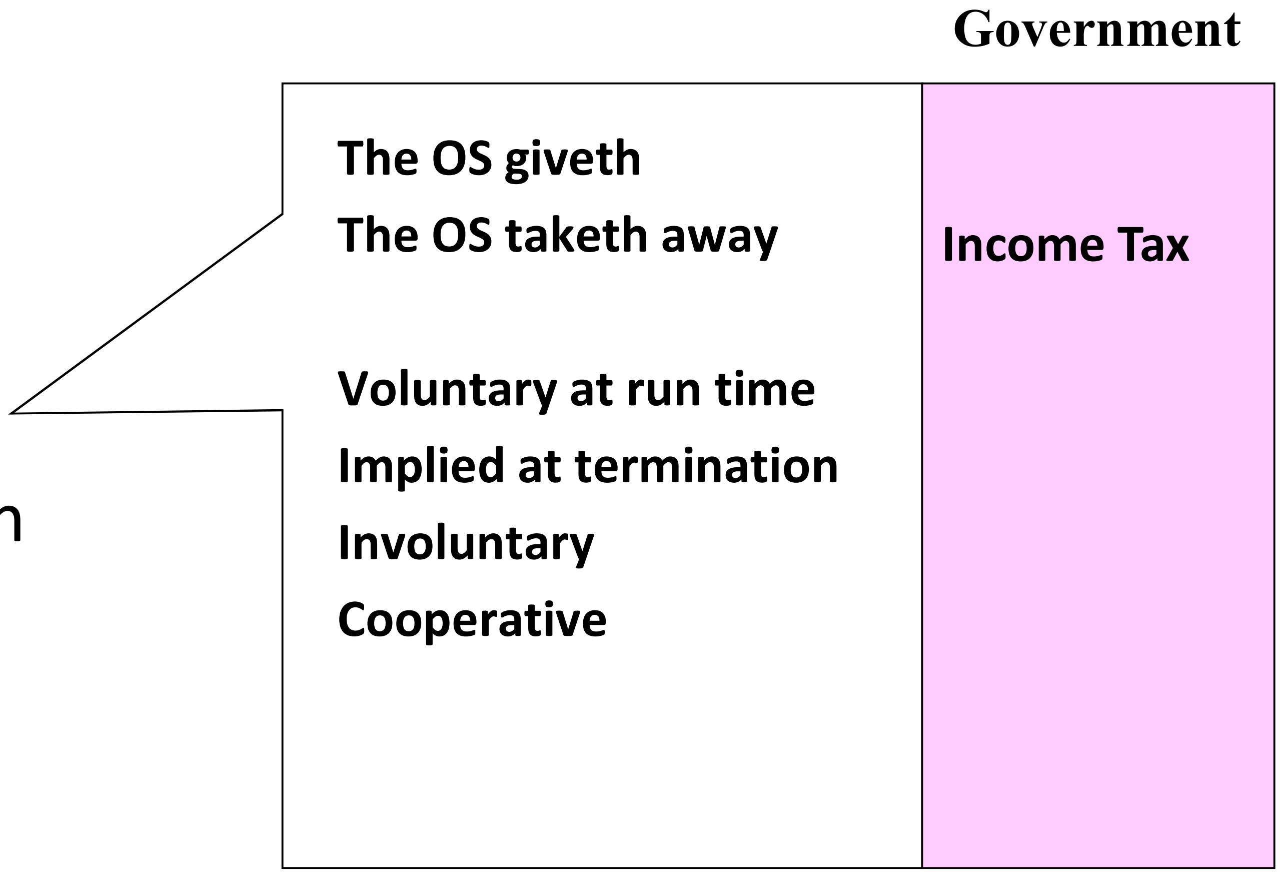
- □ Allocation
- □ Protection
- □ Reclamation
- □ Virtualization

	Government
You can't hurt me **I can't hurt you** **Implies some degree of safety & security**	**Law and order**

As a Resource Manager

Resources

- □ Allocation
- □ Protection
- □ Reclamation
- □ Virtualization



Government

The OS giveth

The OS taketh away

Voluntary at run time

Implied at termination

Involuntary

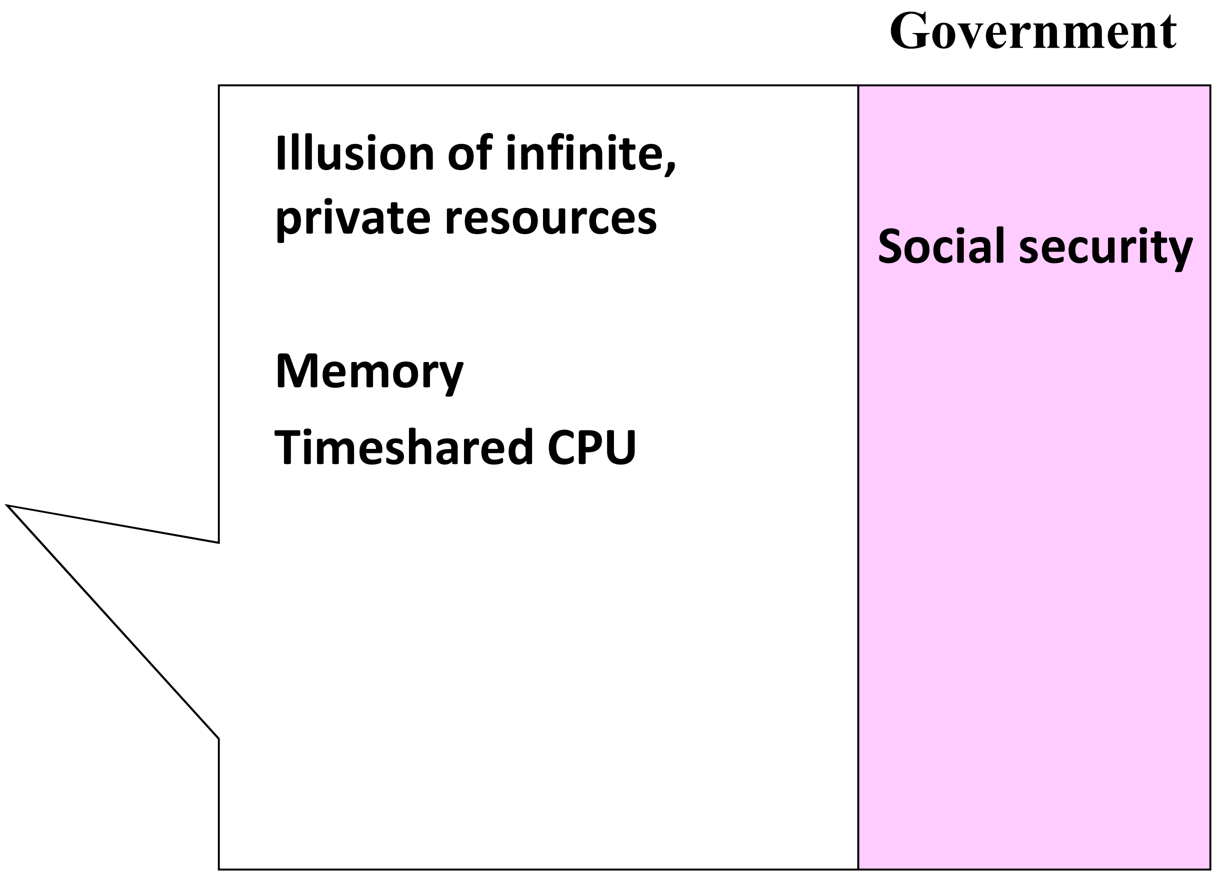
Cooperative

Income Tax

As a Resource Manager

Resources

- □ Allocation
- □ Protection
- □ Reclamation
- □ Virtualization



Government

Illusion of infinite, private resources

Memory

Timeshared CPU

Social security

And Role 2: as an Extended Machine

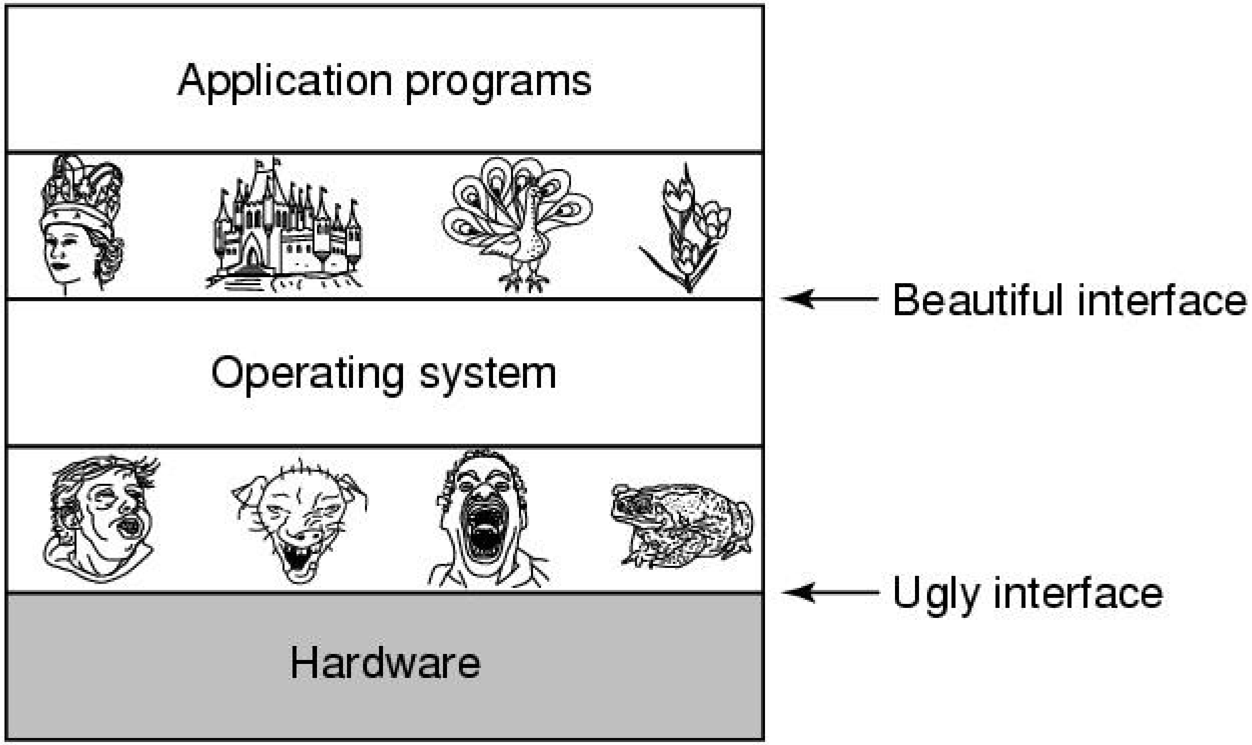


Figure 1-2. Operating systems turn ugly hardware into beautiful abstractions.

The OS and Hardware

- □ The OS abstracts/controls/mediates access to hardware resources (what 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)

The OS and Hardware

- 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/ Windows 7/Windows 8/Windows 10
  - Transportable
    - same program across different OSes(Java)

The OS and Applications

- The OS defines a logical, well-defined environment…
  - Virtual machine (each program thinks it owns the computer)
- …for users and programs to safely coexist, cooperate, share resources
  - Concurrent execution of multiple programs (timeslicing)
  - Communication among multiple programs (pipes, cut & paste)
  - Shared implementations of common facilities
    - No need to implement the file system more than once
  - Mechanisms and policies to manage/share/protect resources
    - File permissions (mechanism) and groups (policies)

What Does an OS Do?

Resources

- □ Allocation
- □ Protection
- □ Reclamation
- □ Virtualization

Services

- □ Abstraction
- □ Simplification
- □ Convenience
- □ Standardization

An OS makes computers simpler!

Fundamental OS Issues

- The fundamental issues/questions in this course are:
  - 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?

Fundamental OS Issues (2)

- 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?
- Scalability: how to handle increased demand?
- Compatibility: can we ever do anything new?
- Distribution: how to coordinate remote operations?
- Accountability: how to charge for/restrict use of resources?

Some Questions to Ponder

- What is part of an OS? What is not?
  - Is the window system part of an OS?
  - Is the Web browser part of an OS?
- Popular OSs today are Windows, Linux, and OS X
  - How different/similar do you think these OSs are?
  - What do you need to do to answer the question?
- OSs change all of the time
  - Consider the series of releases of Windows, Linux, OS X…
  - What is driving the OS evolution?
- What are the most compelling issues facing OSs today?

More Questions to Ponder

- How many lines of code in an OS?
  - Win7 (2009): 40M
  - OS X (2006): 86M
  - Linux (2011): 15M
  - What is the largest kernel component?
- What does this mean (for you)?
  - OSs are useful for learning about software complexity
  - OS is just one example of many complex software systems
    - Chrome (2015): 17M
    - Apache (2015): 1.7M
    - JDK (2015): 6M
    - Unreal Engine 3: 2M
  - If you become a developer, you will face complexity

Summary

- What is an OS
  - As a resource manager
  - As an extended machine
  - Resources: allocation, protection, reclamation, virtualization
  - Services: abstraction, simplification, convenience, standardization

- Next Lecture
  - OS History