PRINCIPLES OF OPERATING SYSTEMS

OPERATING SYSTEM

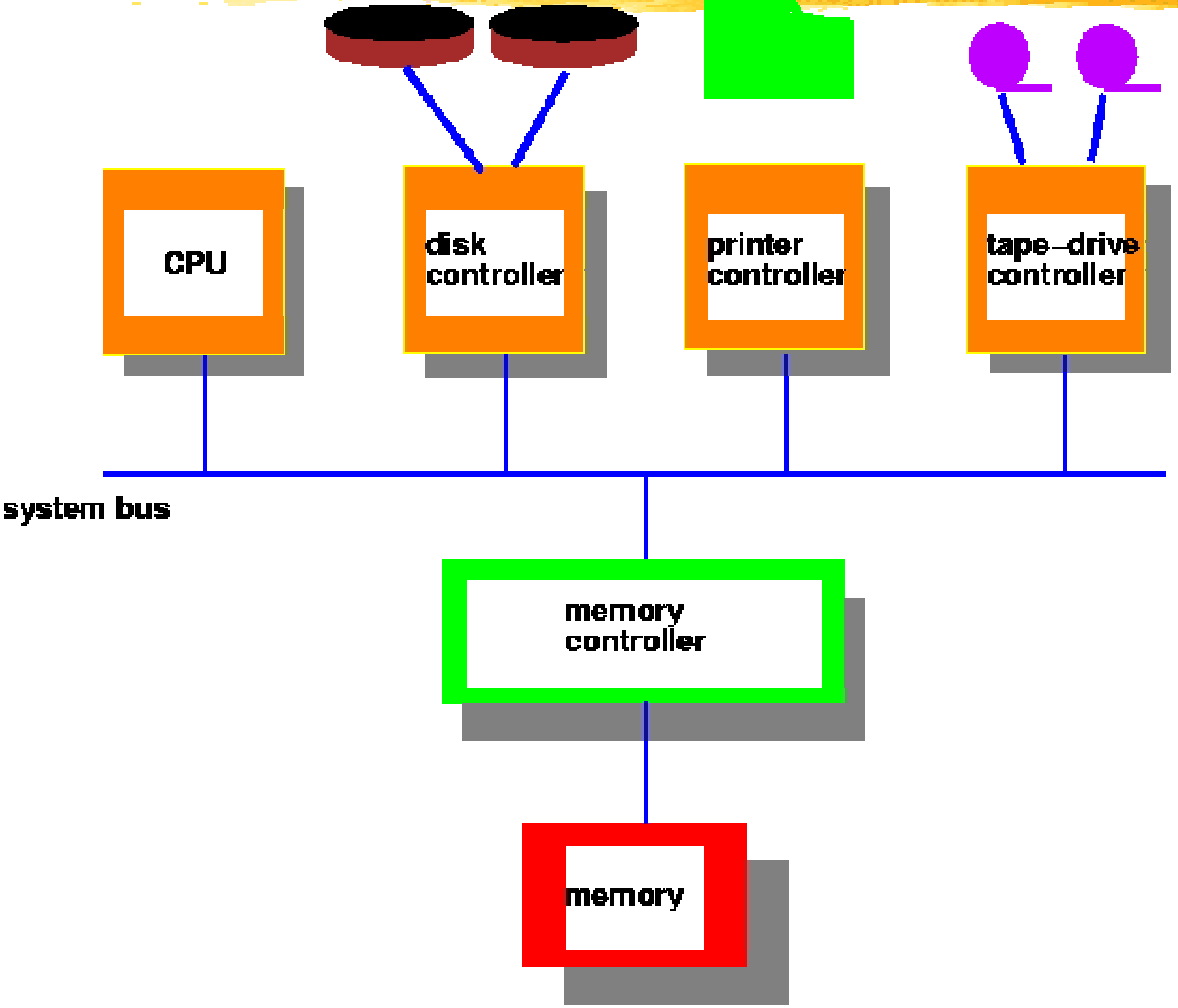
LECTURE-1

INTRODUCTION

Introduction

- What is an operating system?
- Early Operating Systems
  - Simple Batch Systems
  - Multiprogrammed Batch Systems
- Time-sharing Systems
- Personal Computer Systems
- Parallel and Distributed Systems
- Real-time Systems

Computer System Architecture



What is an Operating System?

- An OS is a program that acts an intermediary between the user of a computer and computer hardware.
- Major cost of general purpose computing is software.
  - OS simplifies and manages the complexity of running application programs efficiently.

Goals of an Operating System

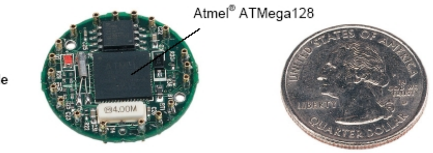
- Simplify the execution of user programs and make solving user problems easier.
- Use computer hardware efficiently.
  - Allow sharing of hardware and software resources.
- Make application software portable and versatile.
- Provide isolation, security and protection among user programs.
- Improve overall system reliability
  - error confinement, fault tolerance, reconfiguration.

Why should I study Operating Systems?

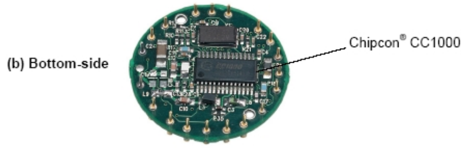
- Need to understand interaction between the hardware and applications
  - New applications, new hardware..
  - Inherent aspect of society today
- Need to understand basic principles in the design of computer systems
  - efficient resource management, security, flexibility
- Increasing need for specialized operating systems
  - e.g. embedded operating systems for devices - cell phones, sensors and controllers
  - real-time operating systems - aircraft control, multimedia services

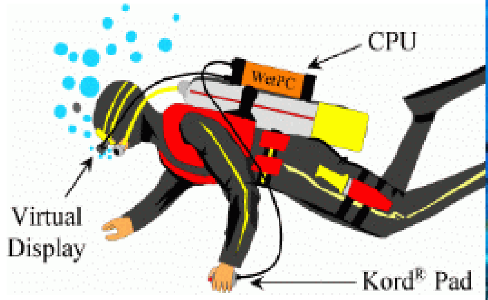
Systems Today

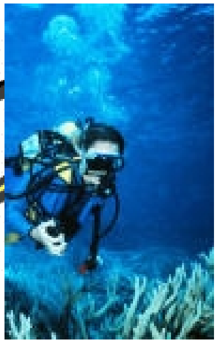
(a) Top-side

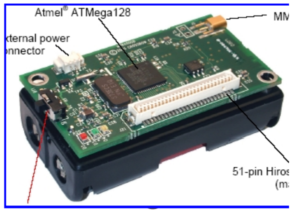


(b) Bottom-side



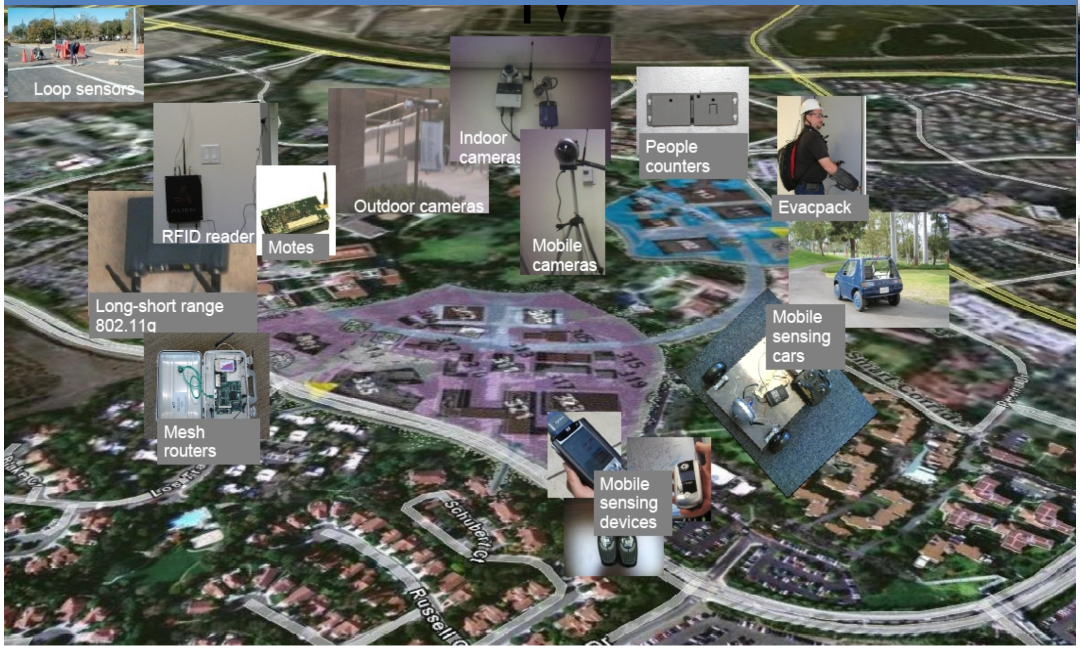






OnStar
by GM
WALNUT BLVD
25 yard
RPT Prvw Cncl
Front Sensors
Cellular Antenna
Sensing Diagnostic Module (SDM)
Side Sensors
OnStar Module

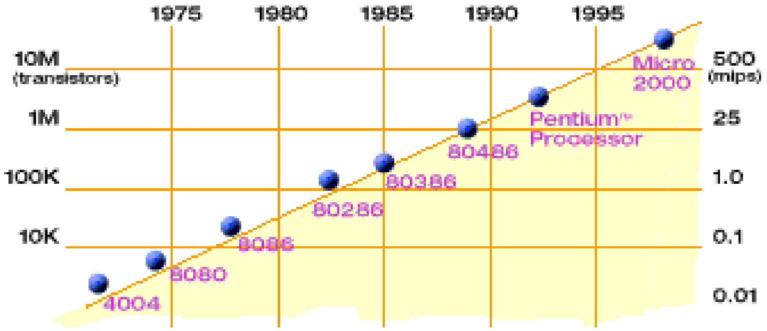
Automated Vehicle Check-In
ONLY

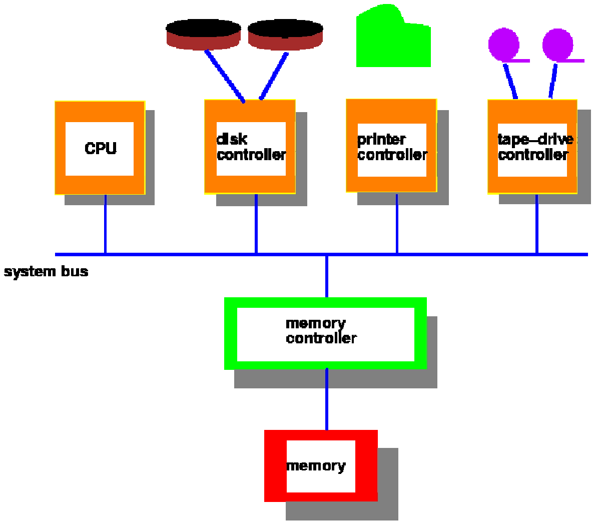

Loop sensors
Indoor cameras
People counters
Outdoor cameras
RFID reader
Motes
Mobile cameras
Evacpack
Long-short range 802.11a
Mobile sensing cars
Mesh routers
Mobile sensing devices

The Smart Grid Can Deliver
BENEFITS
Market
PHEV

Hardware Complexity Increases

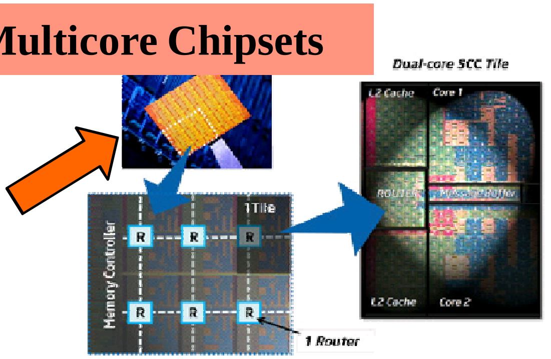
Moore's Law: 2X transistors/Chip Every 1.5 years

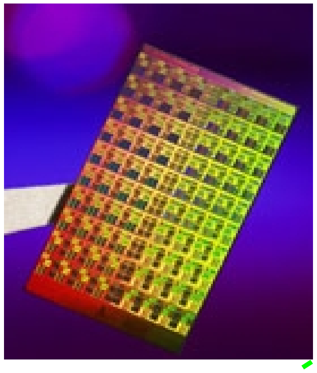




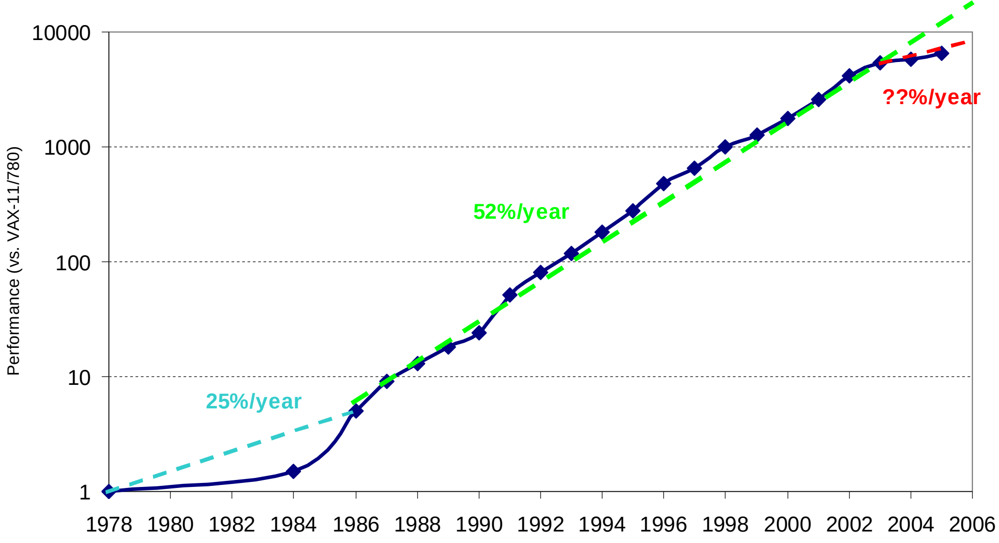
•

Intel Multicore Chipsets

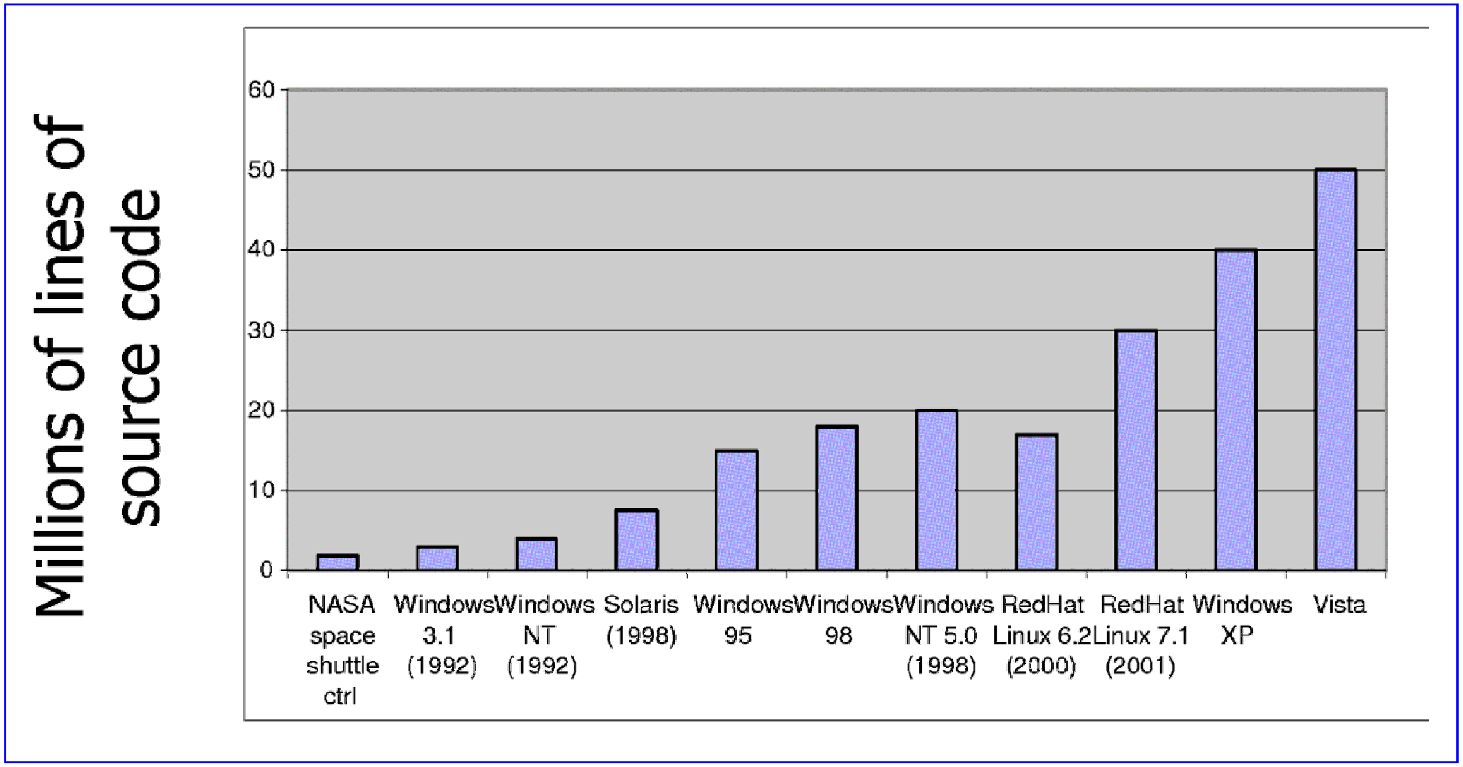




ıw



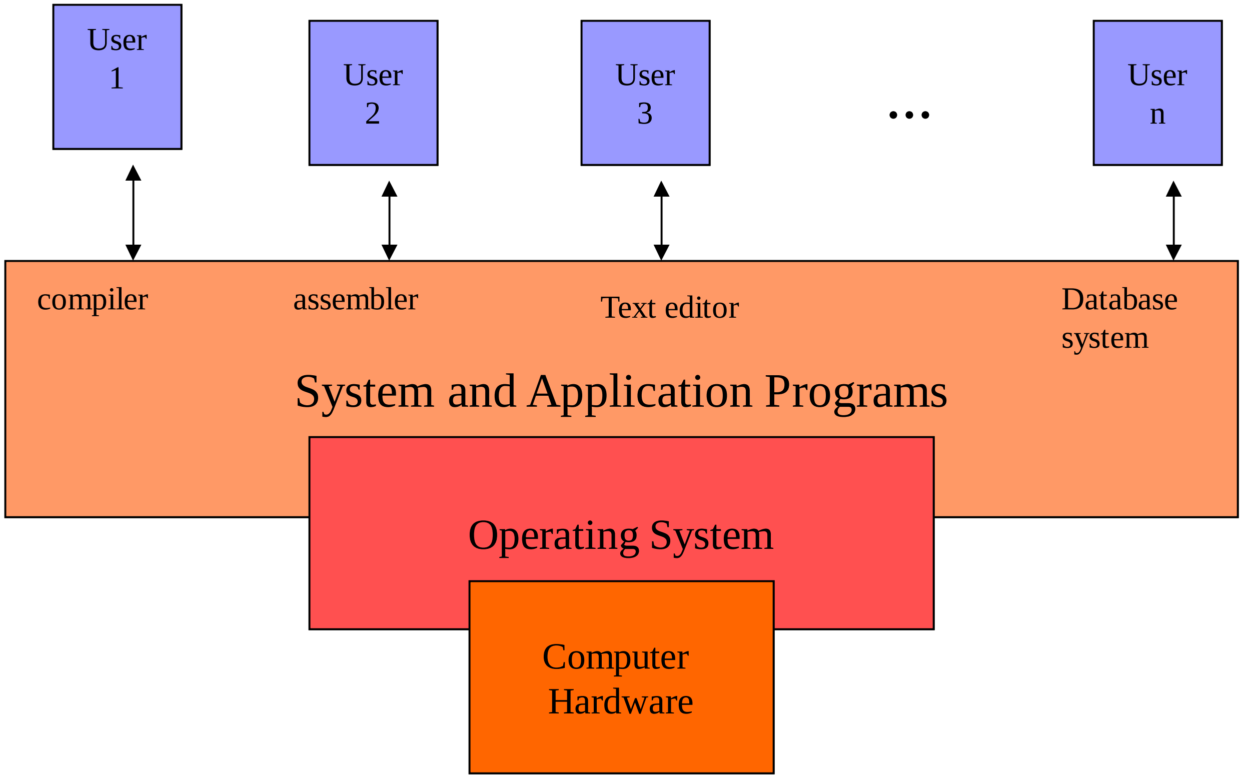
Software Complexity Increases



Computer System Components

- Hardware
  - Provides basic computing resources (CPU, memory, I/O devices).
- Operating System
  - Controls and coordinates the use of hardware among application programs.
- Application Programs
  - Solve computing problems of users (compilers, database systems, video games, business programs such as banking software).
- Users
  - People, machines, other computers

Abstract View of System



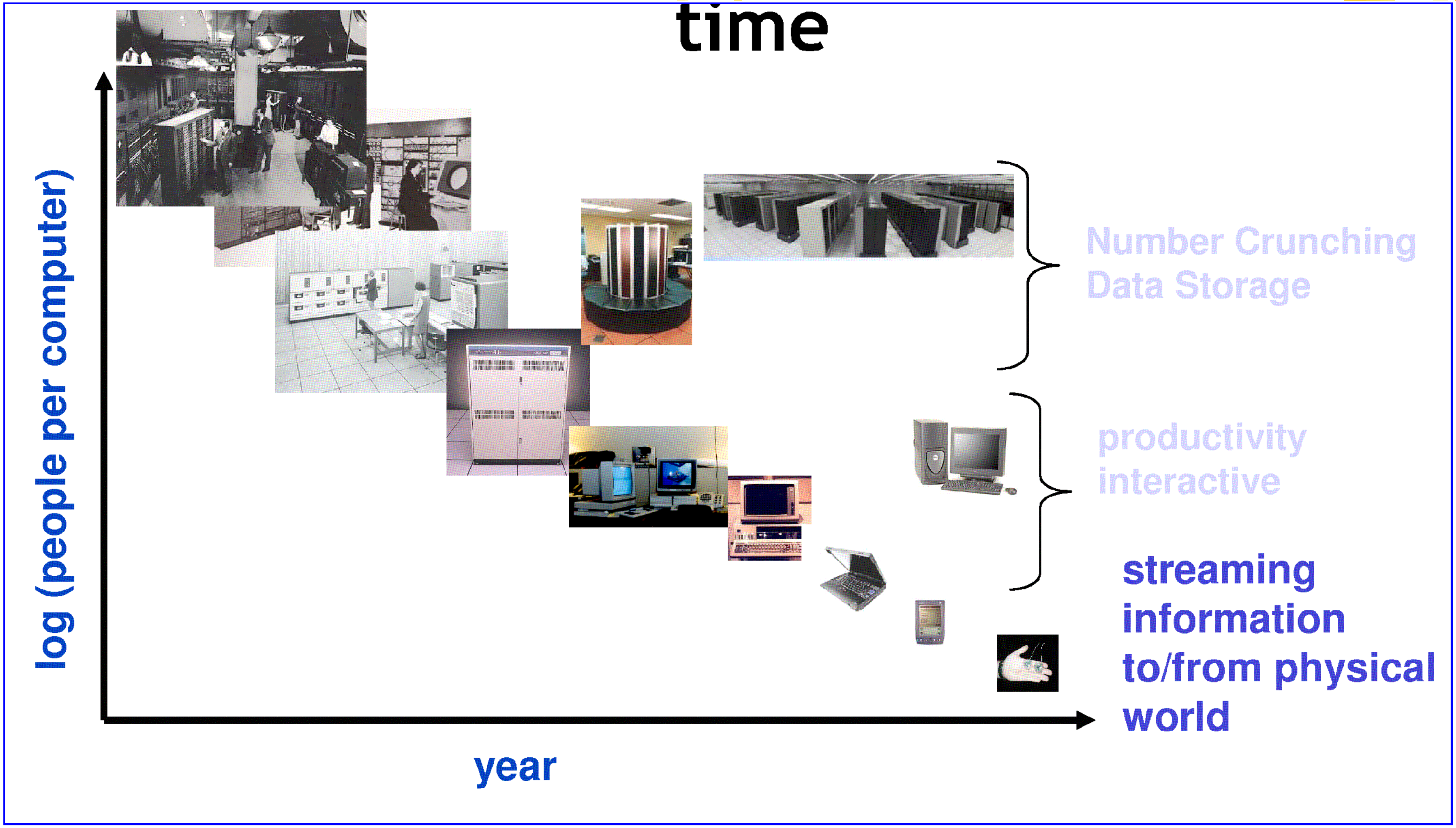
Operating System Views

- ⌘Resource allocator
  - ☒to allocate resources (software and hardware) of the computer system and manage them efficiently.
- ⌘Control program
  - ☒Controls execution of user programs and operation of I/O devices.
- ⌘Kernel
  - ☒The program that executes forever (everything else is an application with respect to the kernel).

Operating System Spectrum

- Monitors and Small Kernels
  - special purpose and embedded systems, real-time systems
- Batch and multiprogramming
- Timesharing
  - workstations, servers, minicomputers, timeframes
- Transaction systems
- Personal Computing Systems
- Mobile Platforms, devices (of all sizes)

People-to-Computer Ratio Over Time



Number Crunching
Data Storage

productivity
interactive

streaming information to/from physical world

Early Systems - Bare Machine (1950s)

Hardware – *expensive* ; Human – *cheap*

- ⌘ Structure
  - ☒Large machines run from console
  - ☒Single user system
    - Programmer/User as operator
  - ☒Paper tape or punched cards
- ⌘ Early software
  - ☒Assemblers, compilers, linkers, loaders, device drivers, libraries of common subroutines.
- ⌘ Secure execution
- ⌘ Inefficient use of expensive resources
  - ☒Low CPU utilization, high setup time.

Simple Batch Systems (1960's)

- Reduce setup time by batching jobs with similar requirements.
- Add a card reader, Hire an operator
  - User is NOT the operator
  - Automatic job sequencing
    - Forms a rudimentary OS.
  - Resident Monitor
    - Holds initial control, control transfers to job and then back to monitor.
  - Problem
    - Need to distinguish job from job and data from program.

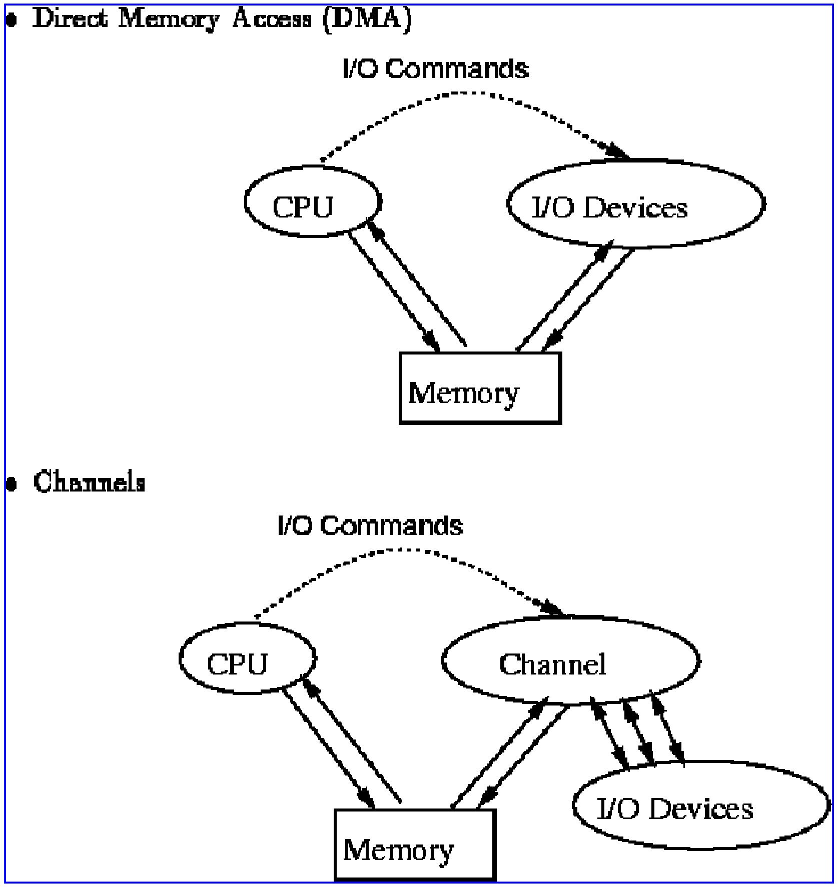
Supervisor/Operator Control

- Secure monitor that controls job processing
  - Special cards indicate what to do.
  - User program prevented from performing I/O
- Separate user from computer
  - User submits card deck
  - cards put on tape
  - tape processed by operator
  - output written to tape
  - tape printed on printer
- Problems
  - Long turnaround time - up to 2 DAYS!!!
  - Low CPU utilization
    - I/O and CPU could not overlap;   slow mechanical devices.

Batch Systems - Issues

- Solutions to speed up I/O:
- Offline Processing
  - load jobs into memory from tapes, card reading and line printing are done offline.
- Spooling
  - Use disk (random access device) as large storage for reading as many input files as possible and storing output files until output devices are ready to accept them.
  - Allows overlap - I/O of one job with computation of another.
  - Introduces notion of a job pool that allows OS choose next job to run so as to increase CPU utilization.

Speeding up I/O



- Direct Memory Access (DMA)

- Channels

Batch Systems - I/O completion

- How do we know that I/O is complete?
  - Polling:
    - Device sets a flag when it is busy.
    - Program tests the flag in a loop waiting for completion of I/O.
  - Interrupts:
    - On completion of I/O, device forces CPU to jump to a specific instruction address that contains the interrupt service routine.
    - After the interrupt has been processed, CPU returns to code it was executing prior to servicing the interrupt.

Summary of lecture

- What is an operating system?
- Early Operating Systems
- Simple Batch Systems