Operating System Principles

操作系统原理

Introduction

李旭东
leexudong@nankai.edu.cn
—Nankai Univ. SE.

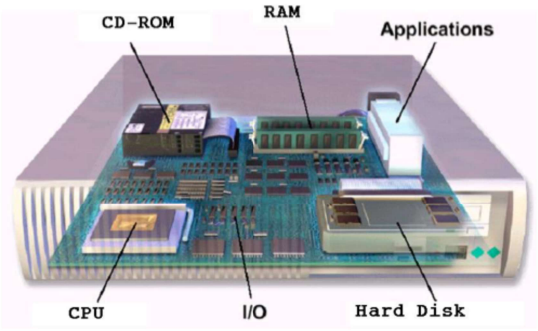
Objectives

- Computer System
- Operating System
- Operating System Zoo
- Operating System History
- Operating System Functions
- Operating System Characters
- Operating System Structure
- Research on OS

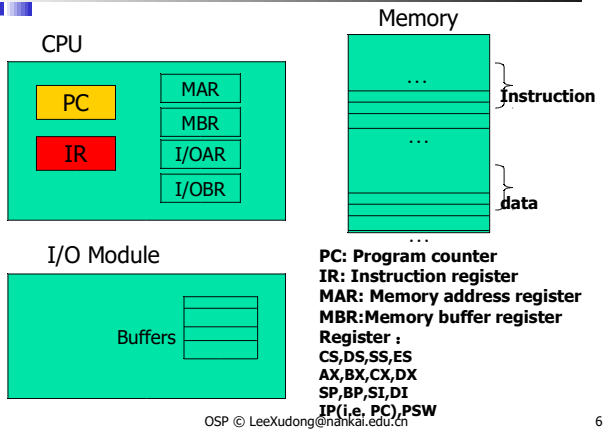
Computer System

Computer System

What is Inside Computer?

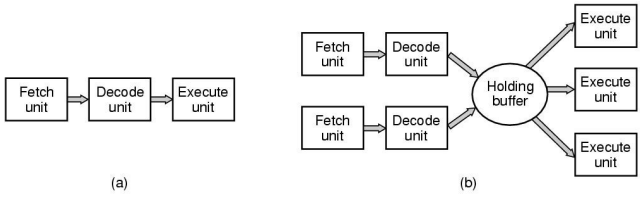


CPU



PC: Program counter
IR: Instruction register
MAR: Memory address register
MBR:Memory buffer register
Register :
CS,DS,SS,ES
AX,BX,CX,DX
SP,BP,SI,DI
IP(i,e, PC),PSW

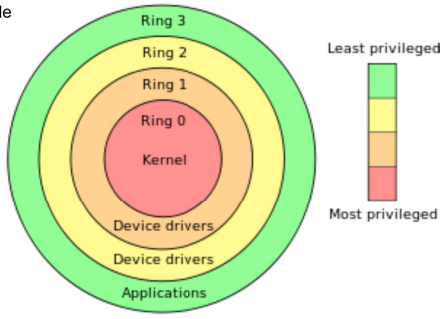
CPU Pipeline



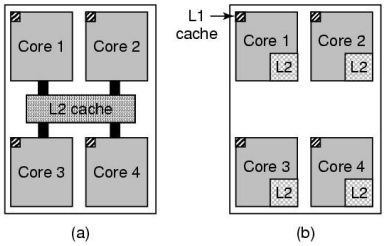
(a) A three-stage pipeline. (b) A superscalar CPU.

CPU: Supervisor / Protected Mode

- Kernel Mode
- User Mode

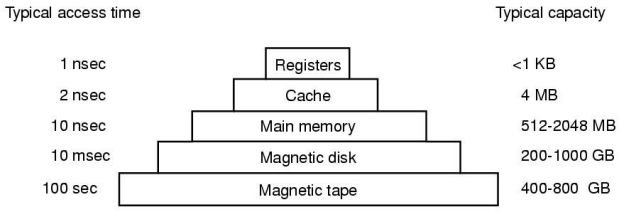


Multithreaded and Multicore Chips



(a) A quad-core chip with a shared L2 cache.
(b) A quad-core chip with separate L2 caches.

Memory



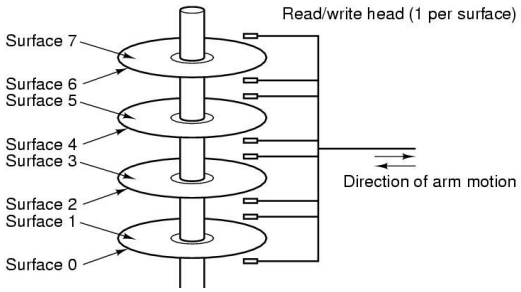
A typical memory hierarchy
The numbers are very rough approximations

Memory

Questions when dealing with cache:

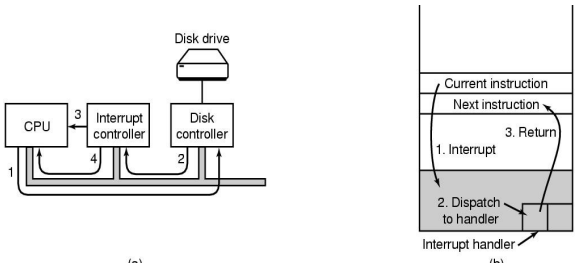
- When to put a new item into the cache.
- Which cache line to put the new item in.
- Which item to remove from the cache when a slot is needed.
- Where to put a newly evicted item in the larger memory.

Disks



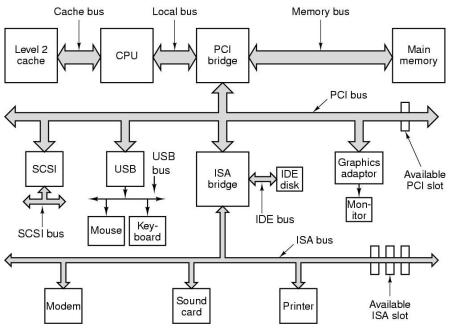
Structure of a disk drive.

I/O Devices



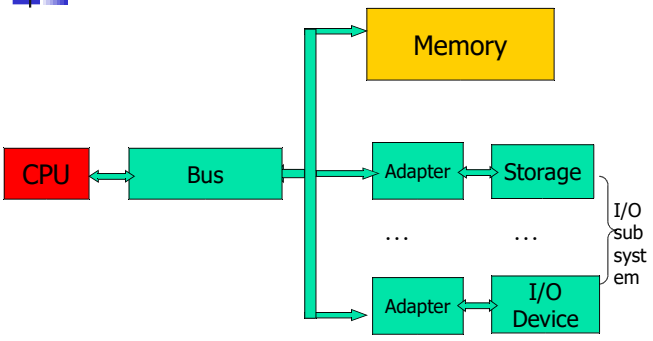
(a) The steps in starting an I/O device and getting an interrupt.

Buses

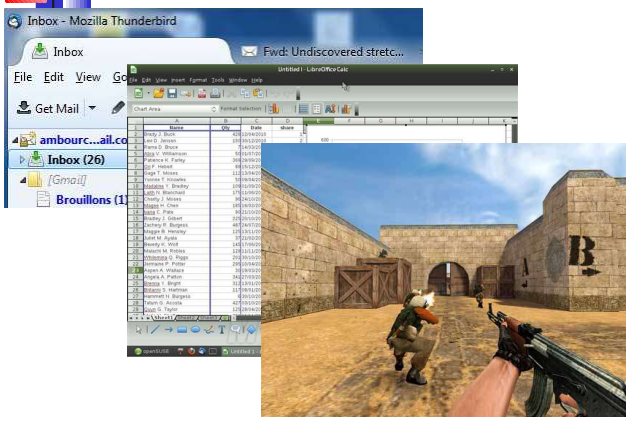


The structure of a large Pentium system

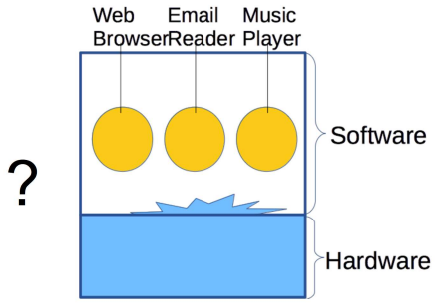
Computer System:Hardware



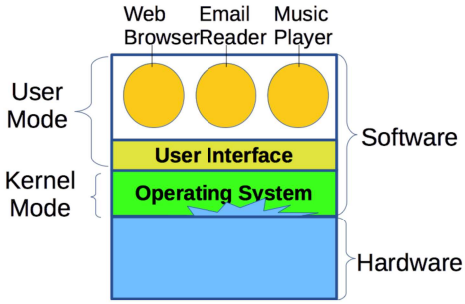
Computer System:Software



Layers of Computer System



Layers of Computer System



"Operating System"

Basic Services of OS

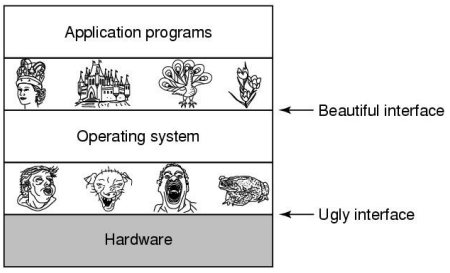
- Program Creation
- Program Execution
- Access to I/O Devices
- Controlled Access to Files
- System Access
- Error Detection and Response
- Accounting

What's an Operating System?

- http://en.wikipedia.org/wiki/Operating_system
- software that manages computer hardware and software resources and provides common services for computer programs
- an essential component of the system software in a computer system
- Application programs usually require an operating system to function

What's an Operating System?

- 1.The Operating System as an Extended Machine



What's an Operating System?

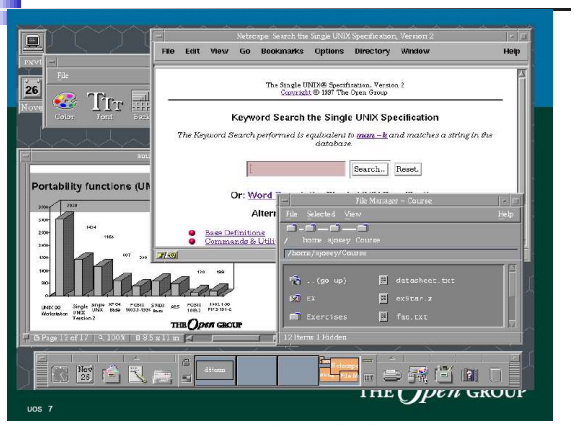
- 2.The Operating System as a Resource Manager
  - Allow multiple programs to run at the same time
  - Manage and protect memory, I/O devices, and other resources
  - Includes multiplexing (sharing) resources in two different ways:
    - In time 时间， In space 空间

What's an Operating System?

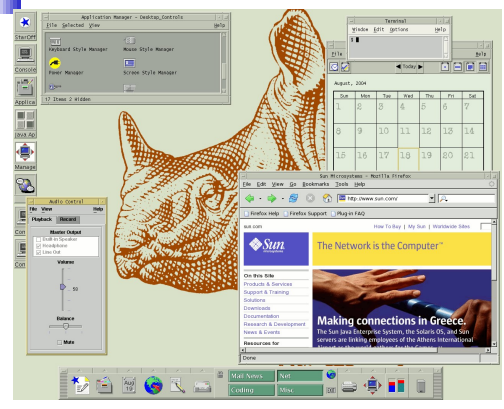
- (3).The Operating System as a Process Manager
  - Process Creation, Scheduling, Termination
- (4).The Operating System as an Extensible Service Machine
  - New Services

Operating System Zoo

UNIX 95

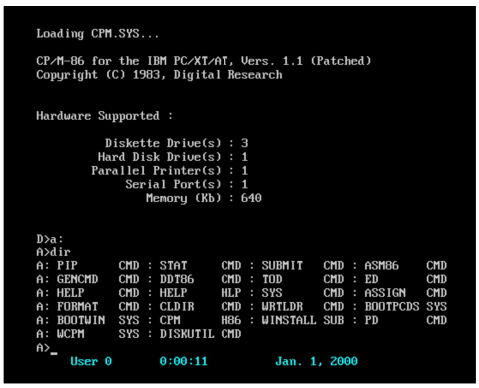


Sun Solaris



FreeBSD

CP/M



CP/M OS

- CP/M (Control Program for Microcomputers) was a mass-market operating system created for Intel 8080/85-based microcomputers by **Gary Kildall** of Digital Research.
- Initially confined to single-tasking on 8-bit processors and no more than 64 kilobytes of memory, later versions of CP/M added multi-user variations and were migrated to 16-bit processors.
- The combination of CP/M and S-100 bus computers loosely patterned on the MITS Altair was an early "industry standard" for microcomputers, and this computer platform was widely used in business through the late 1970s and into the mid-1980s, expanding to include 16-bit CPUs and multiuser capability

MS DOS/IBM DOS (QDOS)

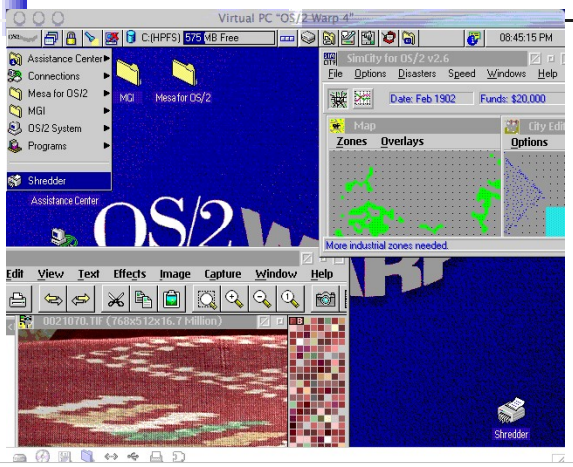
DOS:MS-DOS

- DOS
  - MS-DOS was a renamed form of 86-DOS – owned by Seattle Computer Products, written by **Tim Paterson**. Development of 86-DOS took only six weeks, as it was basically a clone of Digital Research's CP/M (for 8080/Z80 processors), ported to run on 8086 processors and with two notable differences compared to CP/M, an improved disk sector buffering logic and the introduction of FAT12 instead of the CP/M filesystem. This first version was shipped in August 1980.
  - Microsoft, which needed an operating system for the then-new Intel 8086 but had none available, hired Tim Paterson in May 1981 and bought 86-DOS 1.10 for **$75,000** in July of the same year.
  - Microsoft kept the version number, but renamed it MS-DOS. They also licensed MS-DOS 1.10/1.14 to IBM, who, in August 1981, offered it as PC DOS 1.0 as one of three operating systems[10] for the IBM 5150, or the IBM PC.
  - Single user, Single task OS
- DOS Components
  - BOOT MBR, IO.SYS, MSDOS.SYS, COMMAND.COM

Bill Gates

Tim Paterson

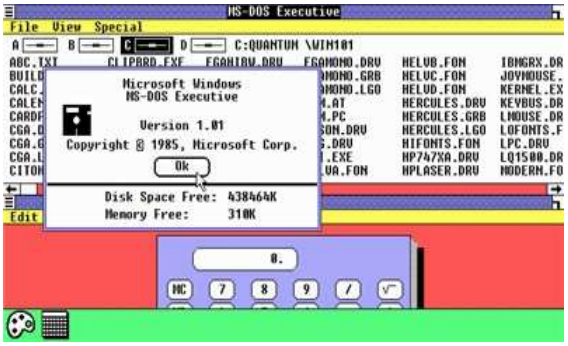
IBM OS/2



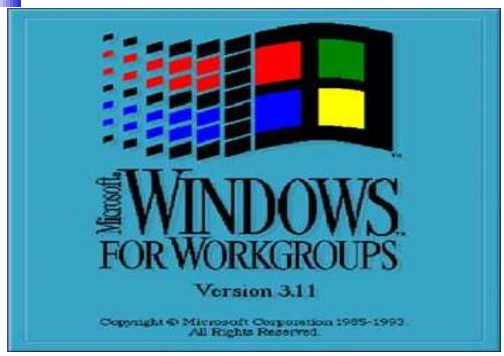
Microsoft Windows

- 1983.11.10, Bill Gate anounced Windows
- 1985.11.20, Windows 1.0
- 1992.4, Windows 3.1
- 1993.5, Windows NT (Dave Cutler)
- 1995, Windows 95
- Windows 98,Windows CE,Windows 2000/XP, Windows Server 2003 （.net）, Windows Server 2008, Windows Server 2012
- Vista
- Windows 7, Windows 8, Windows 10

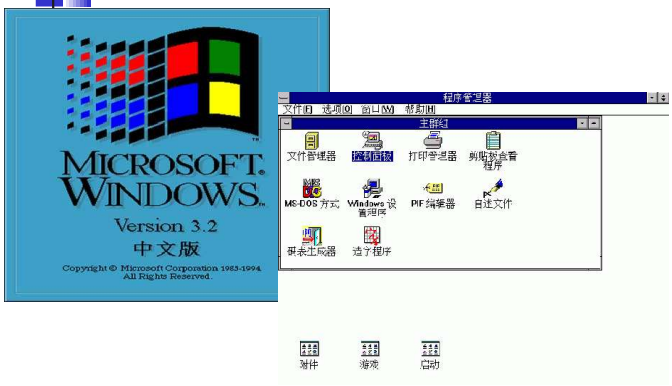
Windows 1.01



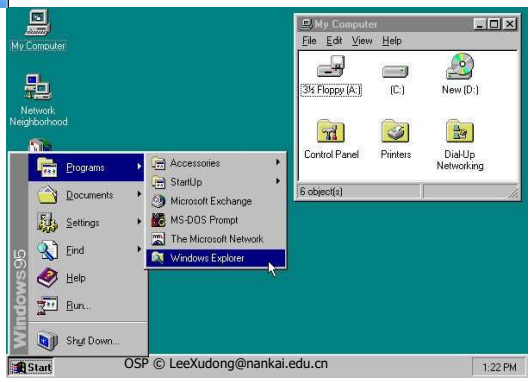
Windows 3.1.1



Windows 3.2



Windows 95



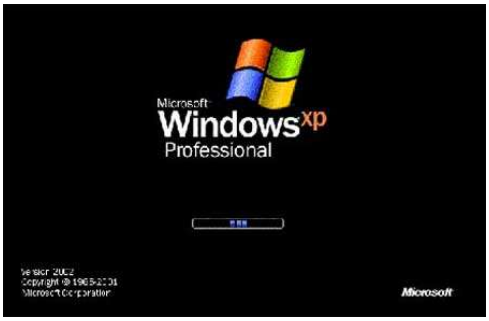
Window NT 4.0

Windows 98

Windows ME

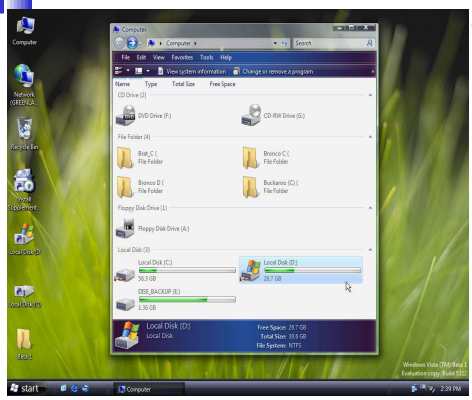
Windows 2000

Windows XP



Windows Server 2003

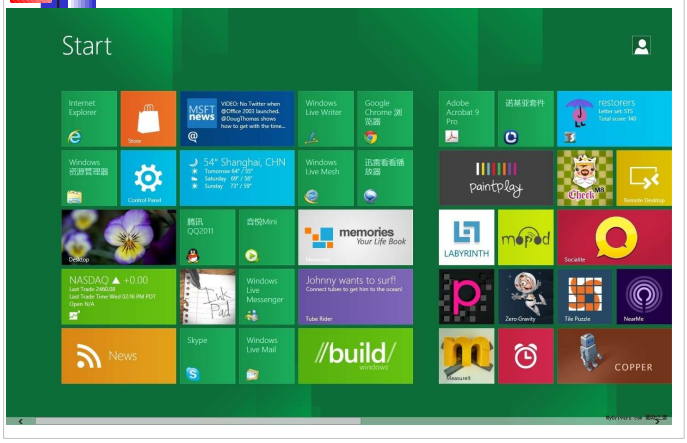
Windows Vista



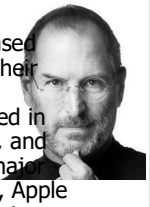
Windows Vista

Windows 7

Windows 8

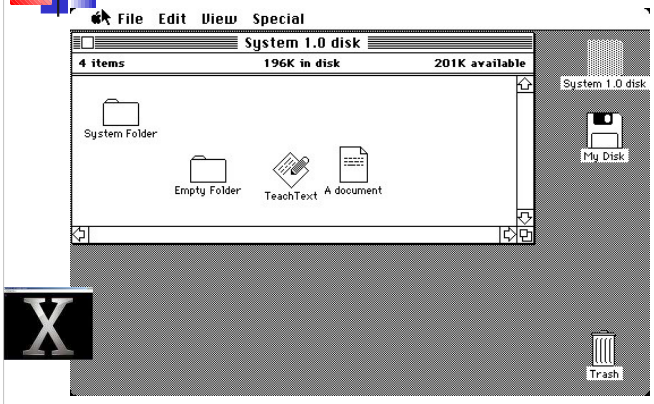


Apple's Macintosh & MAC OS

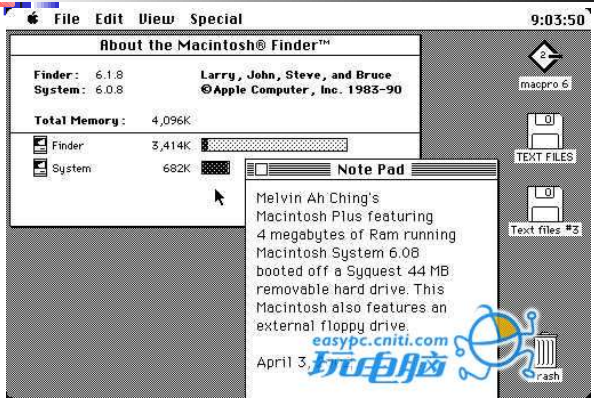


- Mac OS is a series of graphical user interface-based operating systems developed by Apple Inc. for their Macintosh line of computer systems.
- The original operating system was first introduced in 1984 as being integral to the original Macintosh, and referred to as the "System". Referred to by its major revision starting with "System 6 and "System 7", Apple rebranded version 7.6 as "Mac OS" as part of their Macintosh clone program in 1996.
- Macintosh operating systems have been released in two major series. Up to major revision 9, from 1984 to 2000, it is historically known as Classic Mac OS.
- Major revision 10, from 2001 to present, is branded OS X (originally referred to as Mac OS X).

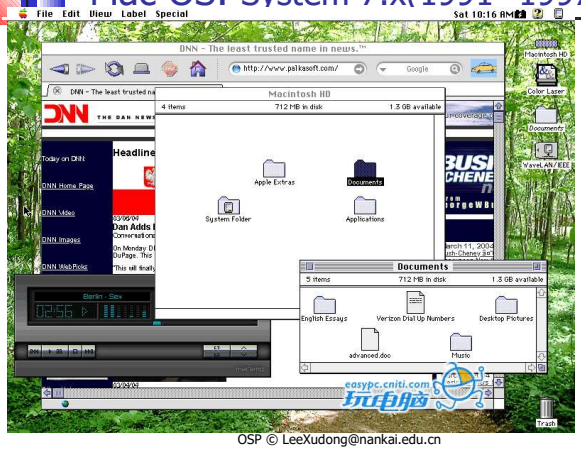
Mac OS: System 0.0-1.1 (1984)



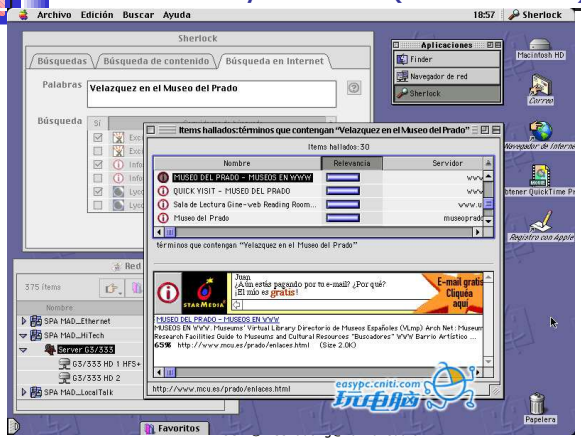
Mac OS: System 2.0 ~ 6.x(1985~1988)



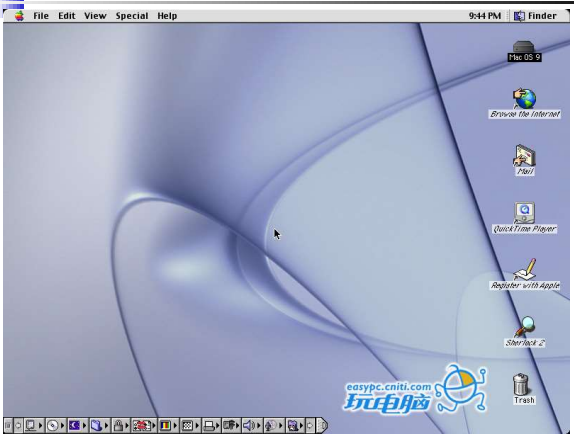
Mac OS: System 7.x(1991~1997)



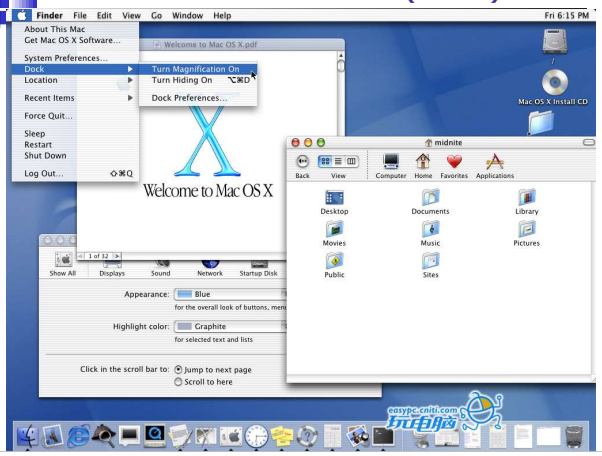
Mac OS: System 8.x(1997~1998)



Mac OS: System 9(1999)



Mac OS X Public (2000)

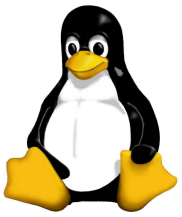


Apple Mac OS …

OS X Yosemite:10.10

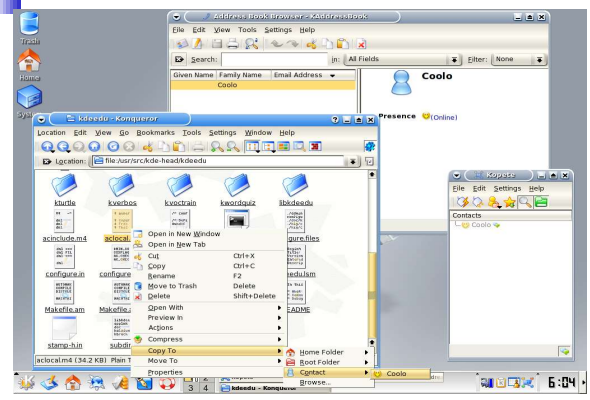
Apples

Linux

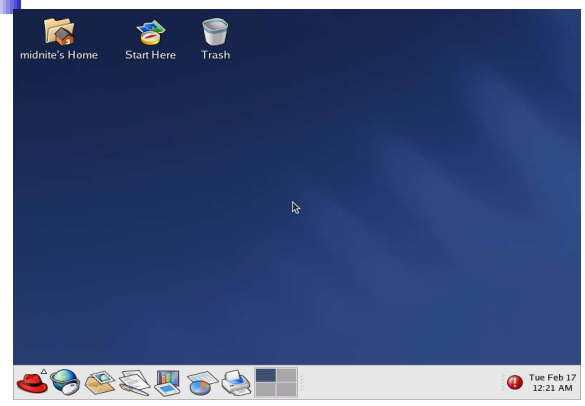


- Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on 5 October 1991 by Linus Torvalds.
- The Free Software Foundation uses the name GNU/Linux to describe the operating system, which has led to some controversy.

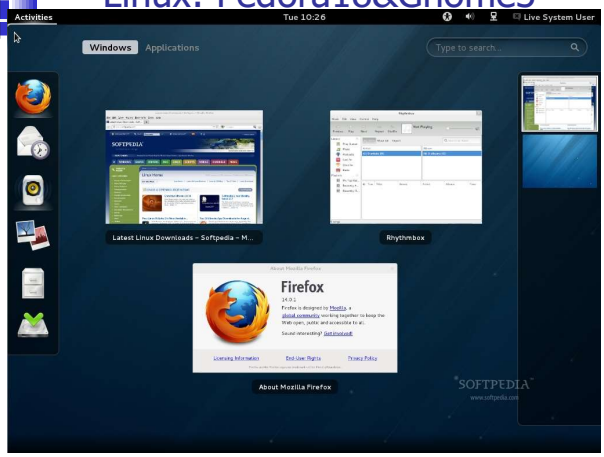
Linux: KDE Desktop



Linux: GNOME Desktop



Linux: Fedora18&Gnome3



GNU/Linux distributions

- Red Hat Linux
  - www.redhat.com,Linux
- Slackware Linux: www.cdrom.com
- Debian Linux:www.debian.org
- Open Linux: www.caldera.com
- S.U.S.E Linux:www.suse.com
- DLX Linux: Erich Boehm
- DOS Linux: Kent Robotti
- Linux Pro+: WorkGroup Solutions
- Stampede Linux:Stampede
- Turbo Linux:Pacific Hi-Tech
- Ubuntu
- Fedora

…

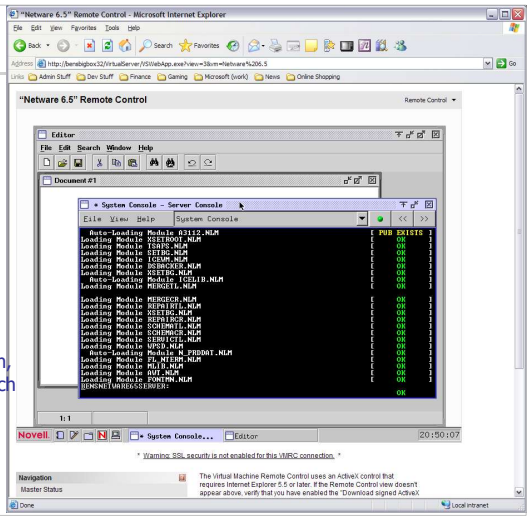
Android: linux+java

Andy Rubin,2003

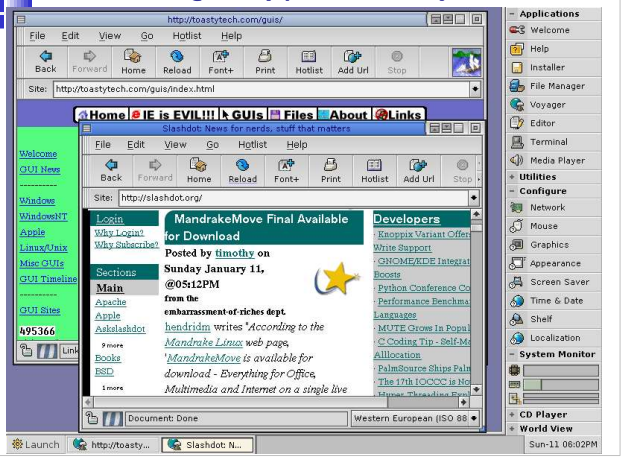
Novell Netware:

Multi-task,
Multi-User,
Network OS

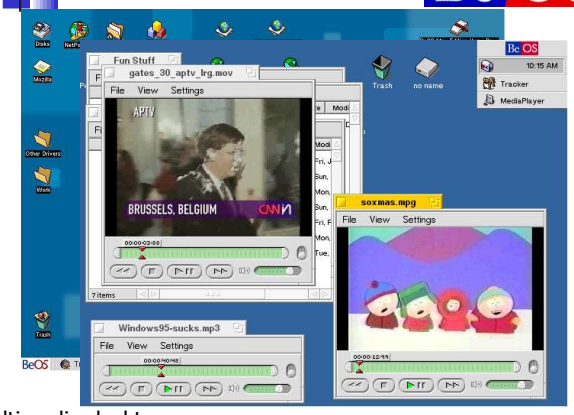
System Fault Tech,
Open Protocal Tech



QNX *(quick UNIX)*



BeOS



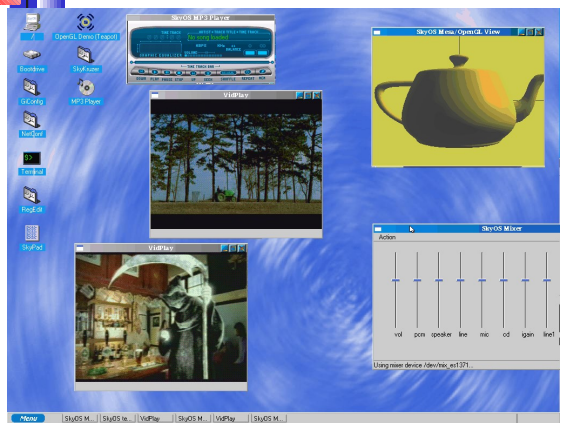
Multimedia desktop

GNU-Darwin OS

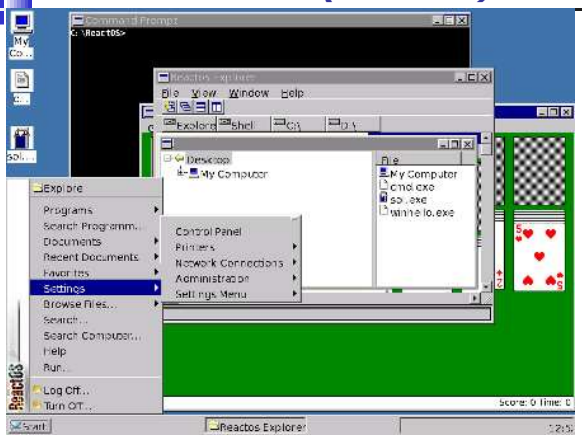
B-right/V (Japan)

SkyOS

■80s: Robert Szeleney, Kelly Rush



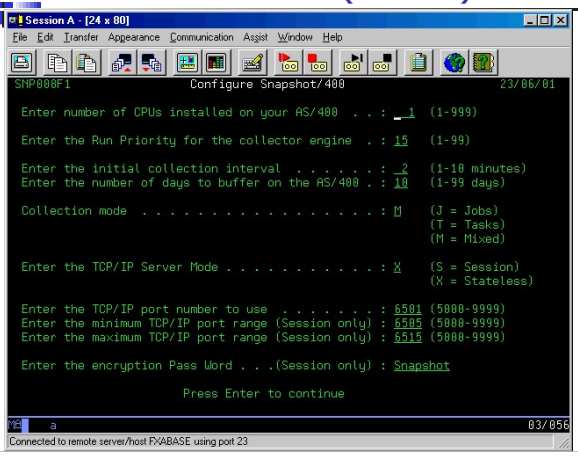
ReactOS *(FreeWin95)*



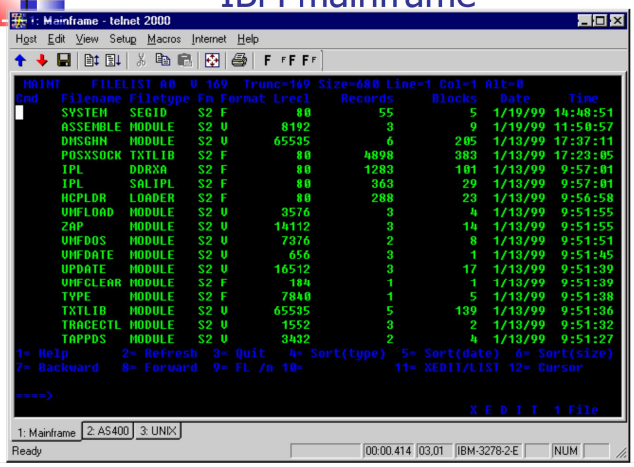
MenuetOS

X86 Assemble

IBM i5OS (AS400)



IBM mainframe



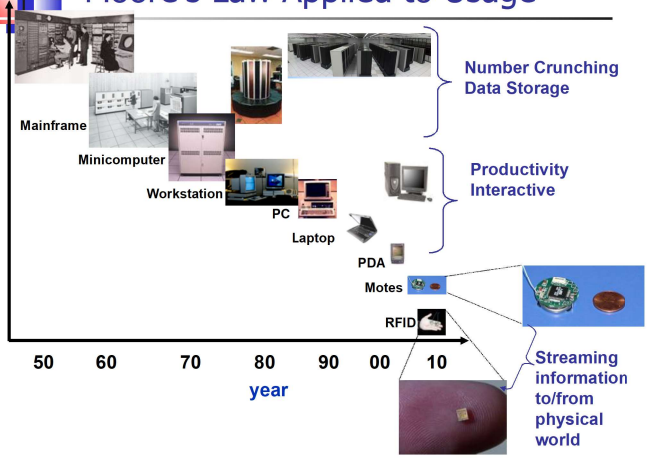
Chinese Operating Systems

- past?
- present?
- future?

Operating System Zoo

- Mainframe OS
- Server OS
- Multiprocessor OS
- Personal Computer OS
- Handheld Computer OS
- Embedded OS
- Sensor Node OS
- Real-time OS
- Smart Card OS

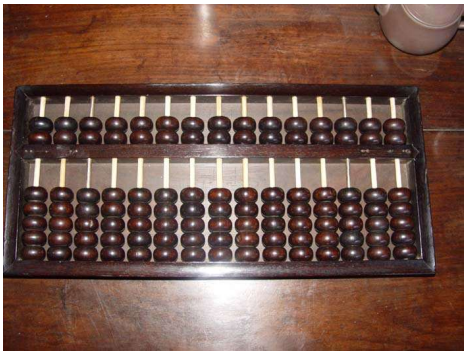
Moore's Law Applied to Usage



History of OS

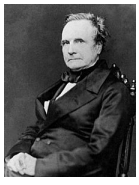
The First Computer

Abacus?



The First Mechanical Computer

- Charles Babbage's difference engine
  - a programmable computer
  - **Charles Babbage**, 1792-1871, an English polymath, a mathematician, philosopher, inventor and mechanical engineer
  - father of the computer
  - The first programmer: **Ada**



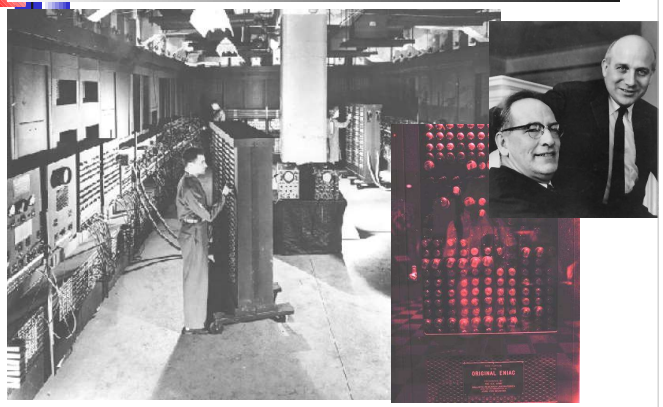
Gear Mechanical Computer

齿轮式计算机

The Most Famous Modern Computer

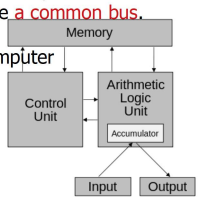
- ENIAC: Electronic Numerical Integrator And Computer
  - the first electronic general-purpose computer
  - **a programmable computer**
  - Electronic: Vacuum Tubles
  - Numerical: **Binary**
  - It cost almost $500,000 (approximately $6,000,000 today)
  - ENIAC contained 17,468 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors and around 5 million hand-soldered joints. It weighed more than 30 short tons (27 t), was roughly 8 by 3 by 100 feet (2.4 m × 0.9 m × 30 m), took up 1800 square feet (167 m*m), and consumed 150 kW of power.
- ENIAC was initially designed to calculate artillery firing tables for the United States Army's Ballistic Research Laboratory.
- It had a speed of one thousand times that of electro-mechanical machines.
- On July 29, 1947, it was turned on and was in continuous operation until 11:45 p.m. on October 2, 1955.
- ENIAC was conceived and designed by John Mauchly and J. Presper Eckert of the University of Pennsylvania.

ENIAC



Von Neumann architecture
冯诺依曼架构式计算机

- Von Neumann Model, Princeton Model
  - This describes a design architecture for an electronic digital computer with parts consisting of **a processing unit** containing an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, **a memory** to store both data and instructions, **external mass storage**, and **input and output** mechanisms.
  - The meaning has evolved to be any **stored-program** computer 存储程序计算机 in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus.
  - advancement over ENIAC etc.,
  - EDVAC:Electronic Discrete Variable Automatic Computer
  - the Von Neumann bottleneck

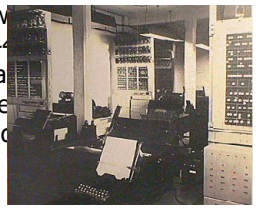


John Von Neumann

- John von Neumann: 1903 ~ 1957
  - a Hungarian and later American pure and applied mathematician, physicist, inventor, polymath, and polyglot. He made major contributions to a number of fields,including mathematics, physics , economics , computing (Von Neumann architecture, linear programming, self-replicating machines, stochastic computing), and statistics.
  - a pioneer of the application of operator theory to quantum mechanics, in the development of functional analysis, a principal member of the Manhattan Project and the Institute for Advanced Study in Princeton, and a key figure in the development of game theory and the concepts of cellular automata, the universal constructor, and the digital computer
  - Von Neumann's mathematical analysis of the structure of self-replication preceded the discovery of the structure of DNA
- Von Neumann architecture

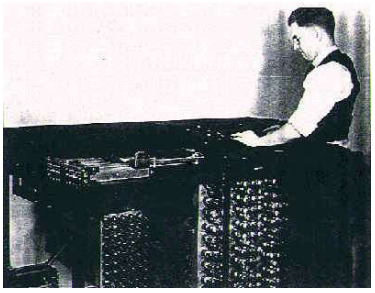
Colossus

- Colossus
  - the world's first electronic digital computer that was programmable
  - developed for British codebreakers during World War II to help in the cryptanalysis of the Lorenz cipher.
  - designed by the engineer Tommy Flowers
  - The prototype, Colossus Mark 1, was shown to be working in December 1943 and w Bletchley Park by 5 February 194
  - An improved Colossus Mark 2 tha to quintuple the speed,first worke 1944,just in time for the Normand



ABC

- ABC: Atanasoff-Berry Computer
  - 1939, John Vincent Atanasoff
  - The First Automatic Electronic Digital Computer



ABC: Atanasoff Berry Computer

- John Vincent Atanasoff and Clifford E. Berry
- http://www.ieee.org/web/aboutus/history_center/atanasoff.html
  - John Vincent Atanasoff conceived basic design principles for the first electronic-digital computer in the winter of 1937 and, assisted by his graduate student, Clifford E. Berry, constructed a prototype here in October 1939.
  - It used binary numbers, direct logic for calculation, and a regenerative memory. It embodied concepts that would be central to the future development of computers.
  - Atanasoff wrote most of the concepts of the first modern computer on the back of a cocktail napkin.
  - in late 1939, John V. Atanasoff teamed up with Clifford E. Berry to build a prototype. They created the first computing machine to use electricity, vacuum tubes, binary numbers and capacitors.
  - The final product was the size of a desk, weighed 700 pounds, had over 300 vacuum tubes, and contained a mile of wire. It could calculate about one operation every 15 seconds, today a computer can calculate 150 billion operations in 15 seconds.

Evolution of An OS

- Maximization of resource utilization
- Hardware upgrades plus new types of hardware
- New Services
- Fixes
- User Experience

History of Operating Systems

- First generation 1945 - 1955
  - vacuum tubes, plug boards
- Second generation 1955 - 1965
  - transistors, batch systems
- Third generation 1965 – 1980
  - ICs and multiprogramming
- Fourth generation 1980 – present
  - Personal computers
- Fifth generation 1990 – present
  - Mobile computers

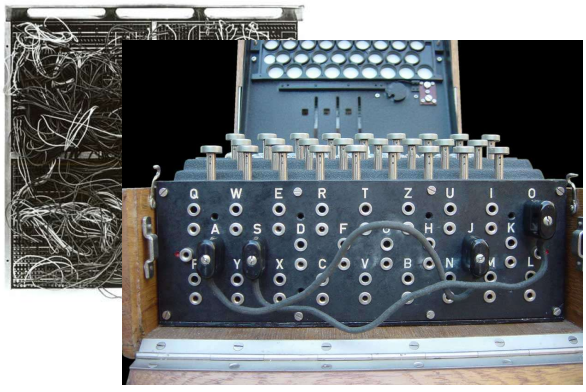
History of OS: prehistory

- Vacuum Tubes
- Plugboard
- No OS
- Machine Language

Vacuum Tubes

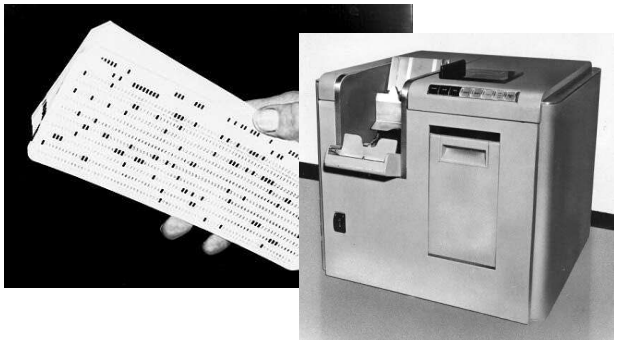
- Vacuum Tubes
  - Lee De Forest, 1906
  - Two States

Plugboard



punched cards

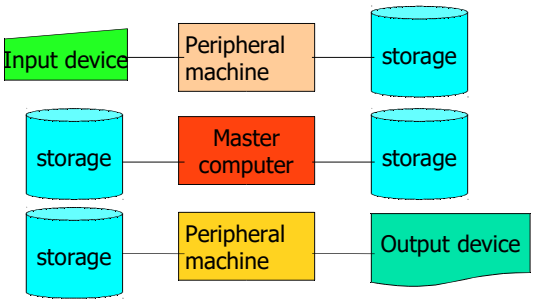
- punched cards, Herman Hollerith, 1890



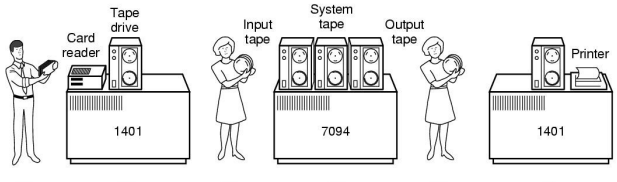
History of OS: batch system

- Second Generation,1955~1965
- Transistors and batch system
  - Transistor:1947, John Bardeen, Walter Brattain, and William Shockley
  - Mainframes: IBM1401-> IBM7094
    - IBM7094: good at numerical calculations
    - IBM1401: business
  - Tape
  - Assembly Language, FORTRAN Math Language
  - OS
    - FORTRAN Monitor System(FMS)
    - IBSYS(IBM7094 OS)
  - Job
  - off-line
  - single batch system

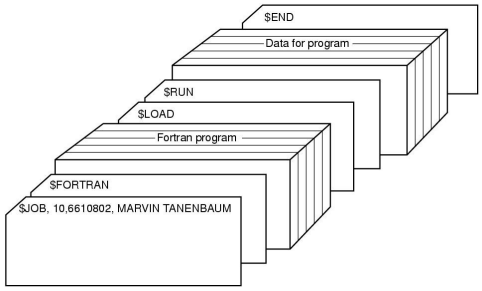
Off-Line I/O



Off-Line I/O



Single Batch System

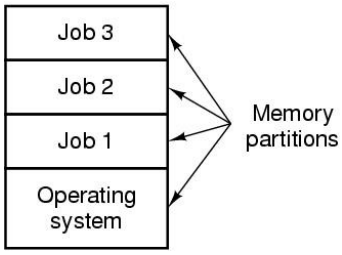


Structure of a typical FMS job.

History of OS: time sharing system

- The Third Generation, 1965~1980
- ICs and Multiprogramming
  - ICs: Integrated Circuits
    - Jack Kilby, Robert Noyce
  - Computer Architecture
    - IBM System/360, 370, 4300, 3080, 3090
  - OS/360:Fred Brooks
  - Multiprogramming*
  - Spooling*
  - timesharing: CTSS, Corbato(1962, MIT
  - PDP-1: Small Computer
  - MULTICS(MULTiplexed Information and Computing Service
    - 1965, MIT,Bell Lab,GE
  - UNIX: Single MULTICS, Ken Thompson(Bell Lab,PDP-7)

Multi-programming



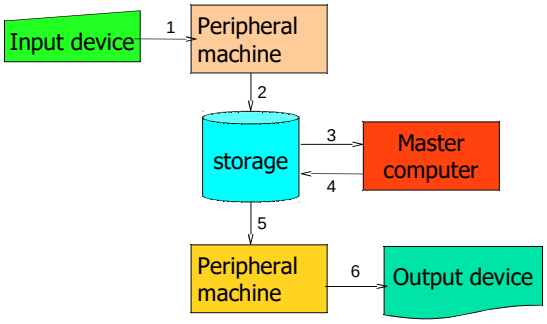
A multiprogramming system with three jobs in memory

Multi-programming

- Multiprogramming
  - Multitasking is a method where multiple tasks (also known as processes) are performed during the same period of time – they are executed concurrently (in overlapping time periods, new tasks starting before others have ended) instead of sequentially (one completing before the next starts)
  - The tasks share common processing resources, such as central processing units (CPUs) and main memory

SPOOLing 假脱机

- Simultaneous Peripheral Operating On-line

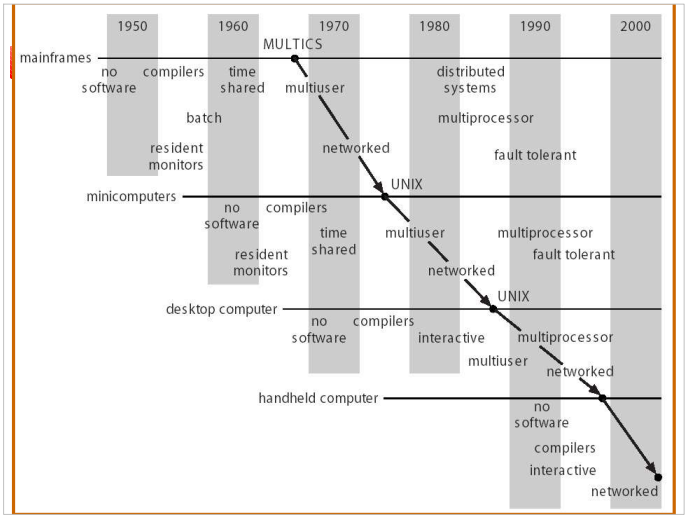


History of OS: modern

- Hardware
  - 32-bit x86-based PCs, Compaq Alpha AXP, Sun SPARC, UltraSPARC, Motorola 68000, PowerPC, PowerPC64, ARM, Hitachi SuperH, Cell, IBM S/390, MIPS, HP PA-RISC, Intel IA-64, DEC VAX, AMD x86-64, AXIS CRIS,Xtensa, Tilera TILE, AVR32 and Renesas M32R

History of OS: modern

- The Fourth Generation:1980~
  - LSI(Large Scale Integration) circuits, chips technology
  - Unix*
  - Intel80x86*
  - Desktop OS for Personal Computer
    - CP/M, DOS, ...
  - Network OS
  - Distributed OS
  - GUI(Graphical User Interface), user friendly
    - X-Window System



History of OS: modern

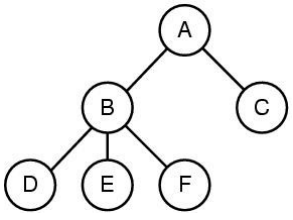
- The Fifth Generation:1990~
  - Handheld phone :1970s
  - PDA (Personal Digital Assistant): 1990, Nokia
  - Smartphone: 1997, Ericssion
  - Symbian OS
  - RIM's Blackberry OS
  - Iphone: Apple's iOS
  - Android
  - Windows Phone

Basic Concepts of OS

OS Basic Concepts

- Processes
- Address spaces
- Files
- Input/Output
- Protection
- The shell
- System Call

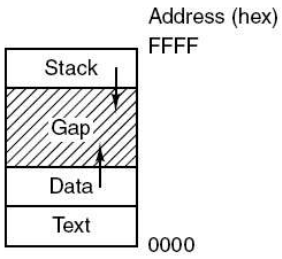
Processes



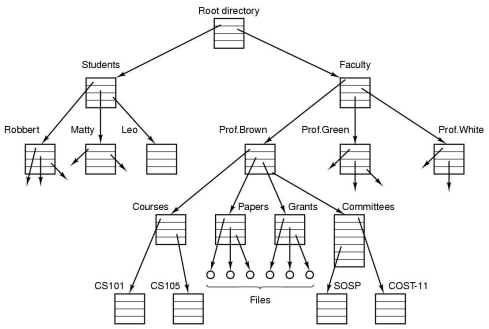
A process tree.
Process A created two child processes, B and C. Process B created three child processes, D, E, and F.

Address Spaces

- 8 bits, 16 bits
- 32 bits, 64 bits
- Physical memory
- Virtual Memory



Files



Files

- Files and Directories
  - Root directory,working directory
  - Path name:/,\
  - File hierarchies are organized as tree
  - File system: root file system
  - Special file
    - block special files 、 character files
  - File descriptor
  - Mount, umount

Shell

- shell
  - Command interpreter:shell
  - Prompt
    - >,#,$
  - Execute Commands:
    - #cat file1 file2 file3 l sort >/dev/lp &
  - Environment variables:
    - $#,$*,$?,$HOME,$PATH,$PS1

A Simple Shell

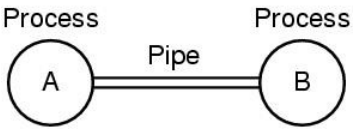
```
#define TRUE 1

while (TRUE) {                          /* repeat forever */
      type_prompt( );                   /* display prompt on the screen */
      read_command(command, parameters);/* read input from terminal */

      if (fork( ) != 0) {               /* fork off child process */
           /* Parent code. */
           waitpid(-1, &status, 0);     /* wait for child to exit */
      } else {
           /* Child code. */
           execve(command, parameters, 0);  /* execute command */
      }
}
```

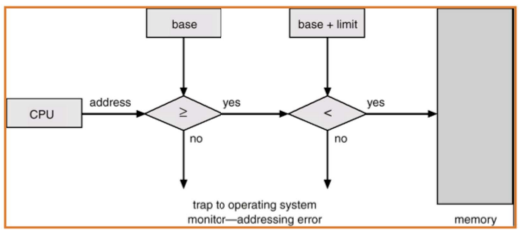
Input/Output

- I/O Subsystem
- IPC: Pipe



Protection

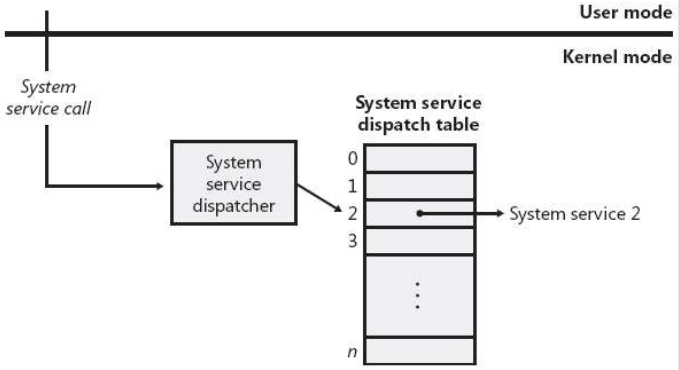
- Hardware
- Software

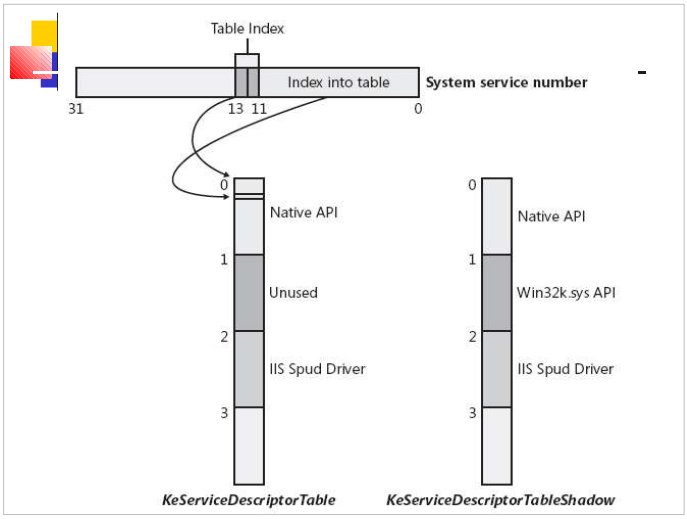


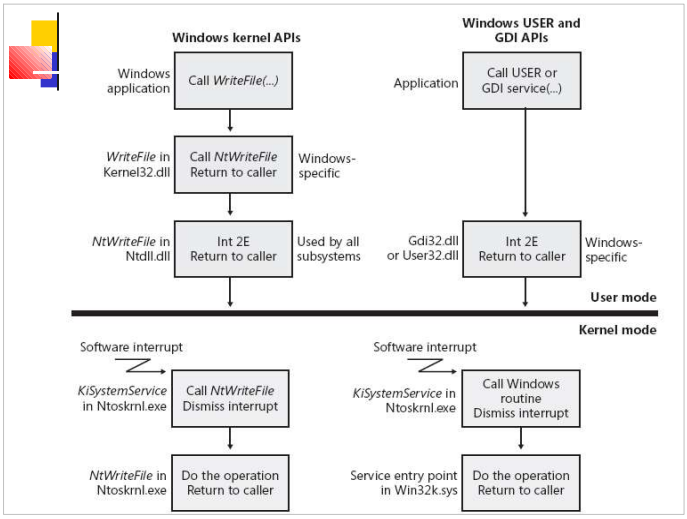
System Call

- System call
  - The interface between user programs and the operating system
  - Executed in kernel mode
  - Computer system running state
    - supervisor mode,kernel mode
    - user mode
- Trap Instruction
  - User mode to kernel mode
- Library Procedure
  - Encapsulates the trap instruction
  - Executed in user mode

System Call

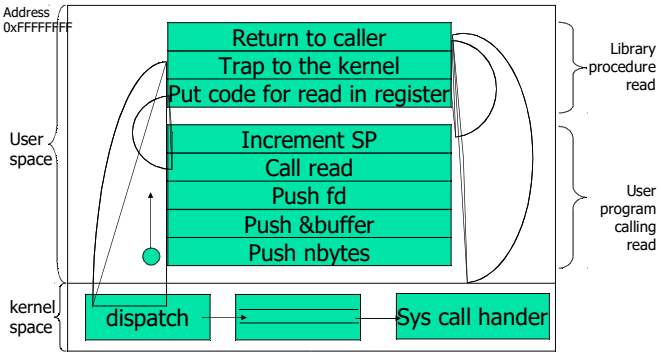






System call Case: read

- read(fd,buffer,nbytes)



Implementation of trap

- On x86 processors prior to the Pentium II
  - **int 0x2e**
- On x86 Pentium II processors and higher
  - Windows uses the special **sysenter** instruction
- On K6 and higher 32-bit AMD processors
  - Windows uses the special **syscall** instruction
- Case

```
NtWriteFile:
  mov  eax, 0x0E   ;system service number
  mov  ebx,esp     ;point to parameters
  int  0x2E        ;system service trap
  ret  0x2C        ;pop parameters off stack
                   ;   and return to caller
```

System Call

- POSIX API
- Windows Win32 API
- ...

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

System Call Types

- Process control
  - Create process 、 Terminate process
  - Get process attributes 、 Set process attributes
- file manipulation
  - Create file,delete file,read,write
  - Get/set file attributes
- device management
  - Request device,release device,read,write
- socket
  - Open connection, accept connection, read msg, write msg, close connection
- information maintenance
  - Getting current date, os version, etc.,

System Call Cases

Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

System Call Cases

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Quiz

- Which of the following several instructions should be executed only in kernel mode?
  - A. mask all interrupts
  - B. read current date
  - C. set current date
  - D. write the image core
  - E. read memory in user address space
  - F. halt

Ontogeny Recapitulates Phylogeny

- Dawrin, On the Origin of the Species
- The development of an embryo (ontogeny, 胚胎) **repeats** the evolution of the species (phylogeny)
  - Large Memories
  - Protection Hardware
  - Disk
  - Virtual Memory

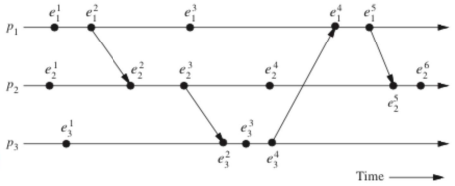
Functions of OS

- Process Management
- Memory Management
- Device Management
- File System Management
- User Interface
  - CLI
  - GUI
  - API
- Job Management

Characters of OS

OS Characters

- Concurrency
  - Concurrency: Logical concurrency
  - Parallel: Physical concurrency
- Share
  - CPU, Main Memory, Storage, I/O Devices
  - Space, Time
- Virtualization
  - 1 to N
  - N to 1
  - 0 to N
- Asynchronism



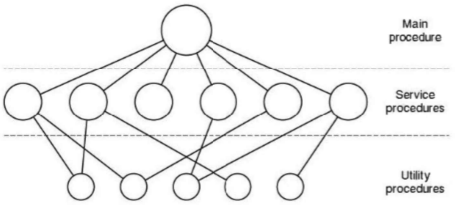
UNIX 操作系统有感

造化阴阳自动机，
体强神弱几多悲。
一朝撼木蚍蜉叹，
跬步移山智叟疑。
万类归一文件树，
千流模化进程池。
当年雏蛋今朝祖，
成败开源事后知。

OS Runtime Structure

OS Runtime Structure

- Monolithic Systems
  - A main program that invokes the requested service procedure.
  - A set of service procedures that carry out the system calls.
  - A set of utility procedures that help the service procedures.



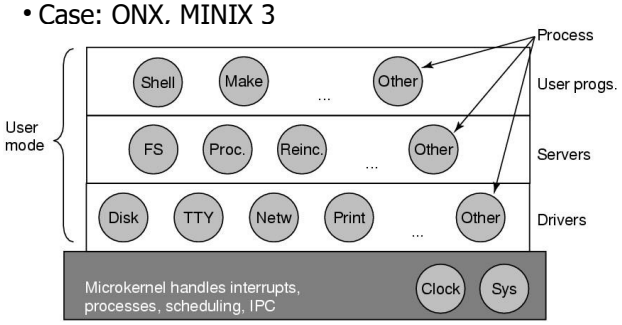
OS Runtime Structure

- Layered Systems
  - Case: THE

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

OS Runtime Structure

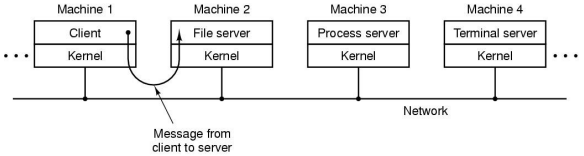
- Microkernels
  - Case: ONX. MINIX 3



Structure of the MINIX 3 system.

OS Runtime Structure

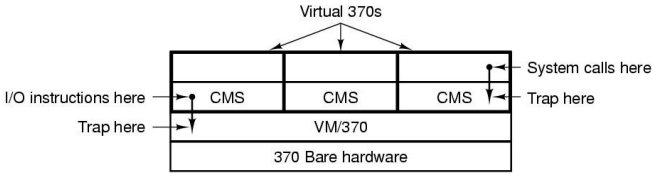
- Client-Server Model
  - Communication between clients and servers is often by message passing



The client-server model over a network.

OS Runtime Structure

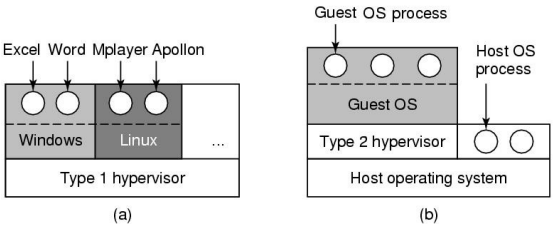
- Virtual Machines
  - Case: IBM's VM370, 1979



The structure of VM/370 with CMS

OS Runtime Structure

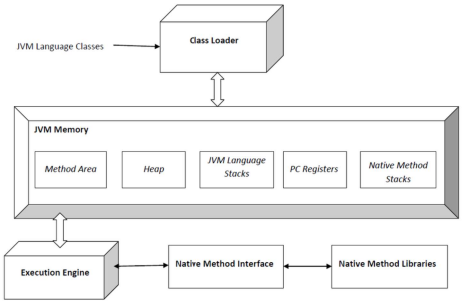
- Virtual Machines
  - Hypervisor I, Hypervisor II



(a) A type 1 hypervisor. (b) A type 2 hypervisor

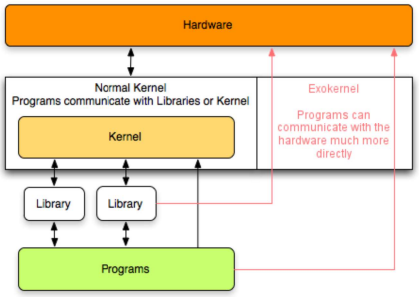
OS Runtime Structure

- The Java Virtual Machine
- Windows .Net Platform



OS Runtime Structure

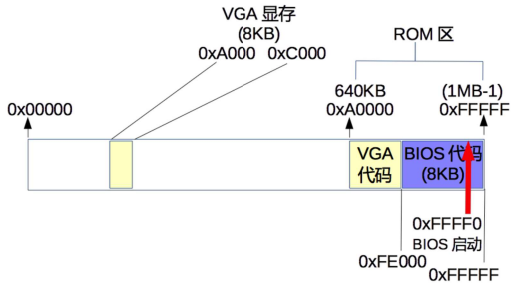
- Exokernels
  - Partitioning the actual machine, rather than cloning the catual machine
  - developed by the MIT Parallel and Distributed Operating Systems group



Booting The Computer

Memory Layout 1

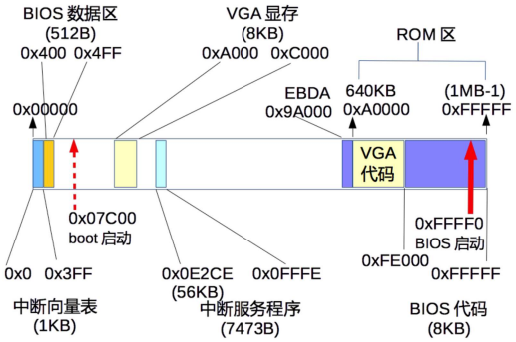
Memory Layout 2



Booting The Computer

- an IBM-compatible personal computer's x86 CPU executes
  - Power On, real mode
    - the instruction located at reset vector (the physical memory address FFFF0h on 16-bit x86 processors and FFFFFFF0h on 32-bit and 64-bit x86 processors, i.e. BIOS entry point
  - BIOS: POST, power-on self-test
  - BIOS: goes through a pre-configured list of non-volatile storage devices ("boot device sequence") until it finds one that is bootable
  - BIOS:load the bootstrap (i.e. MBR, Master Boot Record) from bootable storage device
  - MBR:load the OS Kernel
  - OS Kernel: OS services and shell

Memory Layout 3



Booting The Computer

- bootstrap

0x000~0x002 <A jump instruction to 0xttt>

0x003~… Disk parameters(used by BIOS)

0xttt~0x1fd Bootstrap program

0x1ff~0x1fe 0xaa55

Booting The Computer

- Harddisk Partition table

	00h	01h	02h03h		04h	05h	06h07h		08h~0Bh	0Ch~0Fh
01BEh	BI	Hs	Ss	Cs	SI	He	Se	Ce	HS	N
01CEh	BI	Hs	Ss	Cs	SI	He	Se	Ce	HS	N
01DEh	BI	Hs	Ss	Cs	SI	He	Se	Ce	HS	N
01EEh	BI	Hs	Ss	Cs	SI	He	Se	Ce	HS	N

SI:
00h undefined,01h Dos (12bit),02h XENIX,
04hDos (16bits),
05h extended partition,
06h Dos (32bits)

File System Types

```
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extended  c7  Syrinx
 5  Extended        41  PPC PReP Boot   86  NTFS volume set da  Non-FS data
 6  FAT16           42  SFS             87  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            a0  IBM Thinkpad hi eb  BeOS fs
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         ee  GPT
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD         ef  EFI (FAT-12/16/
10  OPUS            55  EZ-Drive        a7  NeXTSTEP        f0  Linux/PA-RISC b
11  Hidden FAT12    56  Golden Bow      a8  Darwin UFS      f1  SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD          f4  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       ab  Darwin boot     f2  DOS secondary
16  Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      fb  VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 75  PC/IX           be  Solaris boot    ff  BBT
1e  Hidden W95 FAT1 80  Old Minix
```

Metric Unit

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Research On OS

- Computer Science
- Internet
- GUI: Doug Engelbart
- Hot topics
  - Security, engergy, recovery, virtualization, fs, multicore,...
- ACM
  - www.acm.org
- IEEE Computer Society
  - www.computer.org
- USENIX
  - www.usenix.org

Summary

Q&A?

Assignments

- What is the purpose of a system call in an operating system?
- What is spooling? Do you think that advanced personal computers will have spooling as a standard feature in the future?
- What is the key difference between a trap and an interrupt?
- Translate the following paper into chinese
  - D. R. Engler, M. F. Kaashoek, J. O'Toole Jr, and J. O'Toole Jr, "Exokernel: an operating system architecture for application-level resource management," ACM SIGOPS Operating Systems Review, vol. 29, no. 5, pp. 251–266, Dec. 1995.