Introduction : Overview and History

Philosophical Principles of Operating Systems

In Pursuit of Absolute Simplicity 求于至简，归于永恒

Content

- What is operating system
- Why study operating system?
- History of operating system

A Wise Man's Answer

- Once upon a time, a boy asked a wise man:
  - Can you tell if the bird I am holding behind is alive?
- If the answer is "yes"
  - The boy will crunch the bird and show the dead bird
- If the answer is "no"
  - The boy will let the bird fly and laugh at the man

A Wise Man's Answer

- But the wise man would have no such foolishness
- What is the wise man's answer?
  - "As you will"

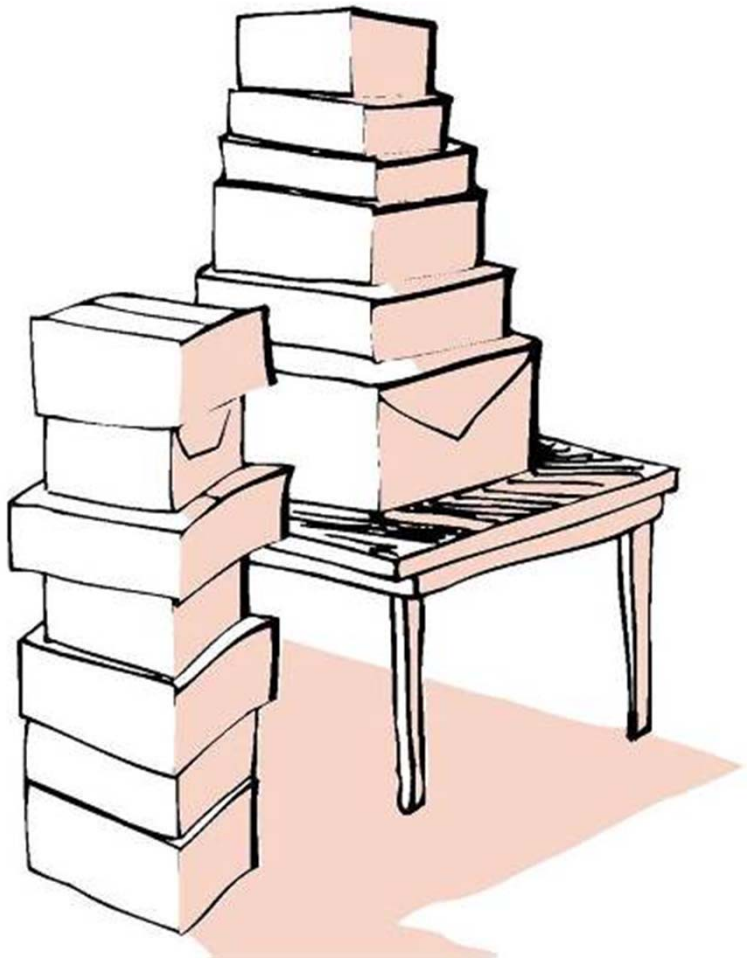
Philosophy of Computer

- What is the defining property of computer?
  - A calculating machine?
  - A computation device?
- An artifact
  - A human creation (vs. God creation)
- Computer science is therefore an artifact science

Artifact vs. True Science

- Artifact science:
  - Inexact, relative
  - Often derive from observation of human activities
  - Subject to human discretion
- True science:
  - Exact, absolute, no discretion is allowed
  - Derive from study of nature (God-made)
  - E.g. mathematics, physics, chemistry

Example of Artifact Science



Stack

Example of Artifact Science

Queue

Example of True Science

$$E=MC^2$$

Computer Science as Artifact

- Often there is no absolute right or wrong
  - Only better or worse
- There take what I teach with a grain of salt
  - Think creatively
  - Your solution may be better than mine

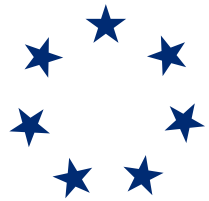
The Three Questions

- What is an operating system?
- Main functions of the OS?
- Why study OS?

What Is an Operating System?

- Something very important
  - Really not much help
- A software layer between hardware and applications

What Is an Operating System?



Application Programs

Virtual machine interface

Operating System

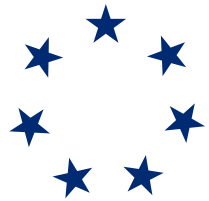
Physical machine interface

Hardware

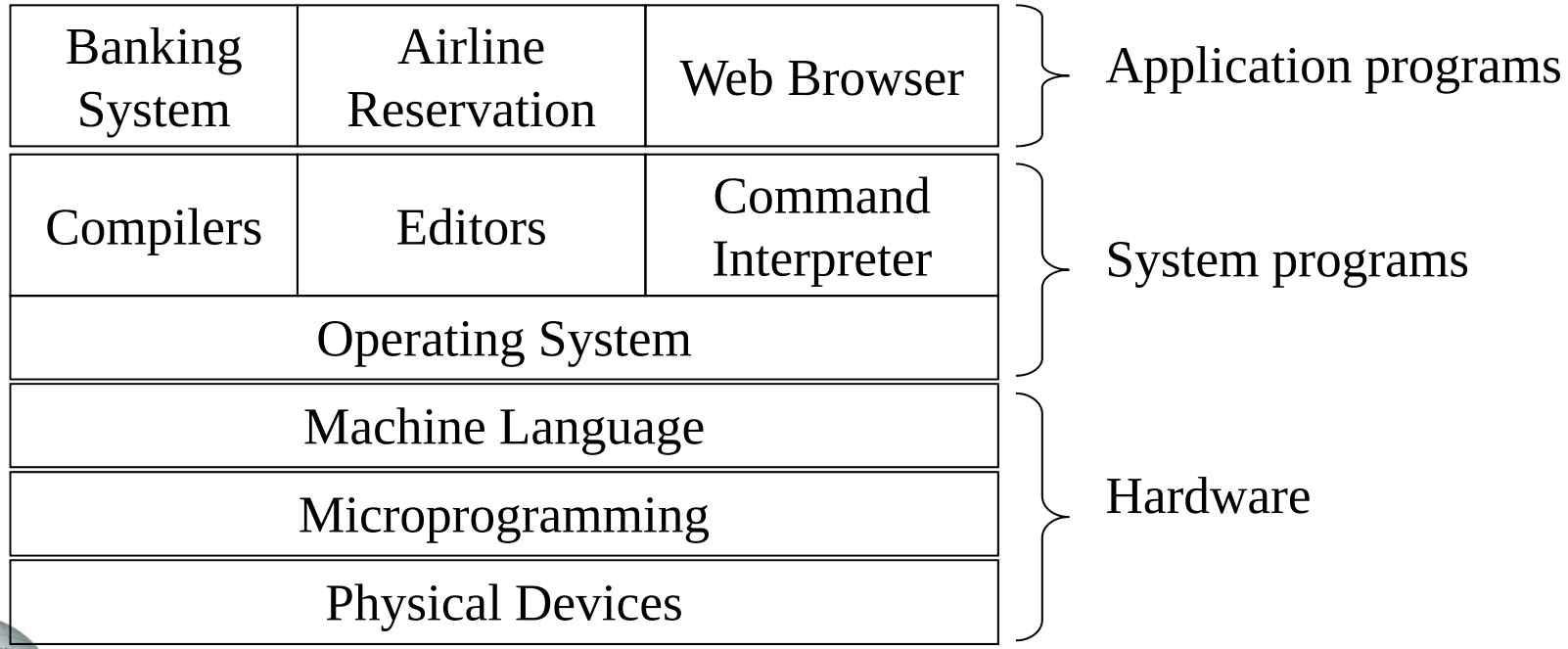
What Is an Operating System?

- A program that
  - manages the hardware resources and
  - makes it easier to use by application
- by presenting nicer abstractions
  - i.e. the virtual machine

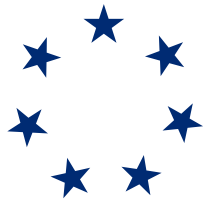
What Is an Operating System?



- OS hides complexity of underlying hardware
- Layered architectures



What Is an Operating System?



- For any area of OS, ask what interface does:
- The hardware present to the software
  - physical reality
- The OS present to its applications
  - the nicer abstraction

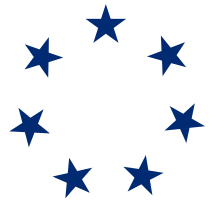
Example Abstractions

- Threads:
  - user doesn’t have to worry about sharing CPU
- Addresses spaces:
  - user doesn’t have to worry about sharing physical memory
  - or physical memory size

Example Abstractions

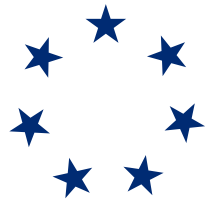
- TCP:
  - user doesn't have to worry about the unreliable network
- Device drivers:
  - frees the user (the rest of the OS) from worrying about differences in devices
  - e.g. the device registers used to control the device

Operating System Areas



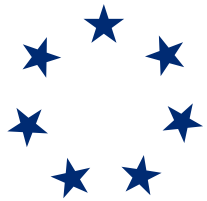
- Process (process/thread management)
- Virtual memory (memory management)
- File systems (storage management)
- Networking (communication management)
- I/O (input and output management)

Relationship: User Programs and OS



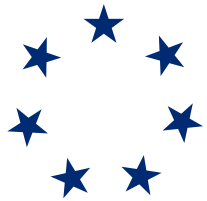
- First perspective:
  - User program is the main program, and
  - Gets services by calling the kernel
- Second perspective:
  - OS is the main program, and
  - calls user programs as subroutines

Two Main Functions of an OS



- Illusionist:
- Makes computer appear to be more or better
  - than it really is
- Examples?

Two Main Functions of an OS

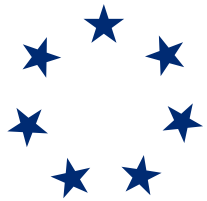


- Government:
- Parcels out single shared hardware resource to
  - multiple competing applications efficiently and fairly
- What hardware resources does OS manage?

Tasks of an Operating System

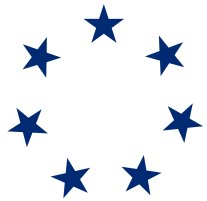
- Processor management
  - Scheduling
- Memory management
  - Virtual memory
- Storage management
  - File system
- Device management
  - Input-Output
- Batch processing- Background processing

Why Study Operating Systems?



- The functions of an OS appear in lots of domains
  - e.g. web server farm, concurrent applications
  - Distributed systems, networking
- Lots of techniques in OS are used in other domains
  - Designing and implementing an abstraction
  - Caching, indirection, concurrency, authentication,
  - naming, atomicity, resource allocation

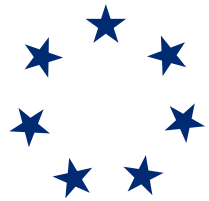
Why Study Operating Systems?



- Fun to “open the hood” and understand:
  - how the thing you’ve been using really works
  - Applicable only to those inquisitive minds
- May not at all be fun to many people!

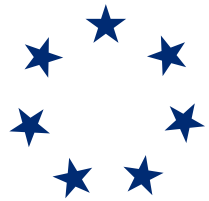
Operating System History

History of Operating Systems



- OS history is dominated by two trends
- Decreasing hardware cost
  - But hardware started out VERY expensive,
- Increasing complexity of OS

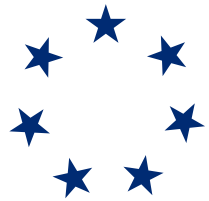
Single Operator at Console



- For first generation computer
  - ENIAC
- Goal: basic functionality
- Characteristics: interactive
- Time: 1940s (1946)

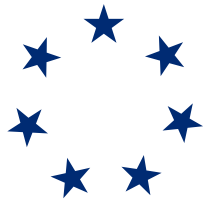
- Some people believe the first electronic computer is not ENIAC
  - Perhaps some military machine made during WWII

Single Operator at Console



- OS is simple:
  - one thing happening on the computer at a time
- OS is just a library of standard services
- Drawbacks:
  - Poor utilization of computer
  - Idle while person thinks and while I/O device works

Batch Processing



- For second generation of computers
  - IBM 1401, IBM 7094, etc.
- Goal: improve CPU and I/O utilization
  - by removing user interaction
- Example: FMS, IBSYS, UMES
  - FMS: FORTRAN Monitor System for the IBM 709
  - IBSYS: tape based OS for IBM 7090 and IBM 7094
  - MAD/UMES: University of Michigan Executive System
- Time: 1950s (1959)

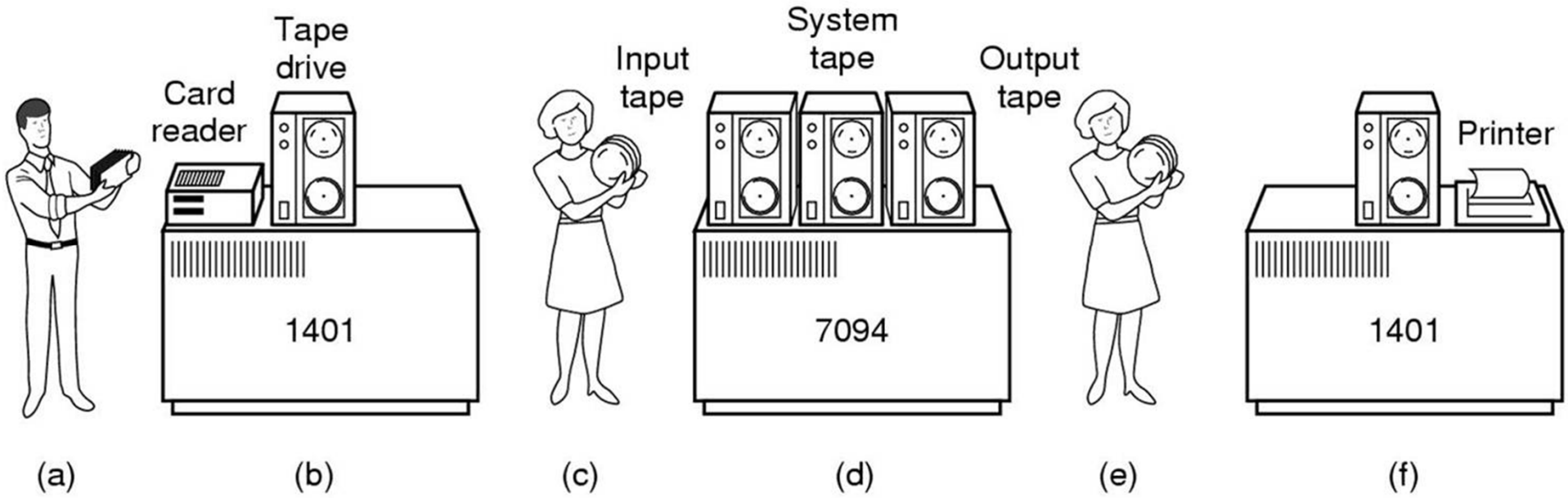
Batch Processing

- Both UM and MIT are owners of 7094
  - The most advanced computer at the time
- UM developed MAD language and UMES Operating Systems
  - R.M. Graham, Bruce Arden and Bernard Galler
- MAD/UMES was installed on MIT's 7094 computer
- R.M. Graham is one of the architect of Multics

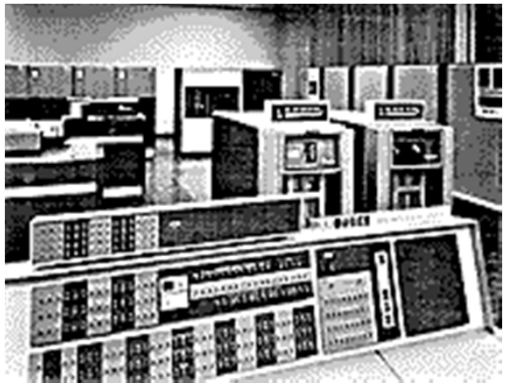
Batch Processing

- Submit job and wait for final answer
  - Use job cards, etc.
- No human interaction during execution

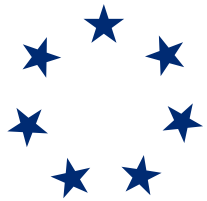
Batch Processing



- Bring cards to 1401
- Read cards to tape
- Put tape on 7094 which does computing
- Put tape on 1401 which prints output

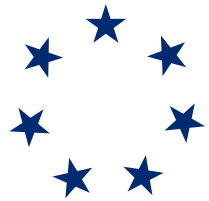


Batch Processing



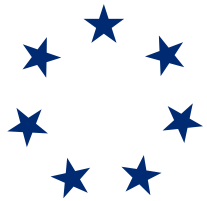
- Still only one job at a time
- OS is batch monitor & library of standard services
- Management of files, I/O devices, secondary storage
- Batch monitor loads program,
  - runs it, prints results, and starts next program

Batch Processing



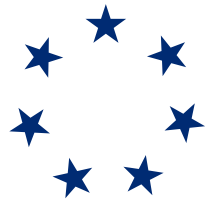
- Protection starts to become an issue
  - Must ensure that batch monitor can run next program in queue
- Why wasn’t this an issue for “single operator at console”?

Multi-programmed Batch



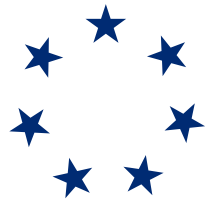
- For third generation of computers
  - System/360, 370, 4300, etc.
- Goal: improve CPU and I/O utilization
  - by overlapping CPU and I/O
- Example: OS/360(M)
  - For new generation and architecture of System/360 hardware - supporting both commercial and scientific applications
  - A powerful and very reliable OS developed by UM for IBM
- Time: 1960s (1964)

Multi-programmed Batch



- Overlap disk I/O and CPU
  - Resource managment and sharing for multiple programs
- Allows multiple I/Os to happen simultaneously
- Allows CPU and disk to work simultaneously
  - Quasi-simultaneous program execution

Multi-programmed Batch

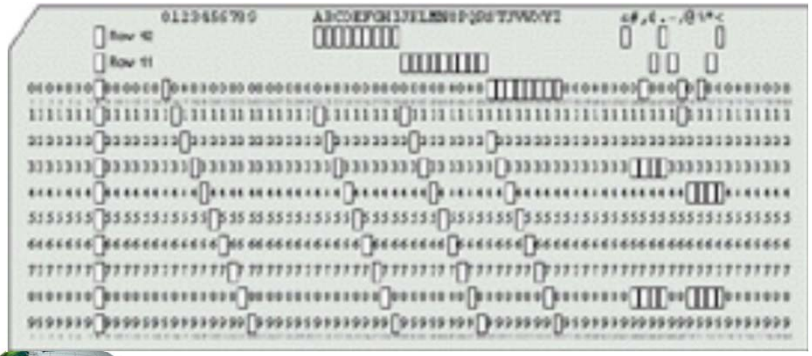


- Single user
- OS getting more complex:
  - OS switches between multiple processes
  - manages multiple I/O devices
- OS needs to protect processes from each other

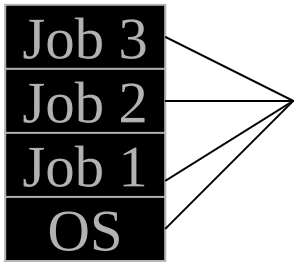
Batch Process

The elements of the basic IBM 1401 system are the 1401 Processing Unit, 1402 Card Read-Punch, and 1403 Printer.

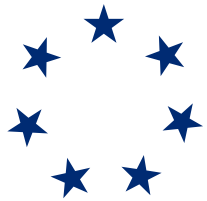
Punching cards



Multiprocessing programming

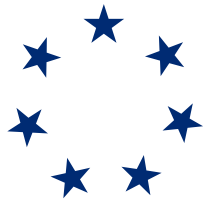


Time Sharing



- For third generation of computers or beyond
  - PDP, VAX, CRAY
- Goal: restore ability for humans to interact with programs
  - insight: human can be modeled as a I/O device
- Example:
  - CTSS: Compatible Time-Sharing System
  - MULTICS, derived from UMES
  - UNIX, derived from MULTICS

Time Sharing

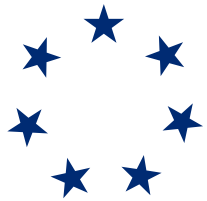


- Mgmt of multiple simultaneous users interconnected via terminals
  - Switch between programs when waiting for users
  - Switch back in time to get the next keystroke
- Fair resource management:
  - CPU scheduling, spooling, mutual exclusion
- OS getting more complex:
- Lots of simultaneous jobs
- Multiple sources of new jobs

Real-Time Systems

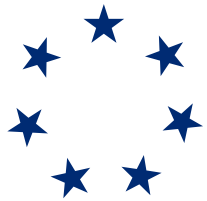
- Some time called process control systems
- Management of time-critical processes
- High requirements with respect to:
  - reliability and availability

Personal Computing



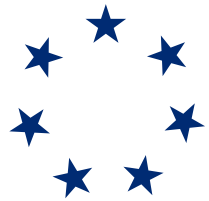
- For fourth generation of computers
  - PC, VAX, Workstation, etc.
- Computer hardware gets cheaper
  - economically feasible to go back to single operator
- OS re-becomes a subroutine library
- Example: DOS, Windows, Unix, Linux
- Don’t need to time-share between multiple simultaneous jobs?
- Don’t need protection?
- PC OS gradually added back most features from time-sharing

Distributed Computing



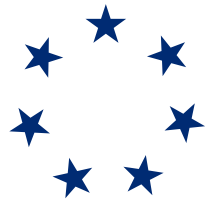
- For networked generation of computing
- Goal: make multiple computers look like one
- Networking operating systems
  - Traditional OS with network interface management
  - Novell Netware
- Distributed operating systems
  - New type of OS with multiple computers in mind

Cloud Computing



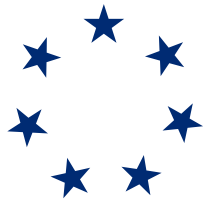
- For massive scale inter-networked generation of computing
- Goal: putting computing power on tap
- Cloud operating systems
  - Manage massive amount of resources
  - Load balance, computing/storage distribution, fault tolerance
- Typical Examples:
  - EMC, Google, Hadoop, Microsoft

Cloud Computing



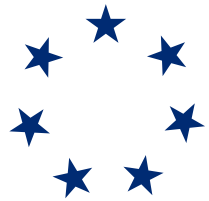
- EMC:
  - MPFS, Vmware, Engenity, DART
- Google:
  - GFS, BigTable, Map/Reduce, Borg, Chubby
- Hadoop,:
  - HDFS, Map/Reduce, ZooKeeper, HBase
- Microsoft:
  - Windows Azure, SQL Azure, Windows Azure AppFabric
  - Derive from Microsoft clustering

Mobile Computing



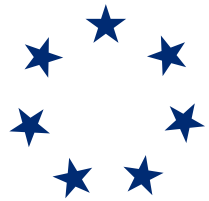
- Google:
  - Android
- Apple:
  - iOS
- Microsoft:
  - Windows Phone 7
- Others:
  - Symbian

Classification of OS



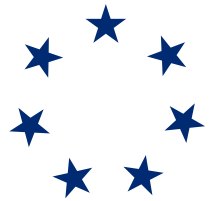
- OS for mainframe computers
  - OS/360, OS/390, CTSS
- OS for servers
  - UNIX, Windows 2000, Linux
- OS for multi-CPU computer
  - Novell Netware
- OS for cloud computing

Classification of OS



- OS for personal computers
  - Windows 98, Windows 2000, Windows XP, MacOS
- OS for real-time systems
  - VxWorks (WindRiver), DART (EMC)
- OS for embedded systems
  - Palm OS, Windows CE, TOPPER
- OS for online computing systems
  - Google Chrome, Web 2.0

Same Machine, Different OS

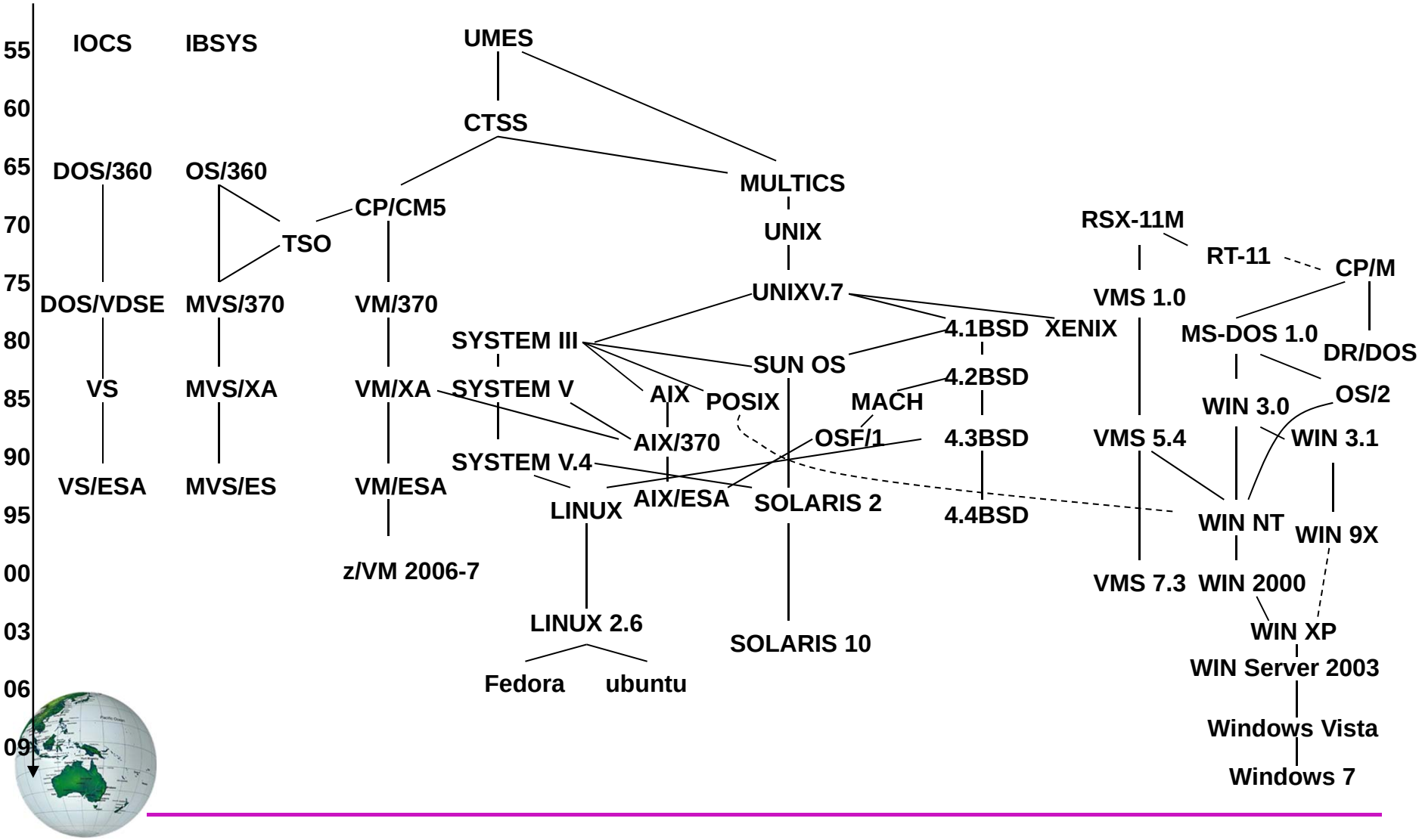


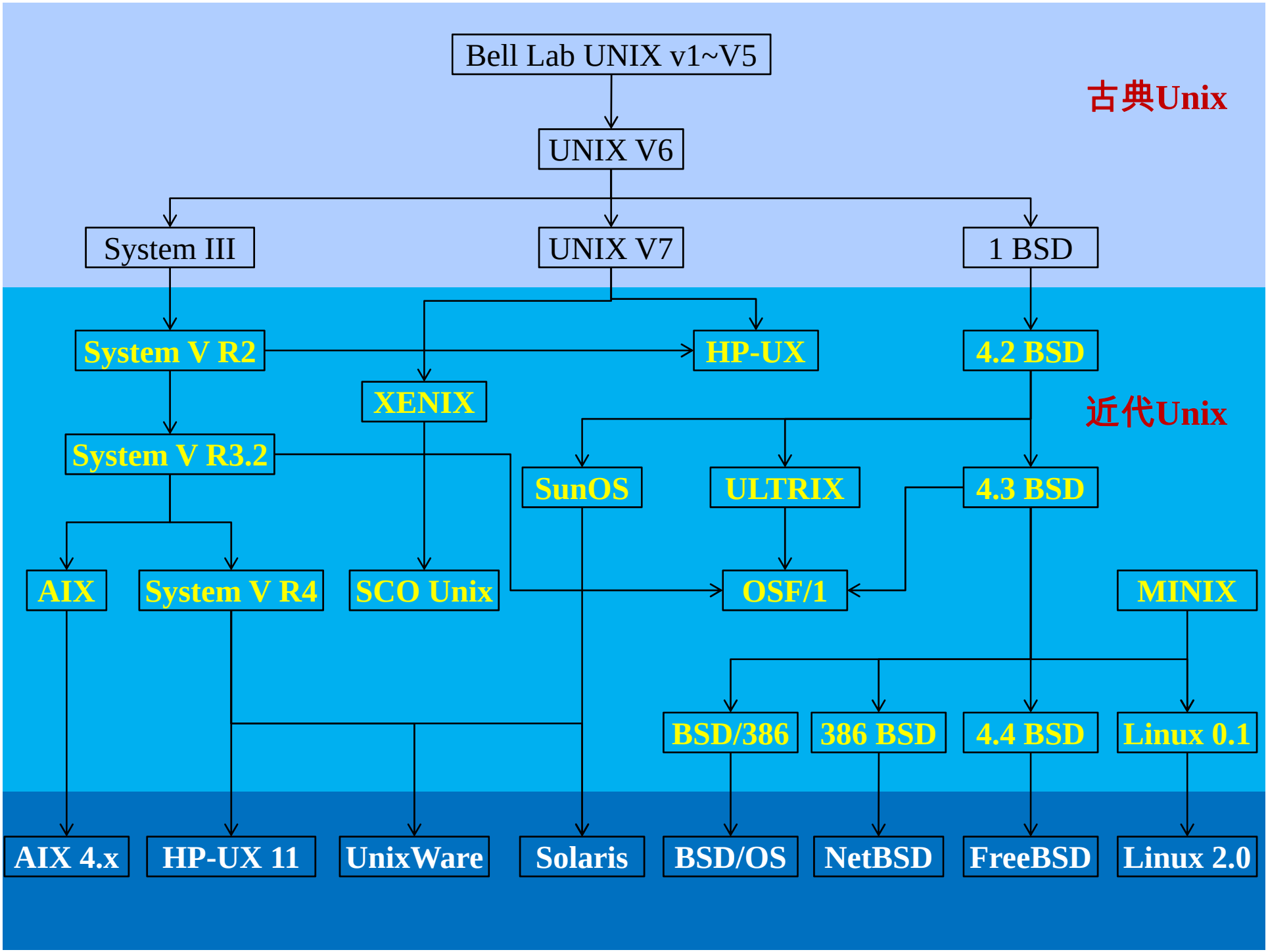
- IBM PC:
  - DOS, Linux, NeXTSTEP, Windows NT, SCO Unix
- DEC VAX:
  - VMS, Ultrix-32, 4.3 BSD UNIX

Same OS, Different Machines

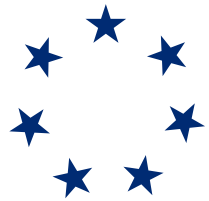
- UNIX
  - PC (XENIX 286, APPLE A/UX)
  - CRAY-Y/MP (UNICOS - AT&T Sys V)
  - IBM 360/370 (Amdahl UNIX UTS/580, IBM UNIX AIX/ESA)
- Windows XP (or Windows NT/2000)
  - Intel i386 (i486 an NT 4.0), Alpha (DEC)
  - PowerPC (MOTO)
  - MIPS (MIPS computer), Itanium (Intel)

Operating Systems Evolution



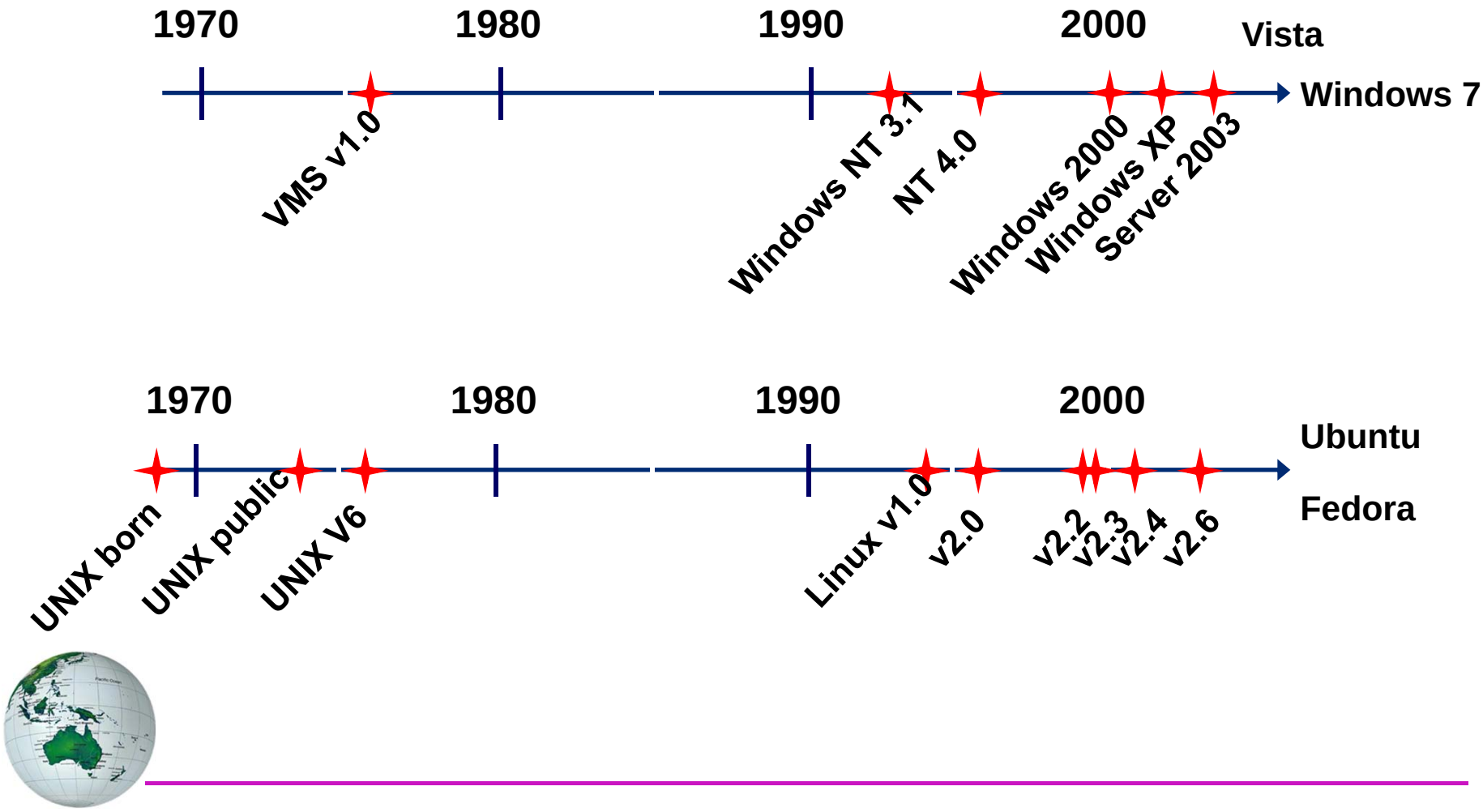

Bell Lab UNIX v1~V5
古典Unix
UNIX V6
System III
UNIX V7
1 BSD
System V R2
HP-UX
4.2 BSD
XENIX
近代Unix
System V R3.2
SunOS
ULTRIX
4.3 BSD
AIX
System V R4
SCO Unix
OSF/1
MINIX
BSD/386
386 BSD
4.4 BSD
Linux 0.1
AIX 4.x
HP-UX 11
UnixWare
Solaris
BSD/OS
NetBSD
FreeBSD
Linux 2.0

Windows Release History

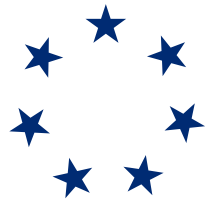


Build #	Version	Date
511	NT 3.1	July 1993
807	NT 3.5	September, 1994
1057	NT 3.51	May 1995
1381	NT 4.0	July 1996
2195	NT 5.0 (W2K)	December 1999
2600	NT 5.1 (XP)	August 2001
3790	NT 5.2 (Server 2003)	Mar 2003
4051	Longhorn PDC Preview	October 2003
	Windows Vista	2007
	Windows 7	2009

Windows And Linux Evolution

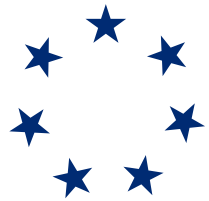


Sucks-Rules-O-Meter 02/16/2009

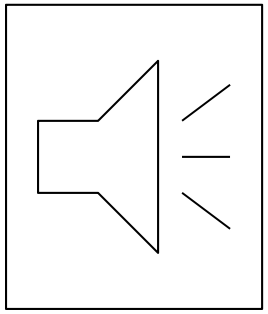


OS	反对票数	赞成票数	反对赞成比率
FreeBSD	673	9090	0.07
LINUX	213000	379000	0.56
MVS	63	168	0.38
MacOS X	35200	29900	1.18
NetBSD	257	654	0.39
Netware	66	135	0.49
OS/2	58	211	0.27
OS/400	10	16	0.63
OpenBSD	19800	781	25.35
SOLARIS	10600	1841	5.76
UNIX	14800	24200	0.61
VMS	103	48840	0.002
WINDOWS	433000	71500	6.05

Song: Every OS Sucks



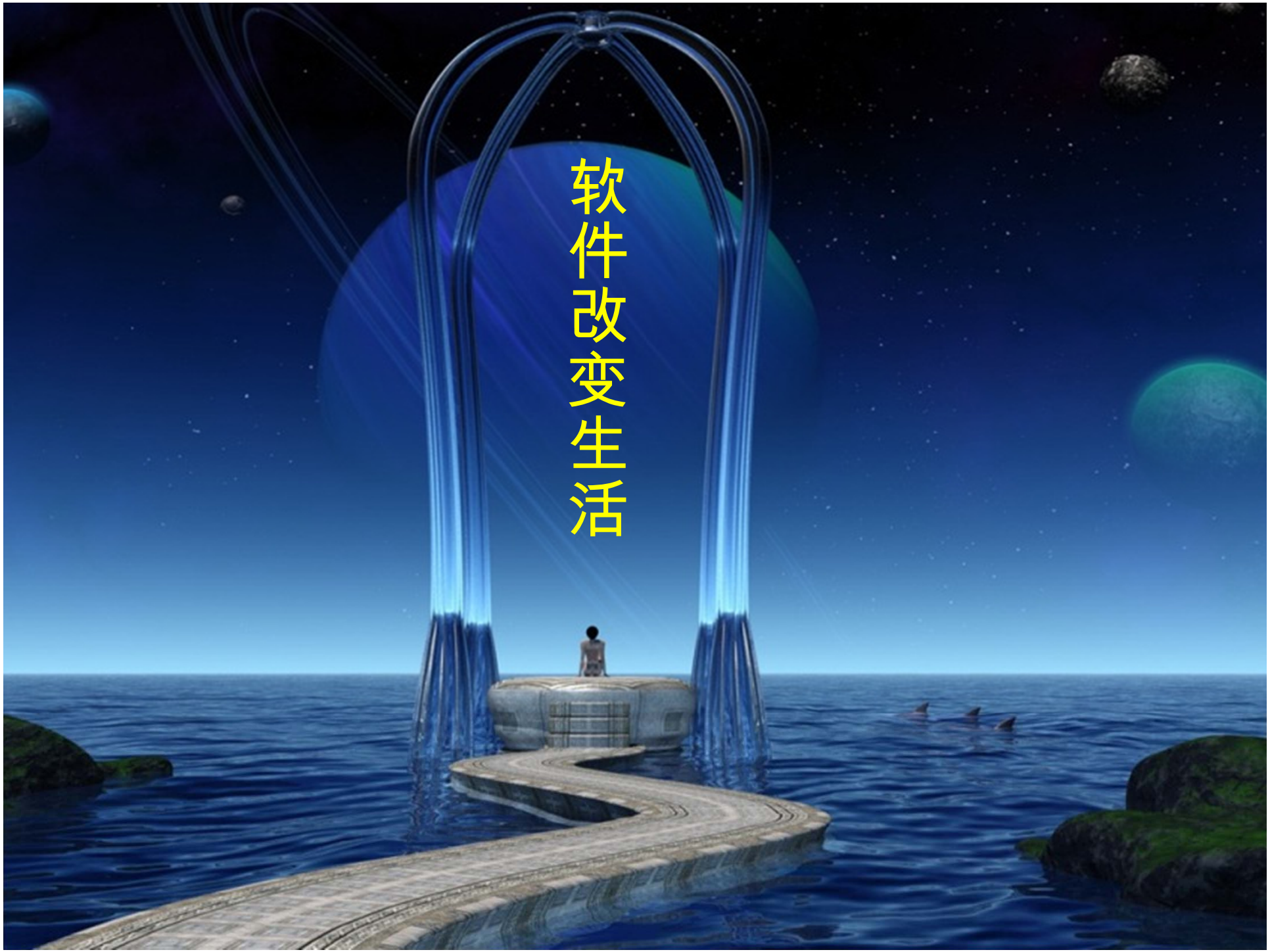
- Every OS wastes your time, from desktop to the lap
- Everything since the abacus, is just a bunch of crap
- From Microsoft to Macintosh, to Linux Linu-nux
- Every computer crashes, cause every OS sucks



Questions?

Then Ask!

- Office: 1309 Software Building
- Email: zou@sjtu.edu.cn
- URL: http://cse.sjtu.edu.cn/zou
- Skype: henmingway
- Work Phone: 3420-4934


软件改变生活