

Operating Systems Concepts

An introduction to the software that makes computers usable, ensures that systems can be made secure, and provides the environment for software to be a marketable commodity

Olav Beckmann

Huxley 449

<http://www.doc.ic.ac.uk/~ob3>

Acknowledgements: There are lots. See end of this chapter.

Home Page for the course:

<http://www.doc.ic.ac.uk/~ob3/Teaching/OperatingSystems/>

This is only up-to-date after I have issued printed version of the notes, tutorials, solutions etc.

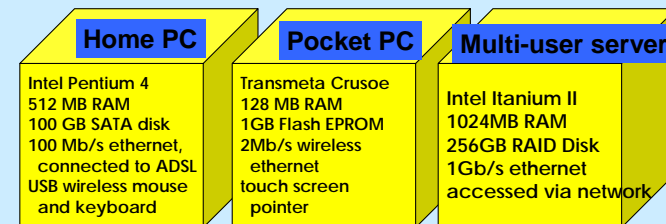
Chapter 1: Introduction

- What is an operating system?
- Why do I care?
- How can I pass this course anyway?
- Why does the lecturer seem to think this is “fun”?

What is an operating system?

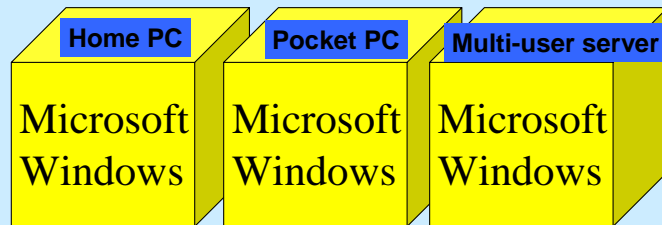
- The OS is software - it's a program
- It “virtualises” your computer
- Your computer needs to be virtualised...
 - Because that's what makes it possible to buy and sell computer programs
 - And that's a major economic activity

Why might I want a virtual machine?



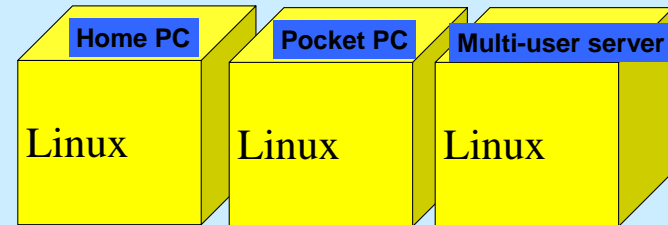
- Machines come in all shapes and sizes
- I want to sell software which can run on whatever machine the customer happens to have

Why might I want a virtual machine?



- Machines come in all shapes and sizes
- But they all run the same software
- The operating system makes each of the diverse physical machines behave like an idealised, “virtual” machine

The virtual machine as a standard platform



- The nice thing about standards is that there are so many to choose from...
- Not all operating systems are equally good
- Some are specially suited to particular purposes

Example: Diverse Hardware



- PowerPC G4 Processor
- Multiple device connections
- Portable environment
- Runs Mac OS X



- 1100 Nodes
- Each node has two 2.3 GHz PowerPC 970FX processors
- 14th most powerful computer in the world in June 2005 according to www.top500.org
- Nodes run Mac OS X

- What are the benefits of virtualising these machines?

Operating Systems as Virtual Machines

An operating system:

- Manages a system's **resources** so that they are used efficiently and safely, *e.g.*,
 - CPU(s)
 - memory
 - devices (modems, disks, network interfaces, video interfaces)
 - Q: Example where efficiency matters? Safety?
- Presents a **virtual machine that provides convenient abstractions**, *e.g.*,
 - files rather than disk locations (device independence)
 - inter-process synchronisation and communication.

Hardware + OS = Usable Virtual Machine

- Microsoft: Windows ME, Windows 2000, Windows CE
- Linux
- BSD (FreeBSD, NetBSD)
- Sun: Solaris
- IBM: AIX
- Compaq: Tru64 Unix, OpenVMS
- Hewlett-Packard: HP-UX
- Apple: MacOS-8, MacOS-X
- Symbian: EPOC
- PalmOS
- QNX
- VXWorks
- LynxOS
- MVS, AS/400
- Many many more, mostly more-or-less Unix/POSIX compatible
- Literally thousands of variants and research prototypes
- Sun (again): the Java Virtual Machine

There are lots...

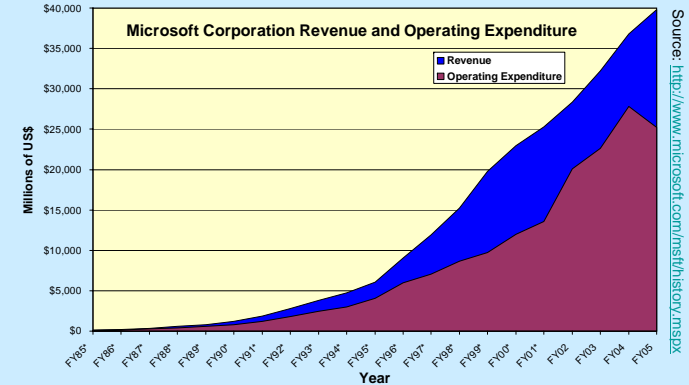
Various kinds of Unix

Designed for handhelds

Embedded real-time

Opening up the software market

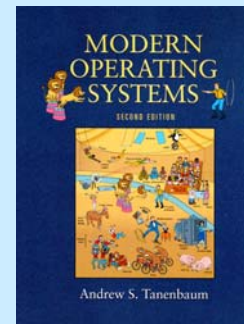
- More than 90% of all software products sold are designed for just one virtual machine



This course

- What is an OS?
- What terminology is involved in understanding OS issues?
- What are the different kinds of OS?
- How does an OS work?
- What principles underly OS operation?
- What does an OS look like “under the bonnet”?
- How can I find out more?

Recommended textbook



Modern Operating Systems
(2nd Ed),
Andrew S Tanenbaum
Vrije Universiteit Amsterdam

((Andrew Tanenbaum is also behind <http://www.electoral-vote.com/>, for those who are interested in US Politics...))

This course is not a substitute for reading the book.

Other books:

- Operating Systems, 3RD edition, Gary Nutt (Addison Wesley)
- Operating System Concepts, 6th edition, Silberschatz, Galvin and Gagne

Read books

- The point of this course is to help you become familiar with a classic, authoritative textbook
- The exam is designed to test your understanding and ability to apply it to new ideas
- Some exam questions will introduce an idea not covered in the course
- It will ask you to explore the idea using what you have learned
- Of course if you read a good book, you might find you already know all about it!

History of Operating Systems

* Material on History slides from Tanenbaum book website

- 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

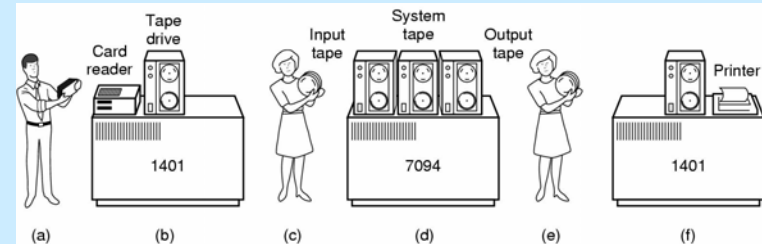
What did the first operating system look like?

1945 to 1955

- Vacuum tubes and plug boards
- No operating system
- Human operators



History of Operating Systems



Early batch system

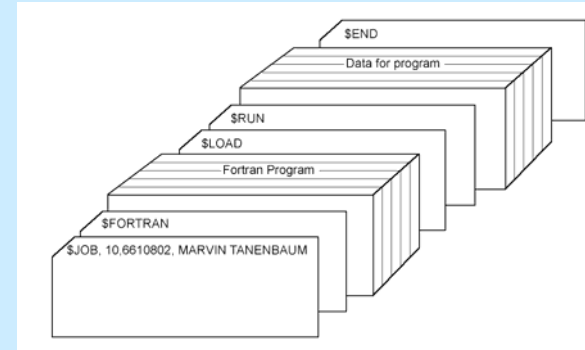
- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

1956 to 1965

- Transistors and batch systems
- Clear distinction between designers, builders, operators, programmers, and maintenance personnel
- I/O channel
- Read ahead / spooling
- Interrupts/ exceptions
- Minimal protection
- Libraries / JCL



History of Operating Systems



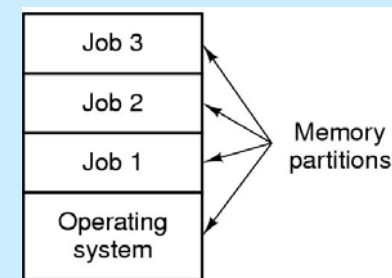
- Structure of a typical FMS job – 2nd generation

1965 to 1980

- ICs and Multiprogramming,
- System 360 and S/370 family of computers,
- Spooling (simultaneous peripheral operation on-line),
- Time sharing, On-line storage for System programs,
- User programs and data, Program libraries,
- Virtual memory,
- Multiprocessor configurations e.g. **MULTICS**



History of Operating Systems (4)



- Multiprogramming system
– three jobs in memory – 3rd generation

1980-now

- Personal computers and workstations
- MS-DOS and Unix
- Massively parallel systems
- Pipelining
- Array processing / SIMD
- General multiprocessing / MIMD
- Symmetric multiprocessing / SMD
- Any process and any thread can run on any available processor
- Computer networks (communication aspect) -- network operating systems
- Distributed computing -- distributed operating systems

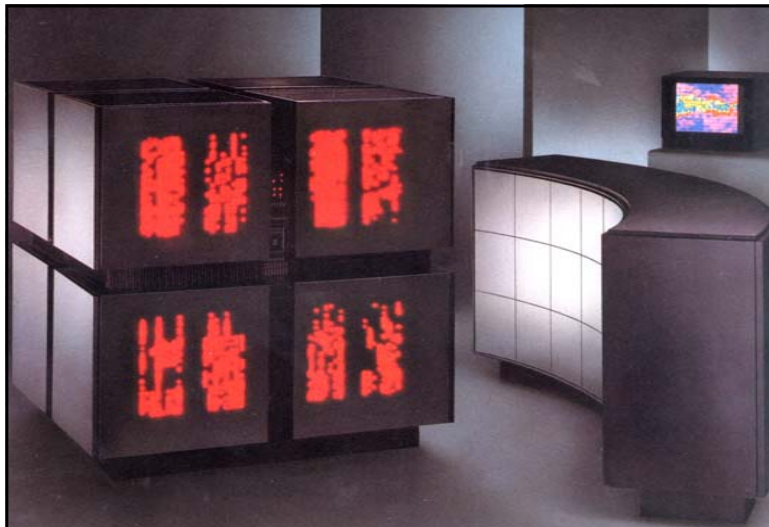


Cray 2



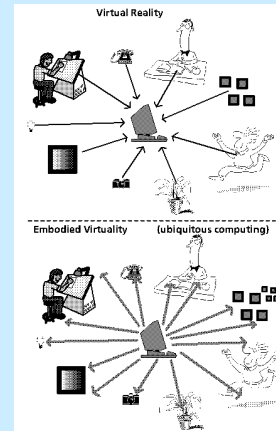
Photo taken from Dennis Ritchie's website

- Dennis Ritchie and Ken Thompson, originators of UNIX



Connection Machine 5

So what's the next (current?) wave?



Ubiquitous Computing...

- Involves many disciplines.
- Inspired by the social scientists, philosophers, and anthropologists
- paradigm shift?
 - currently we expect the user to find ways to use the computer
 - however we currently do not emphasise how the computer can find its own way to serve the user
 - focus on HCI
 - focus on security, privacy → big brother?

Current Technology



Laptop



Best friend

DoCoMo
video phone

(c) 2000 Juergen Spacht



Mobile phone / PDA



Personal digital
assistant (PDA)



Thanks to M. Sloman for this slide

Wearable



Wearable or
luggable?



LCD
Jacket



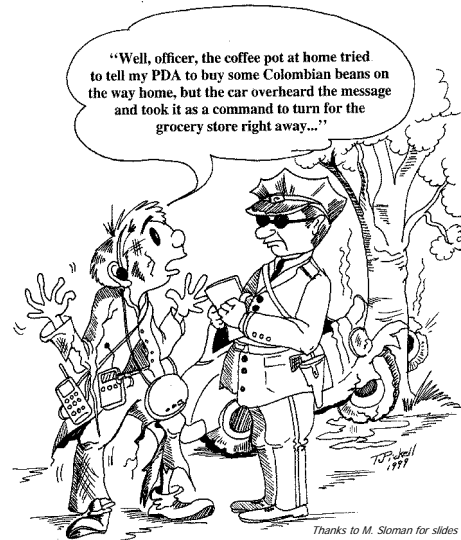
MITHril

Designer
Gear

Thanks to M. Sloman for this slide

Intelligent Environment

- Fridge and cupboards tracks consumption and reorder your groceries
- Your car computer reminds you to pick up your order on the way home when you are near the supermarket.



Thanks to M. Sloman for slides

Intelligent Environment - 2

- Lights, air conditioning, TV automatically switch on and off when you enter or leave rooms
- Sit on your favourite chair and TV switches on to the program you usually watch at this time of the day
- Use communicator/pda for phone, remote control, keys payments, passport, health records, authenticator.
- Route input from 'virtual' keyboard to nearest suitable display.
- *Automatic* detection of new items to control and physical layout in a room or office, using computer vision.

The Operating System Zoo

- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Real-time operating systems
- Embedded operating systems
- Smart card operating systems

What is an operating system?

- An OS is a program. Almost all Operating Systems are written in C or C++
- The job of an OS is to load and run other programs
- These “application” programs have to be able to
 - draw on the screen
 - interact via keyboard, mouse etc
 - access the hard disk (access your files)
 - communicate with other application programs
- The OS should provide a consistent way to do this, which works on whatever hardware you have

An OS provides a virtual machine

- Provides a “virtual” machine for applications to run on
 - which provides consistent interface to devices and services
 - so when you sell your software in the supermarket, it should work on all potential customers’ machines
- Is there a limit? What might make it difficult?

Challenges to the VM approach

- If the machine is too slow to run the application, there is nothing the OS can do to make that customer buy it
- What if the machine has too little memory?

Overcoming challenges to the VM approach

- **What if the machine has too little memory?**
 - Most OSs can solve this problem
 - Later in this course you will see how “virtual memory” works, by “paging” data to and from the hard disk
 - A program which needs 100MBytes of virtual memory can run on a machine with just 10MBytes of physical memory
 - But it might run rather slowly

Overcoming challenges to the VM approach

- What if the machine has too little memory?
 - Fixed with “virtual memory” and “paging”
- **What if the machine is already running lots of other application programs?**

Overcoming challenges to the VM approach

- What if the machine has too little memory?
 - Fixed with “virtual memory” and “paging”
- **What if the machine is already running lots of other application programs?**
 - We need the processor to switch from one “process” to another
 - regularly, using a timer to interrupt the processor
 - dividing its time fairly, according to each application’s needs
 - “time-slicing”, scheduling and priority; see later in course

Overcoming challenges to the VM approach

- What if the machine has too little memory?
 - Fixed with “virtual memory” and “paging”
- What if the machine is already running lots of other application programs?
 - Fixed with “interrupts”, “time-slicing”, scheduling, priority
- **What if some of those other programs behave maliciously, e.g. trying to steal your secrets?**
 - OS has privileged control over “address translation” hardware
 - each application allowed access only to own data and data its (authenticated) user is allowed to access

Overcoming challenges to the VM approach

- What if the machine has too little memory?
 - Fixed with “virtual memory” and “paging”
- What if the machine is already running lots of other application programs?
 - Fixed with “interrupts”, “time-slicing”, scheduling priority
- What if some other programs misbehave?
 - Fixed with privileged execution mode, address translation, authentication, access control
- What if two applications try to access the same device, such as a printer?
 - Fixed with “mutual exclusion”, but beware of deadlock!

Key OS terminology

- What if the machine has too little memory?
 - Fixed with **virtual memory** and **paging**
- What if the machine is already running lots of other application programs?
 - Fixed with **interrupts, time-slicing, scheduling, priority**
- What if some of those other programs misbehave
 - Fixed with **privileged execution mode, address translation, authentication, access control**
- What if two applications try to access the same device
 - Fixed with **mutual exclusion**, but beware of **deadlock!**

A well-known OS at work

This demonstration uses Windows XP

You can start the Task Manager application either by hitting Ctrl-Alt-Del, or via Start->Run->”taskmgr”

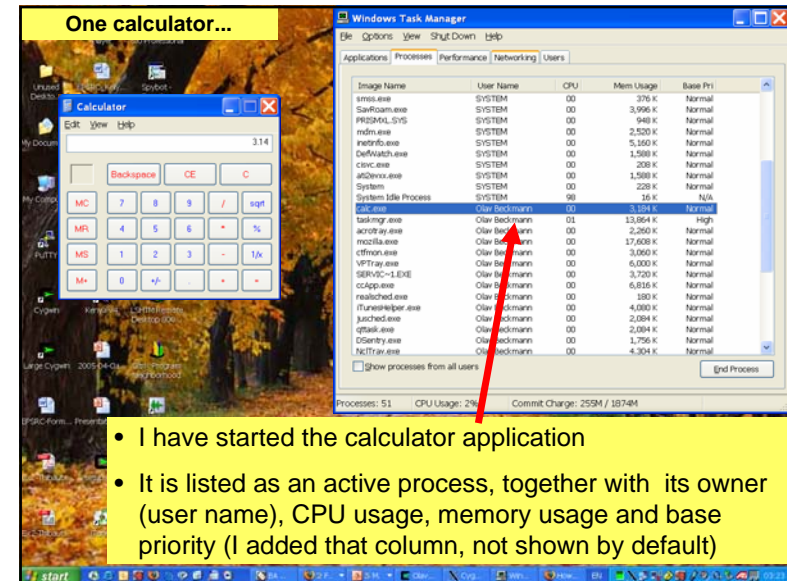
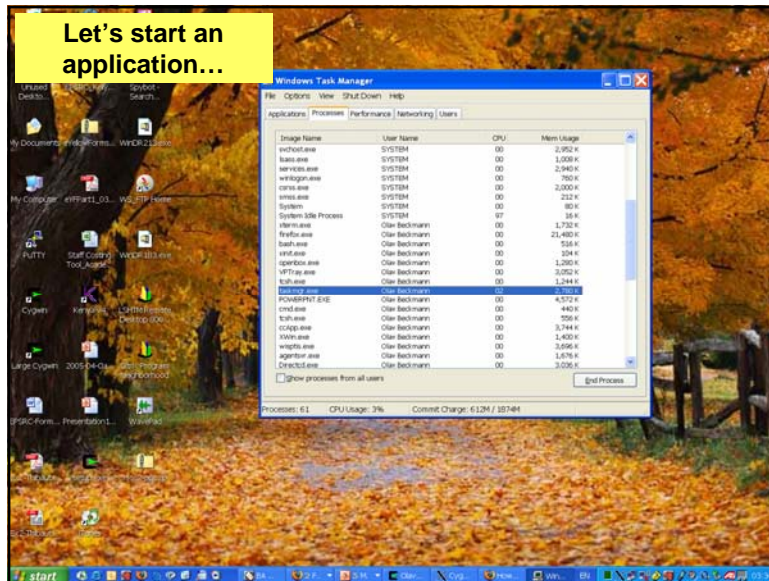
A closer look...

System processes

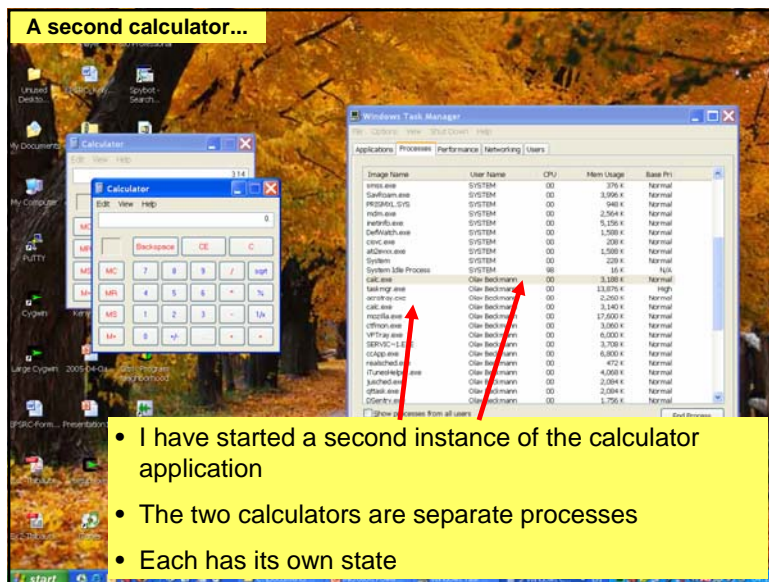
Image Name	User Name	CPU	Mem Usage
svchost.exe	SYSTEM	00	2,952 K
lsass.exe	SYSTEM	00	1,264 K
services.exe	SYSTEM	00	2,940 K
winlogon.exe	SYSTEM	00	760 K
csrss.exe	SYSTEM	01	1,992 K
smss.exe	SYSTEM	00	212 K
System	SYSTEM	02	80 K
System Idle Process	SYSTEM	59	16 K
xterm.exe	Olav Beckmann	00	1,732 K
firefox.exe	Olav Beckmann	00	21,420 K
bash.exe	Olav Beckmann	00	516 K
xinit.exe	Olav Beckmann	00	104 K
openbox.exe	Olav Beckmann	00	1,280 K
VFPTray.exe	Olav Beckmann	00	3,052 K
tchsh.exe	Olav Beckmann	00	1,244 K
taskmgr.exe	Olav Beckmann	03	2,756 K
POWERPNT.EXE	Olav Beckmann	00	2,980 K
cmd.exe	Olav Beckmann	00	440 K
tchsh.exe	Olav Beckmann	00	556 K
ccApp.exe	Olav Beckmann	00	...
XWin.exe	Olav Beckmann	00	...
wisptis.exe	Olav Beckmann	00	...
agentsvr.exe	Olav Beckmann	00	1,676 K
DirectD.exe	Olav Beckmann	00	3,036 K

Application processes

© Department of Computing, Imperial College London. Operating Systems Concepts (MSc Computing / JMC2) 2005/06 – Chapter 1 40



- I have started the calculator application
- It is listed as an active process, together with its owner (user name), CPU usage, memory usage and base priority (I added that column, not shown by default)



- I have started a second instance of the calculator application
- The two calculators are separate processes
- Each has its own state

Course web pages

- URL:
 - <http://www.doc.ic.ac.uk/~ob3/Teaching/OperatingSystemsConcepts/>
- What's there (or rather, what will be there):
 - on-line access to lecture notes to browse and print
 - on-line access to tutorial exercises and some solutions
 - links to other useful background OS material
 - past exam papers (though the course is being revised somewhat)

© Department of Computing, Imperial College London. Operating Systems Concepts (MSc Computing / JMC2) 2005/06 – Chapter 1 44

Concluding...why are OSs fun?

- Essential in making application software a commodity which can be marketed to a wide range of customers with diverse equipment
- Provide really useful abstractions which make programming easier
- Challenging: responsible for security, performance, reliability
- Mediates between application and hardware, so opportunities to manipulate running programs in interesting ways

Credits

- Julie McCann
- Paul Kelly
- William Knottenbelt
- Jeff Magee
- Jeff Kramer
- Kevin Twidle
- Steve Vickers
- Ariel Burton
- David Howarth
- Ken Thompson
- Marshall Kirk McKusick
- C A R Hoare
- E W Dijkstra
- Per Brinch Hansen
- William Stallings
- Richard Stevens
- Andrew Tanenbaum
- Philip Zimmerman
- Robert Zimmerman