Lecture 1 - Introduction

Karl R. Wilcox
Karl@cs.rhul.ac.uk

Objectives Of The Course

- To understand the purpose of operating systems
  - Why do we need them?
- To understand the theory of operating systems
  - How do they work?
- To understand the compromises inherent in operating system design
  - Why do they work that way?
- To identify commonalities between operating systems and other IT & engineering practices

Administration

- Lecturer – Karl R. Wilcox
  - No office, contact by e-mail karl@cs.rhul.ac.uk
- Timetable
  - Mondays 10-11 BLT1 Every Week
  - Wednesdays 11-12 NOT USED
  - Fridays 11-12 MFLT Most Weeks
  - On Monday – schedule for rest of week
- Resources
  - “Operating Systems: Design and Implementation”
    - Tanenbaum & Woodhull, Prentice Hall
  - Material on CS website
- Assessment
  - Final examination will count for 100% of coursemark

Topics

- Processes
- Inter-process communication
- Memory Management
- Virtual Memory
- Page replacement algorithms
- Page analysis
- IO (hardware)
- IO management
- Disk scheduling
- File management
- Directories & access
- + Some coursework

Why Study Operating Systems?

- The Traditional View
  - Computer Science degree have always had an operating systems course
- The Cynical View
  - There are lots of books available and the subject is easy to examine
- The Realistic View
  - Very few of us will ever develop operating systems BUT:
  - Understanding the how & why of OSs is useful
  - Many of the techniques, compromises and problems of OSs are applicable to other areas of IT & CS

What is an Operating System?

- An operating system performs two basically unrelated functions
- Virtual machine
  - From the point of view of the user - Topdown view
- Resource manager
  - From the point of view of the designer of the operating system - Bottomup view

Virtual Machine

- Using a controller chip for floppy disk IO we have commands eg READ and WRITE
- READ requires parameters to specify
  - address of the disk block to be read
  - number of sectors per track
  - recording mode physical medium
- The programmer wants a simple abstraction to deal with
- The operating system hides the truth about the hardware
- The operating system presents a simple file-oriented interface

Resource Manager

- The job of the operating system is to provide for a controlled allocation of processors memories and IO devices among the various programs which use them
- If a computer has various users we need to manage and protect resources
- The operating system has to know who is using which resource to grant requests to protect private les to mediate conflicting requests

History of Operating Systems

- The 1st generation (1945 – 1955)
  - Vacuum Tubes and Plugboards
- The second generation (1955 – 1965)
  - Transistors and Batch Systems
- The third generation (1965 – 1980)
  - ICs and Multiprogramming
- The fourth generation (1980 – now)
  - Personal Computers
- The clientserver model (late 80's – now)
  - Networks

The First Generation

- A single group of people designs builds programmes operates and maintains each machine
- The machines filled up entire rooms
- All programming was in absolute machine language Not even assembly language is known
- No operating systems yet
- The programmer would sign up for a block of time come to the machine room insert hisher plugboard into the computer and wait for the output

The Second Generation

- The computers are now reliable enough to be manufactured and sold
- Separation between designers, builders, operators programmers and maintenance personnel
- The machines were locked away in special rooms where professional operators ran them
- To run a **job** or program a programmer has to write it on paper transmit it on special cards and give them to the operator
- To reduce the wasted time the **batch system** was introduced
- The languages used were FORTRAN and assembly language
- The operating system were FMS (Fortran Monitor System) and IBSYS

The Third Generation

- The 1st series of software compatible machines were developed
- The machines differed only in price and performance but all had the same architecture and instruction set
- IBM introduced the System/360. It used small scale integrated circuits (ICs), not individual transistors any more
- Weakness of the “one family” idea: The operating system had to work in completely different situations
- **Multiprogramming** and **spooling** are introduced
- The principle of **timesharing** is implemented (CTSS, MULTICS)

The Fourth Generation

- Development of LSI (Large Scale Integration) made smaller and more powerful machines possible
- The most powerful ones are usually called **workstations**
- The software is user-friendly
- The computers used the Intel 8088 CPU (286, 386, 486, Pentium I/II/III)
- Operating systems: UNIX, MSDOS, Windows, MacOS etc. UNIX is used on nonIntel computers

Client-Server Model

- In modern operating systems code is moved up into higher layers and removed as much as possible from the operating system leaving a minimal kernel
- Most operating system functions are implemented as user processes
- To request a service a user process (=*client process*) sends the request to a *server process*
- The *server process* sends an answer back to the *client process*
- The kernel is only handling the communication between clients and servers
- OS is split up into parts
- All servers run in user mode and are not connected to the hardware

Further Reading

- For an excellent discussion of the history and philosophy behind the Operating Systems “debate” try:
- “In The Beginning Was The Command Line”
  - Neal Stephenson, Avonbooks (Try Amazon.co.uk)

- Remember:
  - No lecture on Wednesday
  - There IS a lecture on Friday