# Operating Systems

Figure 1:

*January 1997*

*Omer F. Rana*
*omer@cs.cf.ac.uk*

**http://www.cs.cf.ac.uk/User/O.F.Rana/**

# 1 How does one define computer science ?

- Is it science ?
- Is it engineering ?
- Is it mathematics ?

ALL ?!

- **The Analogy approach**
- The developer vs. the user
- **Pragmatic vs. Theoretic**

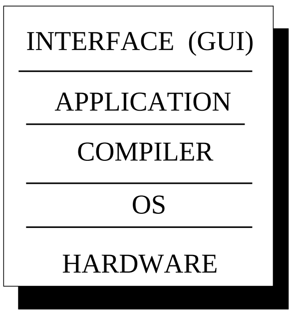


Figure 2: *Computing Systems Hierarchy*

## 2 High-level Abstraction

- Hiding unnecessary detail from the programmer
- Type of recording used for writing to disk and reading data - hidden from the programmer
- simple view of the file structure, not how the files are stored, but how the programmer (user) can access or change them

Hiding low level detail and presenting an interface for applications

*To present the user with the equivalent of an* **extended** *or* **virtual** *machine*

**Processes :** dedicated CPU.
**Virtual Memory :** Infinite RAM.
**File:** Data typed permanent memory (e.g sequential, random, associative access).
**Channel:** Data typed I/O device (e.g sequential, random, associative access).
**Shell :** programmable user interface.

## 3 The world of acronyms and abbreviations

- **DHCP**: Dynamic Host Configuration Protocol (Windows NT)
- **CORBA**: Common Object Request Broker Architecture (Unix/Sun Microsystems)
- **The Orange Book** : Security Protocol
- **SAM**: Security Accounts Manager (Windows NT)
- **RPC**: Remote Procedure Call (General)
- **HAL**: Hardware Abstraction Layer (General)
- and many more

**Welcome to the world of buzz words**

---

Because of operating systems being the **core** of any computer system, the field is over-flowing with buzz words, abbreviations and acronyms

**Concepts and Semantics**

## 4 Name me some Operating Systems ?

- **PCs** : DOS, Windows 3.11, Windows 95, Windows NT : **Microsoft**
- **W/station** : Solaris, SunOS 4.1.x, Linux, 386BSD, AIX, IRIX
- **Mac** : Apple System 7.5.x or Copland (System 8)
- **IBM** : OS/2 with **Microsoft**
- **IBM** : MVS : IBM390 mainframes - OS/390 new
- **Distributed** : CONIC, REX, Plan 9
- **Acorn** : RISC OS - Archimedes PCs - ARM based - replaced the earlier Arthur
- **Real time** : LynxOS from Lynx Real-Time systems
- **Time sharing** : Multics (Bell Labs, MIT, GE) *MULTiplexed Information and Computer Storage*
- **Network** : Novell NetWare
- **DEC** : VAX/VMS, TOPS-10/TOPS-20
- **Proprietary** : Angel : Multiprocessor
- **Lisp** : STING - high level OS for symbolic programming

## 5 What is an Operating System (OS) ?

The low-level software which schedules tasks, allocates storage, handles the interface to peripheral hardware and presents a default interface when no application program is running.

The OS may be split into a kernel which is always present and various system programs which use facilities provided by the kernel to perform higher-level house-keeping tasks, often acting as servers in a client-server relationship.

Some would include a graphical user interface and window system as part of the OS, others would not.

---

*The facilities an operating system provides and its general design philosophy exert an extremely strong influence on programming style and on the technical cultures that grow up around the machines on which it runs.*

## 6 Questions course addresses ?

1. : As an **IT manager** within an organisation, responsible for buying decisions for the company and installation. Issues need to consider include options in various operating systems and an analysis of the problem domain.
2. : As a **developer**. Developing a new operating system for an in-house application, or as a consultant on a team. Also need to address types of operating systems available and their potential use, and issues involved in operating systems : what they perform and how.
3. : As a **user**. Consider two or three main operating systems in use. If presented with either of these, the basics of how they work, and how to put them to use in an application environment.

## Underlying Objectives

- Characteristics of operating systems (OS)
- Why are operating systems needed
- Implementation of main OS functions
- Comparative study between popular OS

## 7 If Beer (non-alcoholic of course) were like Operating Systems ...

### DOS Beer

Requires you to use your own can opener, and that you read the directions carefully before opening the can. Originally only came in an 8oz can, but now comes in a 16oz can. The can is divided into eight compartments of 2oz each, which have to be accessed separately. Soon to be discontinued, although a lot of people are going to keep drinking it after it's no longer available.

---

### Macintosh Beer

At first came only in a 16oz can, but now comes in a 32oz can. Considered by many to be a "light" beer. All the cans look identical. When you take one from the fridge, it opens itself. The ingredients list is not on the can. If you call to ask about the ingredients, you are told that "you don't need to know." A notice on the side reminds you to drag your empties to the trash can.

# Windows 3.1 Beer

The world's most popular beer. Comes in a 16oz can that looks a lot like Mac Beer's. Requires that you already own a DOS Beer. Claims that it allows you to drink several DOS Beers simultaneously, but in reality you can only drink a few of them, very slowly, especially slowly if you are drinking the Windows Beer at the same time. Sometimes, for no apparent reason, a can of Windows Beer will explode when you open it.

---

# Windows 95 Beer

Only just out, Windows 95 Beer looks a lot like Mac Beer but tastes more like Windows 3.1 Beer. It comes in 32oz cans, but when you look inside, the cans only have 16oz of beer in them. Most people will probably keep drinking Windows 3.1 Beer until their friends try Windows 95 Beer and say they like it. The ingredients list, when you look at the small print, has some of the same ingredients that come in DOS Beer, even though the manufacturer claims that this is an entirely new brand.

# OS/2 Beer

Comes in a 32oz can. Does allow you to drink several DOS Beers simultaneously. Allows you to drink Windows 3.1 Beer simultaneously too, but somewhat slower. Advertises that the cans won't explode when you open them, even if you shake them up. You never see anyone drinking OS/2 Beer, but the manufacturer (International Beer Manufacturing) claims that 9 million six-packs have been sold.

---

# Windows NT Beer

Comes in a 32oz cans, but you can only buy it by the truck load. This causes most people to have to go out and buy bigger refrigerators. The can looks just like Windows 3.1 Beer's, but the company promises to change the can to look just like Windows 95 Beer's. Touted as an "industrial strength" beer, and suggested only for use in bars.

# Unix Beer

This very heavy beer comes in 32oz cans, and has been around for years, rumour has it that it was originally brewed as a hoax by a couple of bored workers, only for them to find that some people actually liked the stuff. It tends to be drunk only by freaks or eccentric academics, often with beards; and drinkers of it do not like drinkers of any other beer. In its basic form it doesn't look particularly impressive, but with the addition of a magic ingredient named "X", it can be converted into an all-singing all-dancing beer on a par with the others. Many other varieties exist, with a huge range of flavours and (often unpronounceable) ingredients. It must be stressed, however, that even then it is strictly an acquired taste.

---

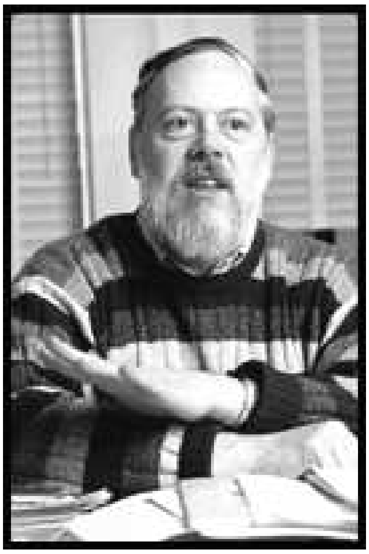

Figure 3: Dennis Ritchie

## 8 Course Content

- **General Issues and where OS fit in**
- **Operating System Evolution**
- **OS Nucleus : The Kernel**
- **Input/Output Systems**
- **Device Management**
- **Filing System and Management**
- **Scheduling**
- **Memory Management**
- **Security**
- **Cross-Platform Glue**
- **Real Time programming**
- **Concurrent and Distributed Systems**

---

# CASE STUDY

Unix vs. Windows NT

**WATCH OUT!**

for **Questions!**

**DON'T FORGET to read the WEB PAGES**

## 9 Books n' stuff


1. Modern Operating Systems, *Tanenbaum* Prentice Hall, 92
2. OS Concepts, *Peterson, Silberschatz and Galvin* Addison Wesley, 94
3. Operating Systems, *Stallings* Macmillan, 92
4. Concurrent Systems : an integrated approach to Operating Systems, Databases and Distributed Systems, *Bacon, J.* Addison Wesley, 93
5. CORBA design patterns, *Mowbray and Malveau* Wiley, 96
6. CORBA : Fundamentals and Programming, *Seigel, ed.* Wiley, 96
7. The design of OS/2, *Deitel and Kogan*, Addison-Wesley, 92
8. Distributed operating systems, *Tanenbaum*, Prentice Hall, 95
9. Essential system administration, *Frisch*, O'Reilly Associates, 93
10. Essential concepts of operating systems : using IBM mainframe examples, *Yuen*, Addison-Wesley, 86

# 10 Examples of Addressing

- **Direct Addressing**
- **Encoded Addressing**
- **Intermediate translation table**
- **Multiple intermediate redirections**

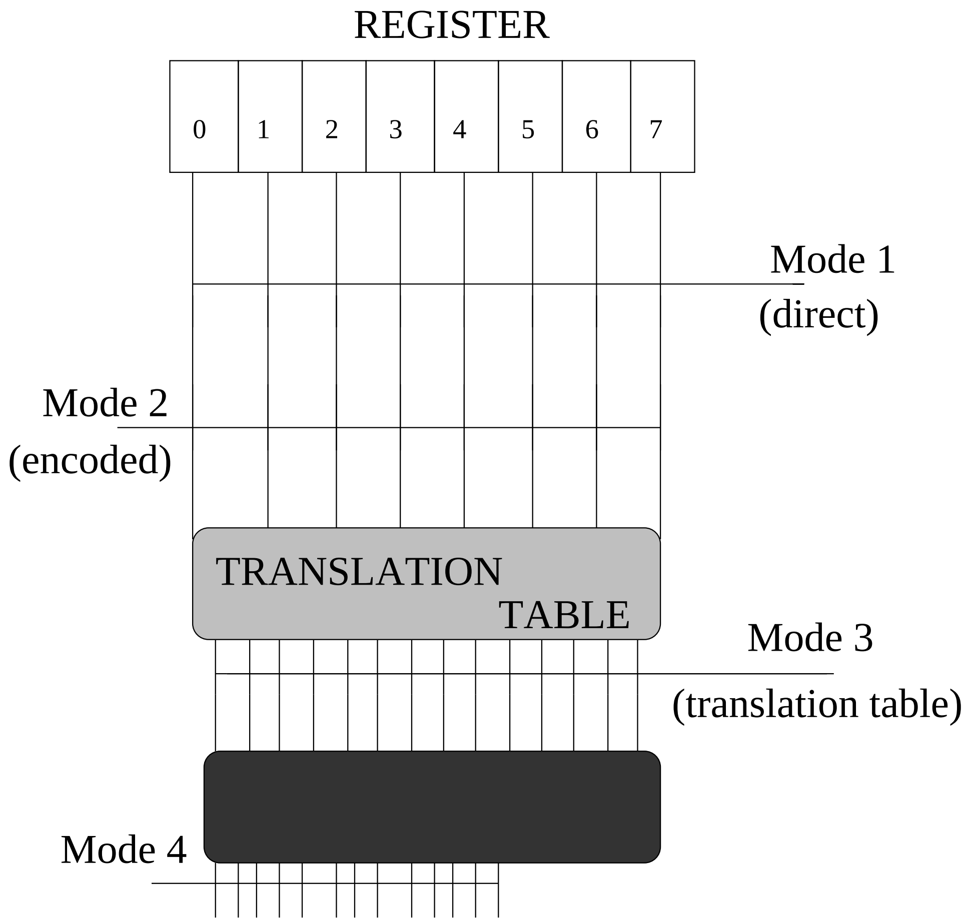


Figure 4: Addressing methods

## 11 Some Interesting Comments ...

"Although their fans speak of them in almost transcendental terms, the major desktop OSes sometimes exhibit all the rigor of a tube of toothpaste: You squeeze here, it bulges there; you squeeze there, it breaks. It's almost axiomatic in computer technology, of course, that every strength is simultaneously a weakness, and nowhere is this more apparent than in the crash performance of the major desktop OSes.

Under stress, Windows 3.1, Windows NT 3.5, OS/2 Warp, and Mac OS 7.5 (the PowerPC and Motorola 68000 versions) are all vulnerable to their own particular brands of system failure. They are often called **General Protection Faults** (GPFs) in the Windows world and **System Bombs** on the Mac side. These flashes of woe (or whoa) typically render the active software application unusable and can freeze the whole system solid, requiring rebooting. As a bonus, you probably lose whatever you were working on, too."

**Bruce Brown** an article in **BYTE**
*Crashing the party*

## 12 What does history tell us?

Take a stroll down memory lane :

| | 1985 | Present | factor |
|---|---|---|---|
| MIPS | 1 | 300 | 300 |
| £/MIPS | 100 K | 50 | 2000 |
| Memory | 128 KB | 64 MB | 500 |
| Disk | 10 MB | 2 GB | 200 |
| Tape | 200 MB | 1 TB | 5000 |

How to deal with this as a designer ?

Tradeoffs ? Time varing techniques

Methods which vary as the problems change ...

**History Phase 1** :
*Cheap Humans - Expensive Machines*

- **Single user** (highly protected environments)
- **Batch** programs (Job Control Languages)
  (load, run, print, perform operation X etc)
- Data Channels / interrups (overlap I/O and computation).
  **DMA** : direct memory access.
- Memory Proection and relocation
  **Multiprogramming** : several programs run at the same time ; users share system.
  *Multiprogramming requires memory protection to keep bugs in one program from crasging system or corrupting other program)*

What does this mean ?
Operating System become large complex programs that must manage all these interactions.
Each step of operation seems logical, but at some point ...

**History Phase 2** :
*Cheap Machines - Expensive Humans*

- **Interactive Timesharing** : cheap terminals with multiple users interacting simultaneously.
  CPU time shared between all users. But what about *Thrashing* (less time in CPU more for I/O).
- **Personal computing** - everyone has their own computer.
- Resource sharing now with a **Network**.

**BUT - what does this all mean ?**

- **Multics** announced in 1963 $\rightarrow$ RAN in 1969
  (Unix based on Multics but simpler!).
- **NT** under development for the last 7 years - *STILL DOESN'T WORK.*
- Situation today much like the past :
  **Operating Systems are enormous complex things**
  *small OS : 100 K lines of code, to big OS : 10 M lines of code.*

**So where do we go from here ?**

---

**Key Aspect of this course :**
**"Understand Operating Systems so we can simplify them"**