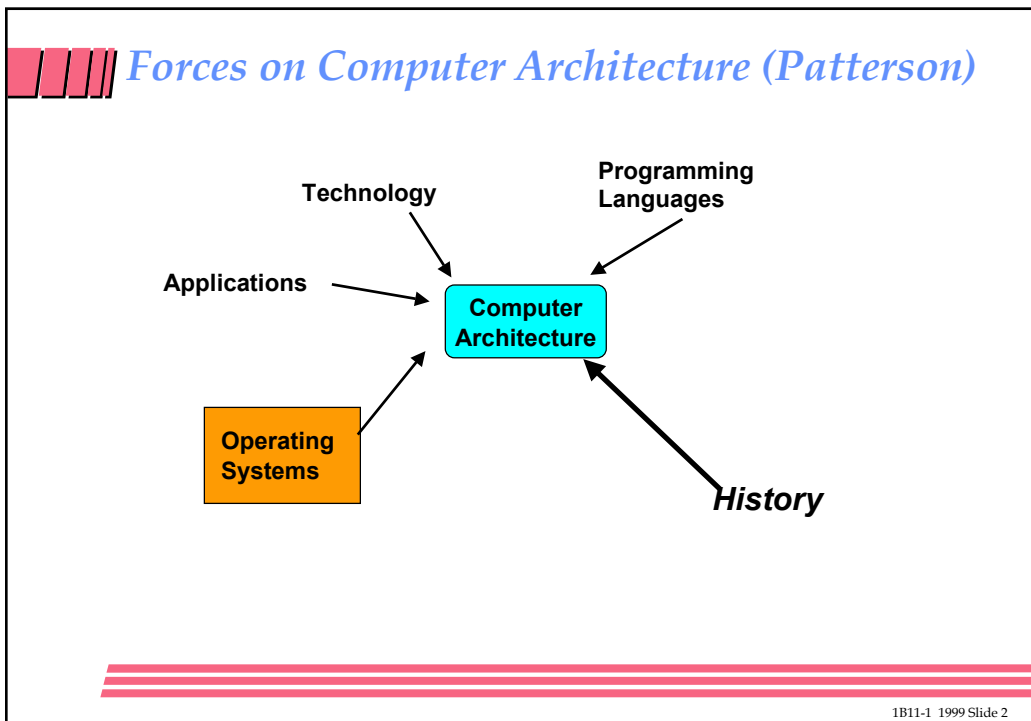



*1B11 Operating Systems*

*Introduction*

Prof. Steve R Wilbur  
s.wilbur@cs.ucl.ac.uk



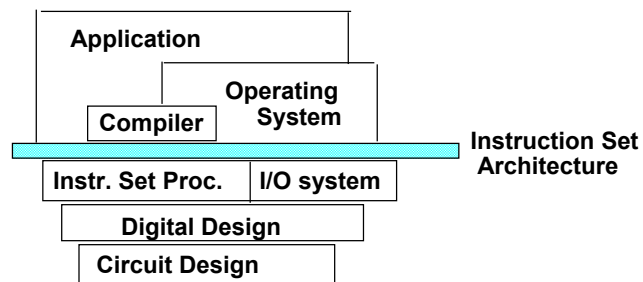
## Lecture Objectives

- 1 What is an Operating System?
- 1 What are the main requirements?
- 1 How have OSs influenced hardware design?

1B11-1 1999 Slide 3

## What is "Computer Architecture" (Patterson)

- 1 Co-ordination of *levels of abstraction*



- 1 Under a set of rapidly changing *forces*

1B11-1 1999 Slide 4

## What is an Operating System?

- 1 Simply, an operating system is a program which enhances the raw hardware to provide *a clean, virtual machine*
- 1 It provides a number of *abstractions* (such as files) not present on the raw hardware
- 1 It provides a number of *services* in an easy-to-use form (such as protection)

1B11-1 1999 Slide 5

## OS Evolution

### *No Operating System*

- 1 Raw machine
- 1 Single user
- 1 Single program
  
- o Programs input from paper tape, cards or by switches
- o *Bootstrap*
- o Data input/output to teleprinter or tape/cards
- o *Polled I/O*

1B11-1 1999 Slide 6

## OS Evolution - 2

### *"Monitor" or Octal/Hex Debugger*

- 1 Similar environment
  - o Entry points to I/O routines at fixed places in memory - saves effort
  - o Managed program loading and memory management
  - o Multiple programs loaded, but only one running
  - o Relocation of programs when loading

1B11-1 1999 Slide 7

## OS Evolution - 3

### *Batch Multiprogramming*

- 1 Share processor between several programs - use "idle" time when processor waiting for I/O
- 1 Disks used directly
  - o Invention of *interrupts* for I/O
  - o Need to avoid damaging others when one program goes out of control - *memory management*
  - o Need to define *scheduling policy*

1B11-1 1999 Slide 8

## OS Evolution - 4

### *Timesharing*

- 1 Interactive shared use of machine by many people
  - o *File* abstraction introduced
  - o *File sharing* and *file protection* needed
  - o Notion of *user identity*
  - o Each user has *virtual machine* - apparently independent of all other users
  - o *Virtual memory* - as much as addressable

1B11-1 1999 Slide 9

## OS Evolution - 5

### *Personal Computer*

- 1 Single user machine with simple file abstractions
  - o Lost concept of user identity
  - o Lost concept of sharing and protection of resources
  - o Lost concept of scheduling

1B11-1 1999 Slide 10

## OS Evolution - 6

### *Workstations*

- 1 *Serially re-usable* single user machine with file and service abstractions
  - o As timesharing systems plus...
  - o Screen as resource to be managed - *windows*
  - o *Event-driven* paradigms
  - o *Networked services* - file and print servers

1B11-1 1999 Slide 11

## History of Operating Systems

- 1 50s-60s: Tape/punch card reader driver/printer + some assembler routines
- 1 60s batch processing and file systems
- 1 70s: multiprocessing and timesharing
- 1 80s: personal computers, workstations and networks
- 1 90s: distributed operating systems
- 1 00s: "Microsoft delay release of Windows 2000 to early 2nd quarter 1901 :-)"

(Courtesy Jon Crowcroft)

1B11-1 1999 Slide 12

## *What is an Operating System?*

- 1 Clean, Abstract Machine
  - o OS provides services and abstractions such as files, processes, windows
  - o Portability, eg. Linux, Unix
- 1 Resource Manager
  - o Scheduler to share processor - batch, foreground/background, timesharing
  - o Memory management - paging, swapping
  - o Screen manager - windows
  - o Mechanisms for accounting; limit enforcement - e.g. printer, file quotas

1B11-1 1999 Slide 13

## *What is an Operating System-2?*

- 1 Protection
  - o Accountability demands user identification and authentication
  - o Prevention of users from interfering with others' entities (files, processes, memory)
  - o Prevention of avoiding OS enforcement

1B11-1 1999 Slide 14

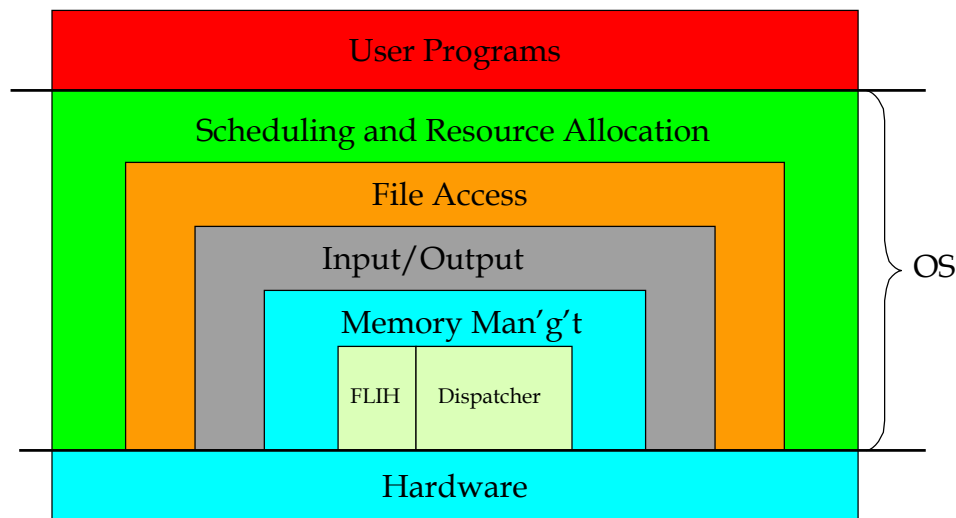
## What is an Operating System-3?

### I Services

- o File Services
  - ] Structure
  - ] Naming
  - ] Sharing/protection/accounting
  - ] Backup/archive - may be integrated
- o Uniform access to I/O abstractions and real devices
- o Measurement, monitoring and debugging tools

1B11-1 1999 Slide 15

## Operating System Model



1B11-1 1999 Slide 16



## *“Onion Skin Model”*

- 1 *Inner layer* built on real hardware
- 1 *Successive layers* built on inner services
- 1 *User Processes* outside Onion-Skin
- 1 *Boundary* between user processes and OS provided by *“system call”* mechanism

  
1B11-1 1999 Slide 17

## *Important Abstractions*

### *Process*

- 1 A process is an “executing” program
- 1 Can be in one of several states:
  - o Running
  - o Ready to Run
  - o Blocked - waiting for e.g.. I/O to complete

  
1B11-1 1999 Slide 18

## ||||| Important Abstractions - 2

### The User

- 1 All this needs the notion of a *user*
  - o accounts are managed through some special files which record user *id*, *name*, *password (shadowed)*, *home directory etc*
  - o user id is used as tag for processes - lets OS track what user can do in terms of IO, Files and other processes at (short term) runtime of programs
  - o user id used to track ownership and protection of entities over the long term
  - o users can be in "hierarchy" and in groups

1B11-1 1999 Slide 19

## ||||| Summary

- 1 Operating System + Hardware = Virtual Machine
- 1 Easier to program, provides additional services like:
  - o multi-user, multi-task, multi-device
  - o long term storage organisation: files
  - o simpler I/O programming paradigm
  - o protection/security/integrity

1B11-1 1999 Slide 20



## *Further Reading*

- | Ritchie C, "Operating Systems inc. UNIX and Windows", 3rd ed, 1997, Letts, 1 85805 302 1, £13.95
- | and many more