Teaching London Computing

Programming for GCSE Topic 5.2: Operating Systems

William Marsh
School of Electronic Engineering and Computer Science
Queen Mary University of London

Aims

- Introduce Operating Systems
- Core concepts
  - Processes
  - Memory management
  - Files
  - I/O
- Creating illusions

Teaching Issue

- Operating Systems are COMPLEX
- What are the big issues?
  - It's just a PROGRAM
  - It creates ILLUSIONS

Syllabus – Operating Systems

- GCSE (OCR)
  - *Candidates should be able to: (a) explain the need for the following functions of an operating system: user interface, memory management, peripheral management, multi-tasking and security*

From AQA A-Level Unit 3

- Role of an Operating System
- Provision of a virtual machine
- Resource management

Understand that the role of the operating system is to hide the complexities of the hardware from the user. In addition, it manages the hardware resources in order to provide for an orderly and controlled allocation of the processors, memories and I/O devices among the various processes competing for them.

History & Variety

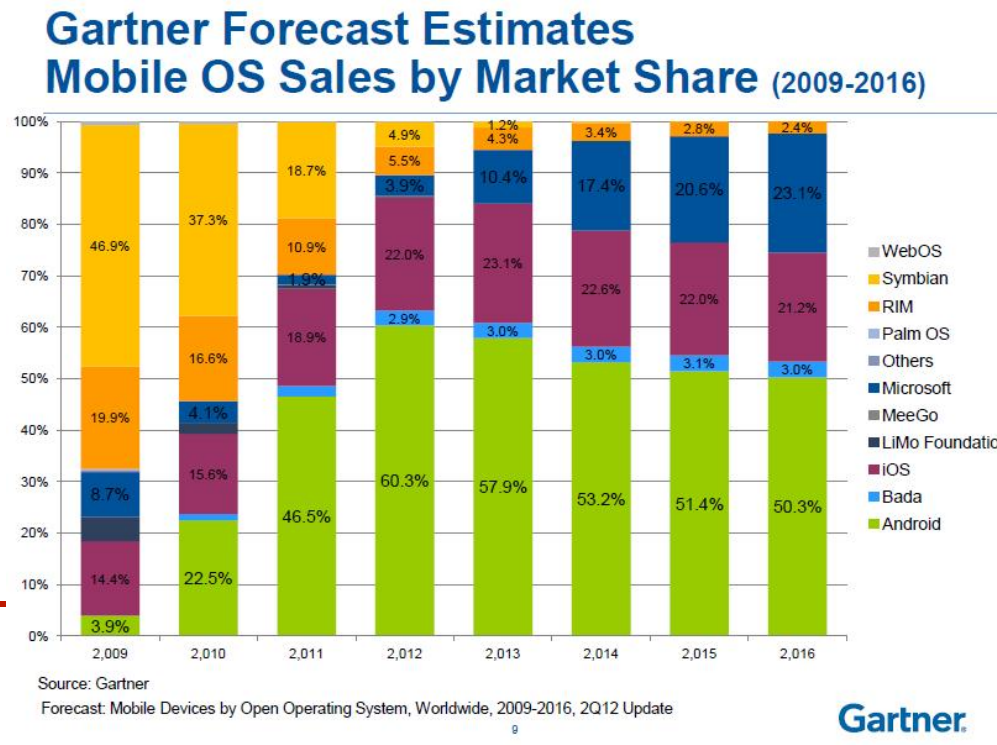
History

- Mainframes
  - Multi-programming
  - Time sharing
- Personal computer
  - *Now with Mainframe OS*

Variety of Operating Systems

- Mainframes
  - IBM z/OS, also Unix: Linux, Solaris
- Servers
  - Linux (2/3) and Windows (1/3)
- PC (desktop, laptop)
- Phones and PDAs

Versions of Linux used from supercomputers to mobiles

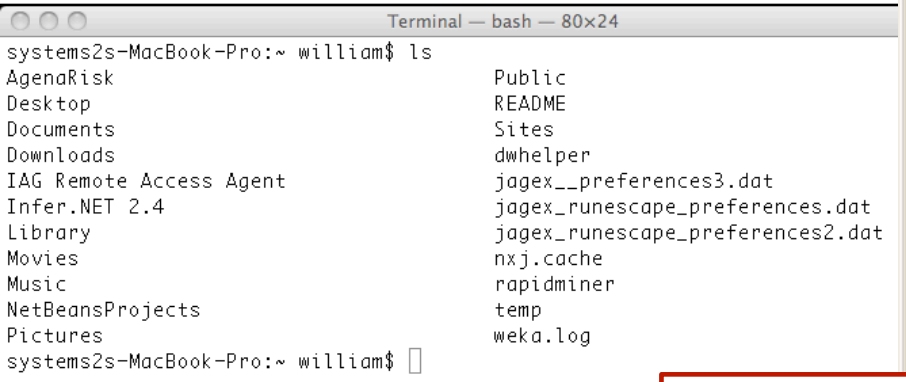


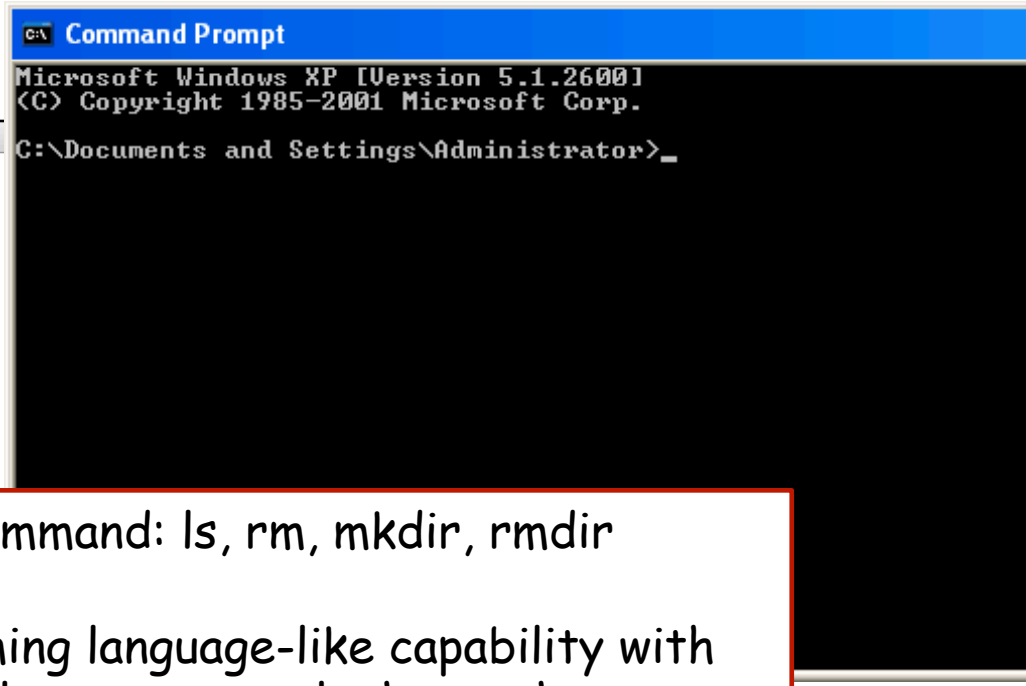
Interface To The Operating System

- Shell
- System calls

Command Line

- Command line interface to Operating System
  - Unix 'shell'
  - Windows command prompt
  - Windows Powershell





Simple command: ls, rm, mkdir, rmdir

Programming language-like capability with scripting languages: perl, php, python

Unix Command for Files

- ls *list the directory*
- mkdir *make a new directory*
- cd *change current directory*
- pwd *show current directory*
- rm *delete a file*
- rmdir *delete a directory*
- cp *copy a file*
- mv *move / rename a file*
- cat *look at a file*

System Calls

- The OS **is** a program
  - Other programs use OS functions
- Some examples using Python

```
import os, sys

print("OS name:", os.name)
print("Platform:", sys.platform)
print("Process id:", os.getpid())
print("User id:", os.getuid())
print("File meta data:", os.stat("os1.py"))
print("Current directory", os.getcwd())
os.chdir("..")
print("Current directory", os.getcwd())

for root, dirs, files in os.walk('.'):
    if len(dirs) > 0 :
        print(root, "contains directories:", dirs)
```

Core Concepts

- Processes
- Managing memory
- File systems
- I/O – hardware
- OS structure

Processes

- An instance of a program running
- A process has
  - Memory
  - Use of resources (I/O)
- Process can exchange data
- Processes are created by other processes

Why Processes?

1. Multi-tasking
   - A single program must wait for the disk
   - Several programs improves throughput

2. Multi-users
   - Different users share computer
   - One user has multiple programs

How Many Programs Run at Once?

- Uniprocessor – only one really
  - Illusion of multiple processes
  - Multiple processes increase throughput
  - Multiple processes allow timesharing
- Multiprocessors – several programs at once
- Concurrency v parallelism
  - Parallelism: use multiple processors to go faster
  - Concurrency: play music while editing

Memory – Who?

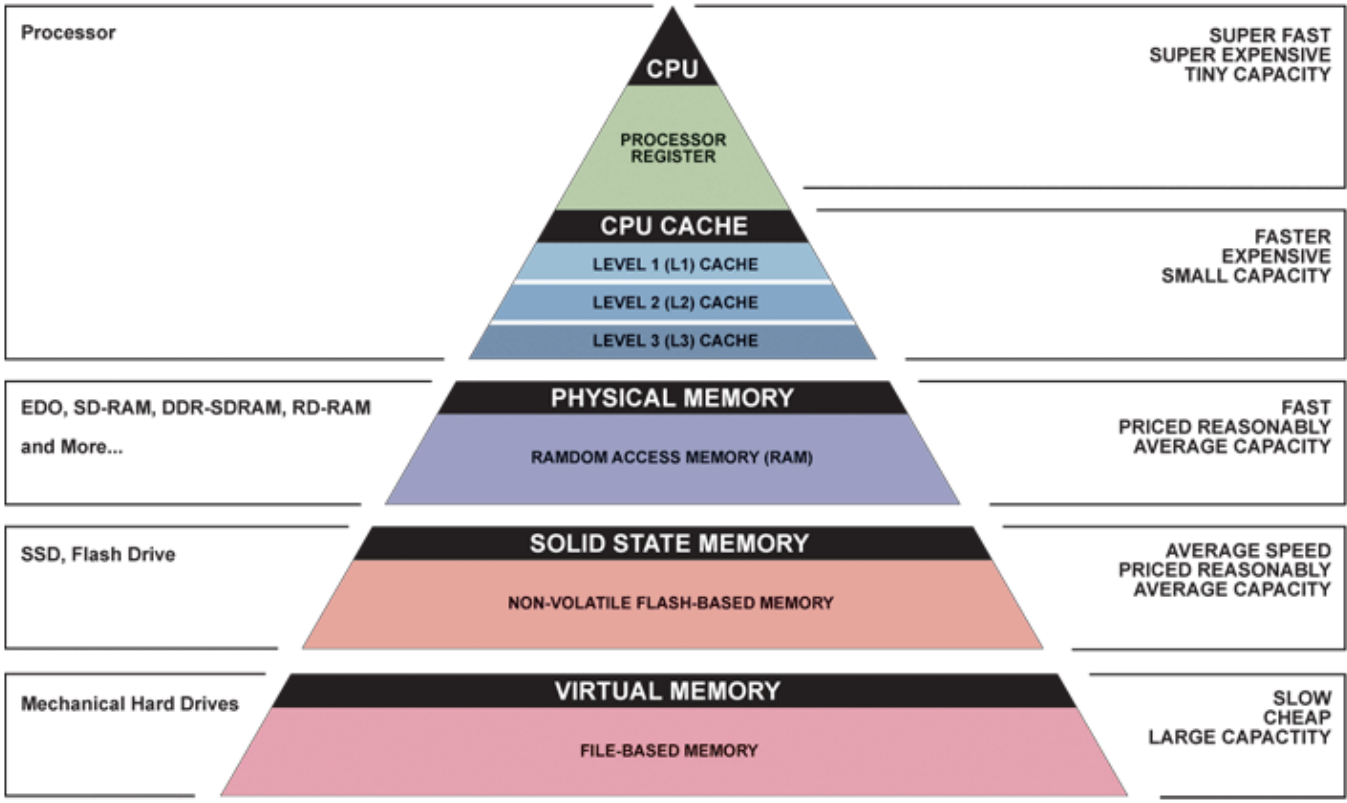
- Lots of process ➔ sharing memory
- Who get the memory?

- Virtual memory
  - Memory as it appears to the program
  - Illusion: your program can use all the memory

- Physical memory
  - What is really happening underneath

Memory – Where?

- Is your data in cache or main memory?
- What happens when the programs overfill memory?
- The OS 'hides' this from the programmer

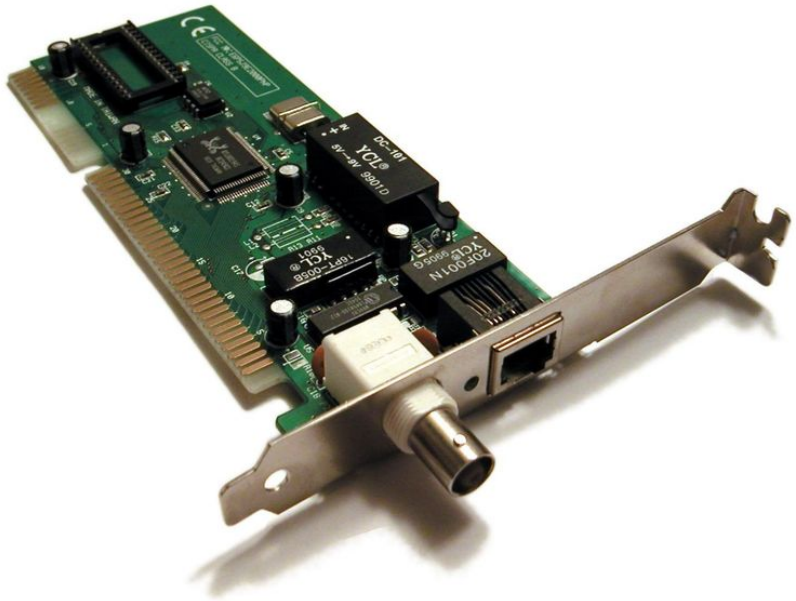


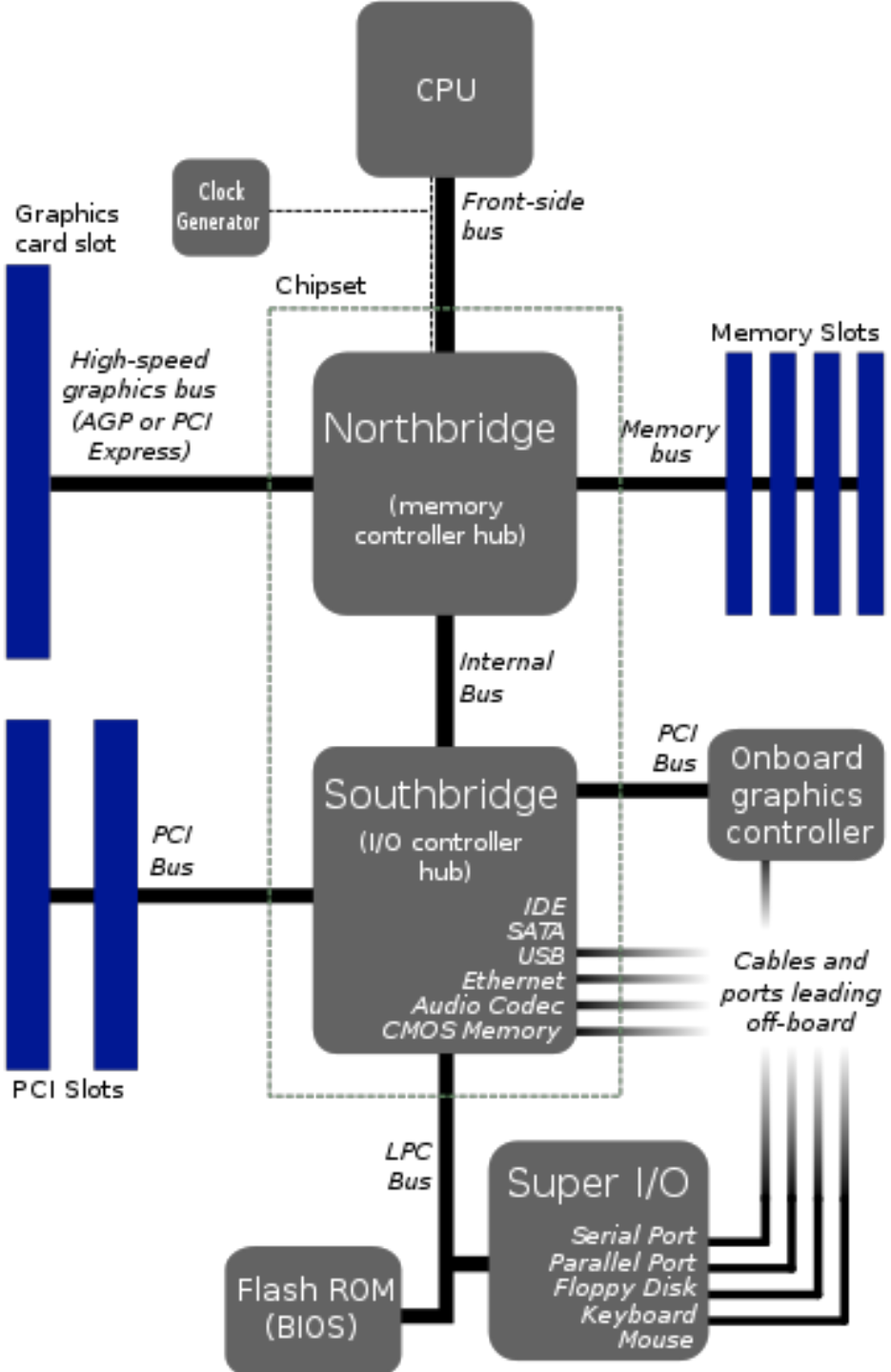
File Systems

- Files organised into directories
- Hierarchy
- Ownership and protection
- Attributes
  - Size
  - Dates

- Illusion: disk is organised into file

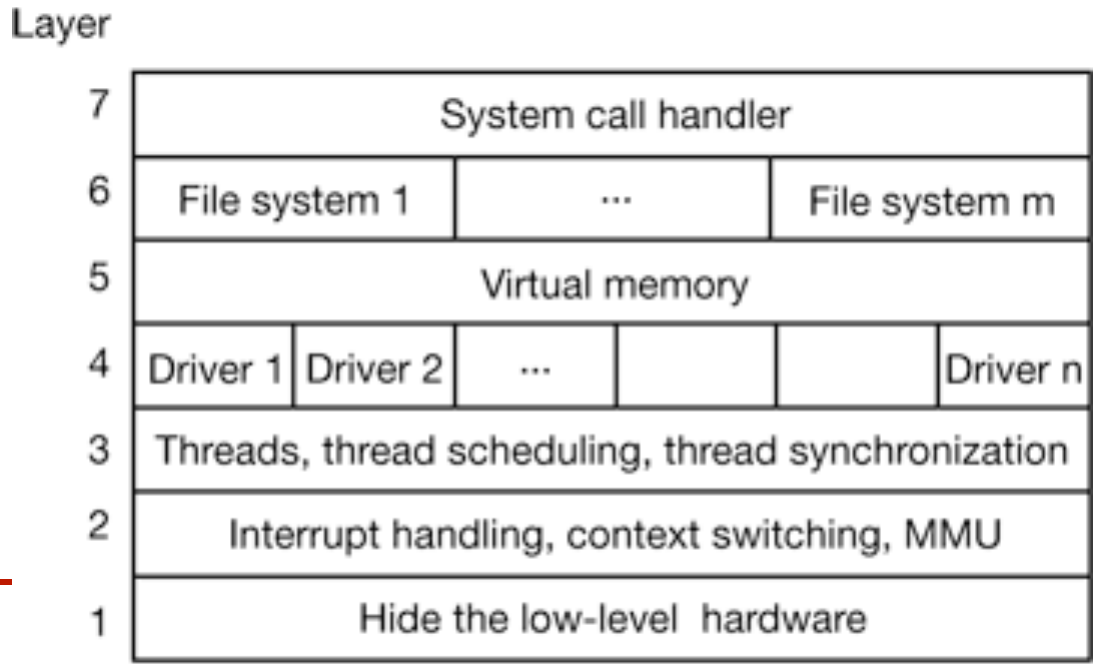
Manage I/O Buses and Drivers





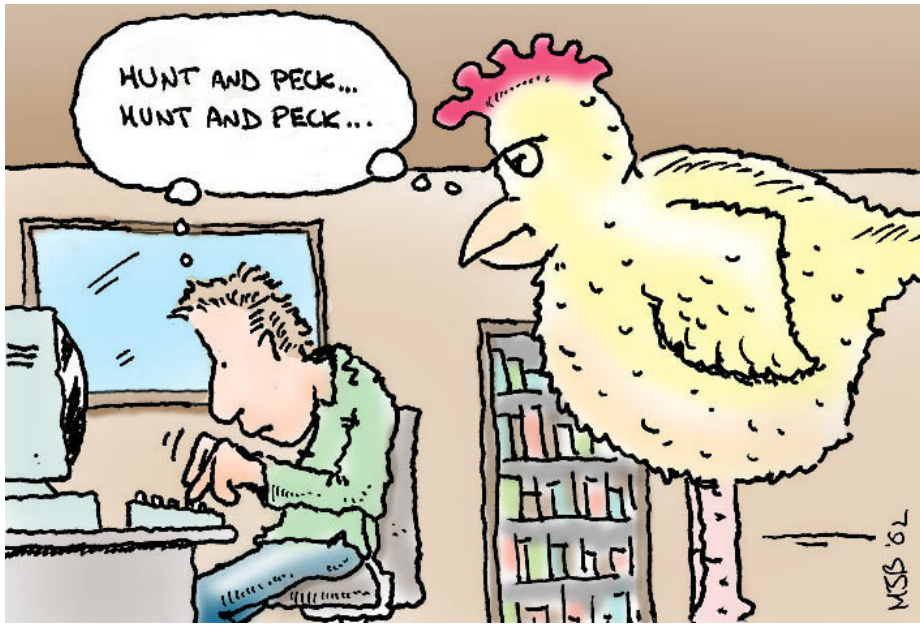
Operating System Structure

- Principles
  - Operating system calls user programs
  - Kernel: control what runs
  - Library of system calls
  - Services
- Kernel
  - Able to use all processor instruction & registers



Interrupts

- I/O device speed varies
- How many CPU clock cycles per keystroke?

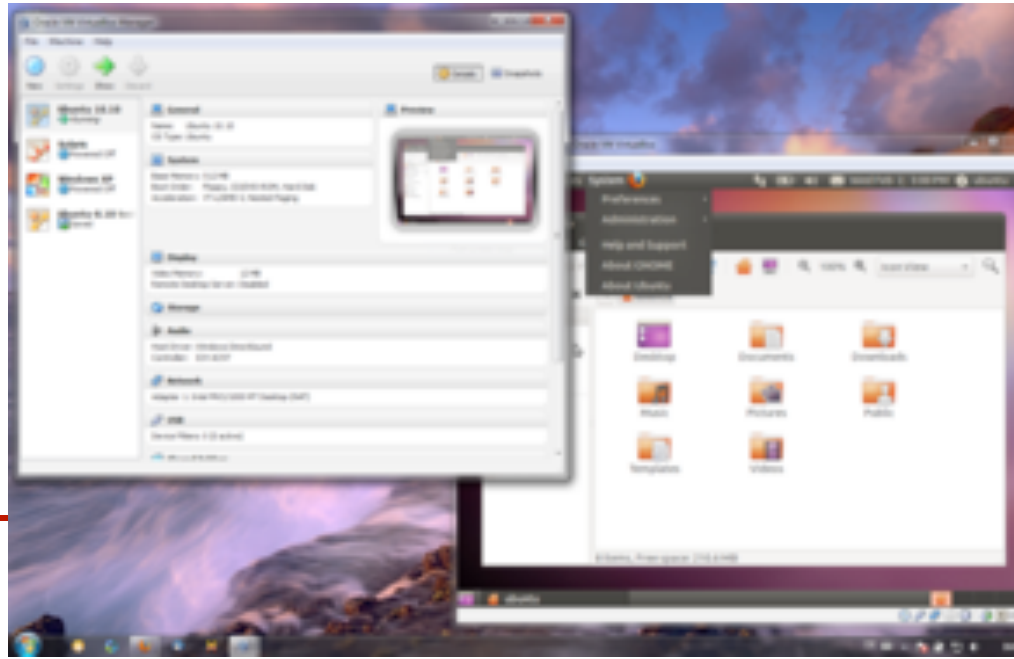


- How to avoid CPU waiting for I/O?
- Interrupt
  - Signal from outside the CPU
  - … changes the program

Install Linux on a Virtual Machine

- Host v. guest
- Virtual machine
  - Oracle virtual box www.virtualbox.org
  - Vmware player http://www.vmware.com/products/player/

- Run linux without reboot
- Memory



SUMMARY

What Is An Operating System?

- Kernel
  - Control of processes
  - Hardware I/O drivers
  - Memory management
- Services and utilities
  - File systems
  - Network interface
- System call: programmers interface
- Shell: simple interface to the OS
- NO clear division: application or OS?

OS is about Illusions

- Several programs are running simultaneously
  - *The computer switches from one to another*
- My program has all the memory
  - *The memory is shared between programs*
- Disk is organised in files
  - *The disk has blocks; the OS maps file names to blocks*
- Storage devices work the same
  - *Files may be arranged differently on magnetic, flash and optical drives*
- OS creates an 'ideal' computer from a real one