

# 操作系统概论

李中国

苏州大学计算机科学与技术学院

# 虚拟机软件及Ubuntu操作系统下载地址

虚拟机软件**VirtualBox**: [www.virtualbox.org](http://www.virtualbox.org)

**Ubuntu**操作系统: [www.ubuntu.com](http://www.ubuntu.com)

# 安装Ubuntu 11.10操作系统: ISO文件


ubuntu-11.10-desktop-i386.iso [read only]

Open Extract

Back Location: /




Name	Size	Type	Date M
.disk	185 bytes	Folder	
boot	589 bytes	Folder	
casper	718.6 MB	Folder	
dists	11.7 kB	Folder	
install	1.7 MB	Folder	
isolinux	982.4 kB	Folder	
pics	14.9 kB	Folder	
pool	4.1 MB	Folder	
preseed	1.2 kB	Folder	
autorun.inf	143 bytes	unknown	12 Octo
md5sum.txt	4.4 kB	plain text d...	12 Octo
README.diskdefines	225 bytes	README d...	12 Octo
wubi.exe	2.5 MB	DOS/Wind...	12 Octo

# 用Wubi安装Ubuntu 11.10操作系统

 **Ubuntu Installer** 

 **You are about to install Ubuntu-11.10**  
Please select username and password for the new account

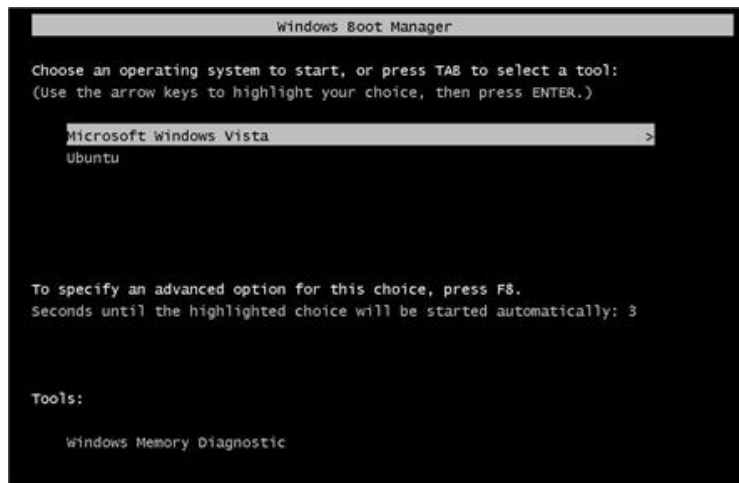
---

	Installation drive: <div>C: (56GB free) ▾</div>		Language: <div>English (GB) ▾</div>
	Installation size: <div>18GB ▾</div>		Username: <div><input type="text"/></div>
	Desktop environment: <div>Ubuntu ▾</div>		Password: <div><input type="password"/></div> <div><input type="password"/></div>

Rev 241

[Accessibility](#) [Install](#) [Cancel](#)

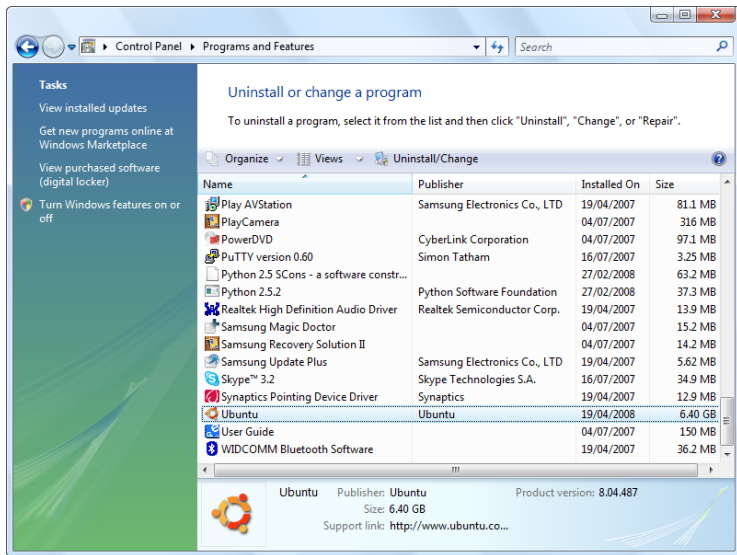
# 用Wubi安装Ubuntu以后的启动界面



# Ubuntu 11.10图形界面



# 卸载Wubi安装的Ubuntu 11.10



# 课程内容

- ▶ 操作系统概念及理论
- ▶ 阅读操作系统源代码并完成编程作业
- ▶ 学习本课程的知识要求:
  - ▶ C语言程序设计
  - ▶ 数据结构
  - ▶ 计算机组成原理
  - ▶ 计算机体系结构
  - ▶ 汇编语言程序设计



# 课程要求及考核方法

- ▶ 期末考试: 50%
- ▶ 平时作业(程序设计): 50%
- ▶ 两项均需及格方能通过本课程

Pass: 30, 30

Fail: 50, 10

Excellent: 40+, 40+

- ▶ 两点要求:
  - 编程作业 需要独立完成
  - 课堂考勤 要求

# 什么是操作系统

- ▶ 介于用户与计算机硬件之间的程序
- ▶ 思考: 操作系统的用户
- ▶ 操作系统的设计目标
  - ▶ 执行用户程序
  - ▶ 使计算机系统易于使用
  - ▶ 高效利用计算机硬件
- ▶ 对比: 应用软件的设计目标



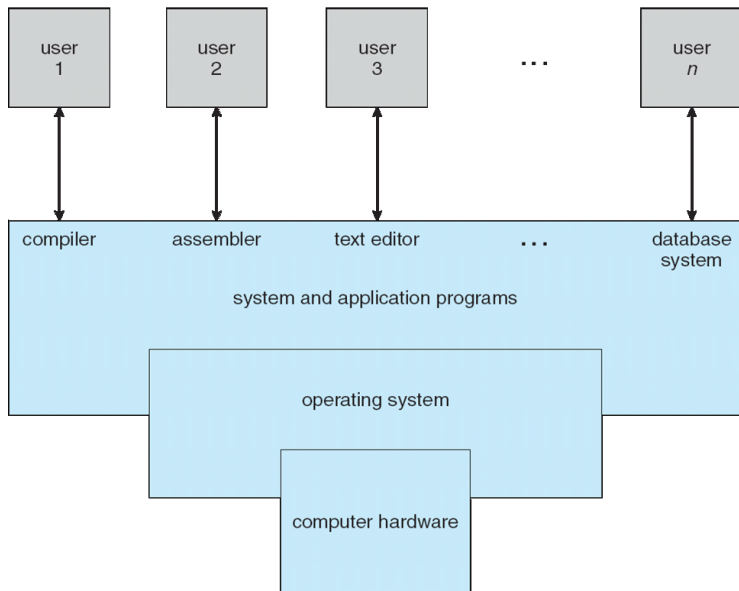
# 为什么学习操作系统

- ▶ 计算机系统的不可缺少的关键部分
- ▶ 非常复杂
  - ▶ Linux Kernel 2.6.35 — 13.5 M
  - ▶ FreeBSD — 8.8 M
  - ▶ Mac OS X 10.4 — 86 M
  - ▶ Windows Server 2003 — 50 M
- ▶ 真正理解计算机系统的工作原理
- ▶ 涉及硬件、编程语言、数据结构、算法等多个领域

# 计算机系统组成

- ▶ 计算机硬件: 提供基本计算资源
  - ▶ CPU, Memory, I/O devices...
- ▶ 操作系统
  - ▶ 控制、协调用户和程序对硬件资源的使用
- ▶ 应用程序: 解决用户的计算问题
  - ▶ Word processors, compilers, web browsers, database systems, video games
- ▶ 用户
  - ▶ People, machines (other computers)

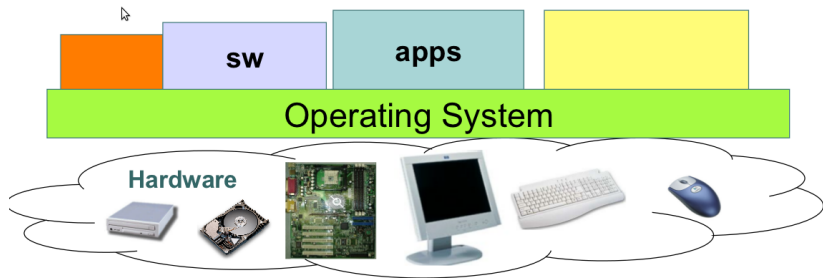
# 计算机系统组成



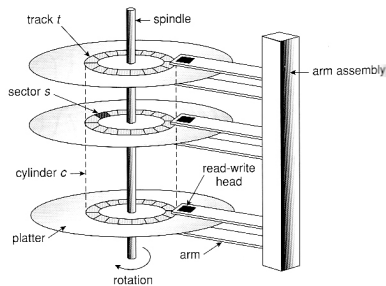
# 操作系统的作用

- ▶ 提供容易使用的界面(终端用户及程序员)
- ▶ 最大限度地提高资源利用率(CPU,内存)
- ▶ 为多用户提供分时服务(time sharing system)
- ▶ 在多用户多系统之间实现资源共享(存储、打印机)
- ▶ 嵌入式设备：界面问题、电池寿命问题(不只是OS的任务)

# 操作系统的作用：资源管理



# 操作系统的作用: 提供易于使用的界面

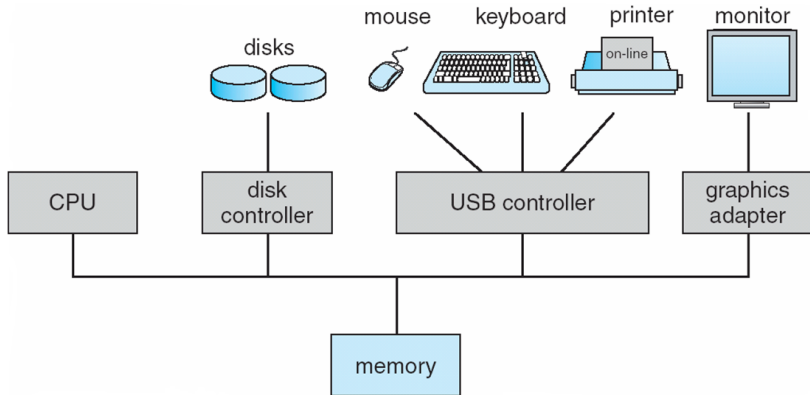


**Figure 12.1** Moving-head disk mechanism.



# 计算机硬件系统

- ▶ CPUs, 设备控制器等通过总线与内存连接
- ▶ 竞争内存周期, 实现CPU、设备控制器间的并发执行

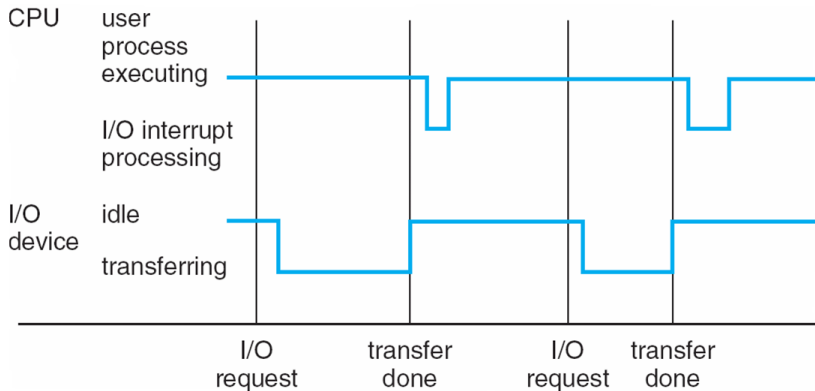


# 计算机硬件系统

- ▶ I/O设备与CPU可以并发运行
- ▶ 每个设备控制器控制一种I/O设备
- ▶ 设备控制器有自己的本地缓存
- ▶ CPU — 内存— 设备缓存— 设备
- ▶ I/O: 从设备到设备控制器的缓存
- ▶ I/O完成后，设备控制器通过中断通知OS

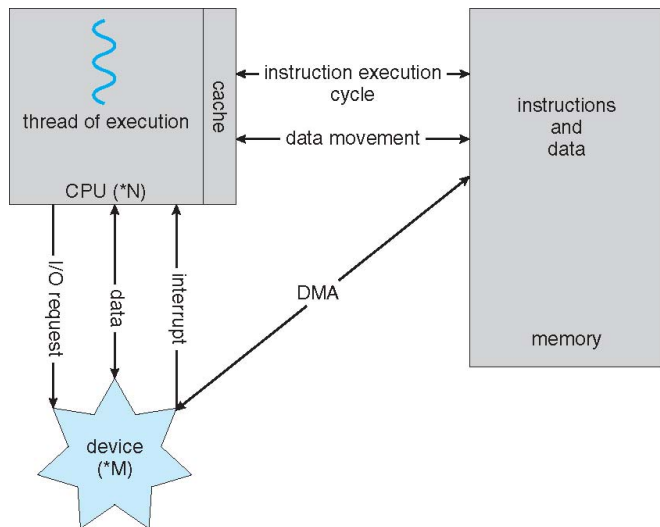
# 中断(interrupt)

- ▶ 操作系统由中断驱动
- ▶ 硬件中断与软件中断



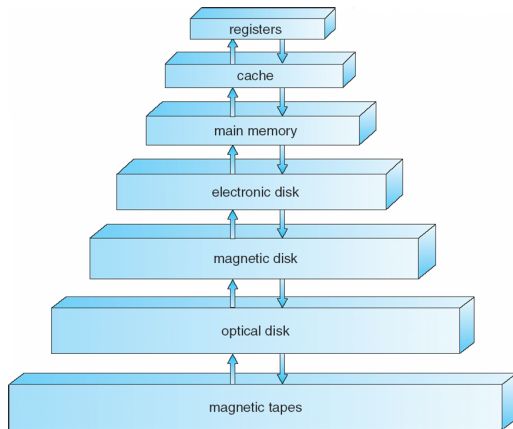
# 直接内存访问(DMA)

- ▶ 高速I/O设备与内存之间采用**DMA**传输数据
- ▶ 每传输1块数据产生一个中断

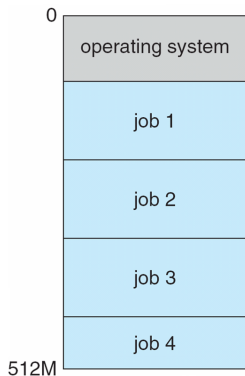


# 存储层级

- 存取速度
- 制造成本
- Caching技术



# 多道程序设计与分时系统



- ▶ 多道程序设计(multiprogramming)
  - ▶ 单个用户或程序无法使CPU或外设保持忙碌
  - ▶ 引入多道程序设计技术
  - ▶ 多个作业(jobs)驻留内存
  - ▶ 操作系统需要进行作业调度
  - ▶ 需要等待I/O时，调入另一作业运行
- ▶ 分时系统(timesharing system)与交互式计算(interactive computing)

# 核心态与用户态

- ▶ 核心态: 执行操作系统代码
- ▶ 用户态: 执行用户程序代码
- ▶ 思考: 哪些指令需要在核心态下执行?
- ▶ 系统调用导致从用户态转入核心态

