

新工科|普通高校移动开发系列课程优秀教材

- 著名移动开发专家关东升系统论述iOS开发的经典著作
- 配教资源：教学课件|教学视频|源代码
- 高校推荐：大连理工大学|中南大学|北京工业大学
青岛科技大学|中国传媒大学
- “从零开始学Swift” 视频课程1400分钟
- “iOS会员——iOS开发学习路线图” 视频课程9900分钟



关东升 著

移动操作系统原理与实践

——基于iOS与Swift编程语言



清华大学出版社

移动操作系统原理与实践

——基于 iOS 与 Swift 编程语言

关东升 著

清华大学出版社
北京

内 容 简 介

本书是专门为高校师生学习 Swift 语言和 iOS 应用开发而编著的教材。书中首先详细介绍了 Swift 语言,然后展开介绍 iOS 开发技术,最后安排了综合案例。全书分别介绍了如下内容:操作系统的基本概念;苹果硬件设备和 iOS 操作系统;Swift 语言的基本语法与程序设计;iOS 应用开发基础;iOS 应用界面技术;表视图;界面布局与 Auto Layout 技术;屏幕适配与 Size Class 技术;应用导航模式;多媒体开发;图形图像开发;数据存储;网络数据交换格式;网络通信。最后一章介绍了综合案例——奥运会应用设计与开发,该章给出了完整的 iOS 应用分析和设计,以及编程实现,架构采用分层设计实现,开发中采用敏捷开发方法。

本书适合作为高等院校与职业院校“移动操作系统”与“iOS 应用开发”等课程的教材,也适合广大 iOS 应用开发人员自学参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

移动操作系统原理与实践:基于 iOS 与 Swift 编程语言/关东升著. —北京:清华大学出版社,2017
ISBN 978-7-302-48156-0

I. ①移… II. ①关… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 208479 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:李建庄

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:26

字 数:631 千字

版 次:2017 年 12 月第 1 版

印 次:2017 年 12 月第 1 次印刷

印 数:1~2500

定 价:69.00 元

产品编号:076188-01

前言

PREFACE

随着 iOS 应用开发在企业中的需求越来越多,很多高校也考虑开设 iOS 应用开发课程,但目前图书市场上缺乏针对高校教学而定制的 iOS 应用开发教程。笔者有多年的 iOS 应用开发经验,出版了多种畅销的 iOS 应用开发图书,其中《iOS 应用开发指南》更是获得了广大读者的认可。为满足高校教师的需求,应清华大学出版社盛东亮编辑之邀,笔者决定出版一套 iOS 应用开发方面的教材。

本书主要采用苹果新推出的计算机语言——Swift,这主要是因为 Swift 是一门开源计算机语言,未来将应用于多个平台和领域开发,是非常重要的语言。因此,本书首先详细介绍 Swift 语言,然后再介绍 iOS 应用开发的主要技术。内容安排前后合理,由浅入深。

为了帮助读者更好地消化吸收书中内容,在每章后面都有同步练习,并在附录中附有解答。另外,还为购买本书的高校教师提供配套课件和素材。

本书虽然为高校编写,满足高校教学需要,介绍 Swift 语言和 iOS 应用开发,但是也适合对 Swift 语言和 iOS 应用开发技术感兴趣的读者。

为了更好地为广大读者提供服务,我们专门为本书建立了一个网站,具体网址为 www.51work6.com/book/ios61.php,读者可以了解书中内容、下载源代码、下载课件等,并对本书内容发表评论,提出宝贵意见。

为了及时地把书中的问题、失误和纠正反馈给广大读者,我们专门在 www.51work6.com/book/ios61.php 中建立了一个勘误专区。另外,读者如果有什么问题,可以在网上留言,也可以发送电子邮件到 eorient@sina.com,我们会在第一时间回复您。也可以通过笔者的新浪微博(@tony_关东升)或微信(q270258799)联系我们。

本书主要由关东升编写。此外,智捷课堂团队的赵大羽、赵志荣、关锦华、赵浩丞、王馨然和闫婷娇也参与了本书部分内容的写作。感谢赵大羽老师手绘了书中全部草图,并从专业的角度修改书中图片,更加真实完美地奉献给广大读者。感谢清华大学出版社的盛东亮编辑给我们提供的宝贵意见。感谢我的家人容忍我的忙碌,以及对我的关心和照顾,使我能抽出较多时间,投入全部精力专心编写本书。

由于时间仓促,书中难免存在不妥之处,敬请批评指正。

关东升

2017 年 8 月

学习说明

Study Shows

一、源代码及教学课件

下载地址为 www.51work6.com/book/ios61.php。

二、赠送学习视频教程

1. “从零开始学 Swift”视频教程

所有购买本书的读者可获赠约 24 小时的下述视频教程：

- (1) 从零开始学 Swift——语法篇(第一季)：508 分钟。
- (2) 从零开始学 Swift——语法篇(第二季)：241 分钟。
- (3) 从零开始学 Swift——编程规范(第三季)：38 分钟。
- (4) 从零开始学 Swift——Swift 与 Objective-C\C\C++ 混合编程(第四季)：325 分钟。
- (5) 从零开始学 Swift——同步练习题解析：307 分钟。

说明：上述课程为正在热销课程，定价 119.00 元人民币，读者购买本书后，凭书中央带的学习卡中的代金卡号到 zhijieketang.com 网站购买该课程，自购买之日起三月内有效。视频教程学习地址为 <http://www.zhijieketang.com/classroom/8/courses>。

代金卡号使用具体说明：首先智捷课堂视频平台(www.zhijieketang.com)注册并登陆，然后找到相应课程，接着选择页面中的【购买课程】→【去支付】→【输入优惠码】→【使用】。

2. “iOS 会员——iOS 开发学习路线图”视频教程

凡是订购本书作为课堂教材使用的高校教师，可获赠时长约 160 小时的下述视频教程：

- (1) iOS 开发指南第一季——第一个 iOS 应用程序：62 分钟
- (2) iOS 开发指南第二季——Cocoa Touch 框架与构建应用界面：101 分钟
- (3) iOS 开发指南第三季——UIView 与视图：170 分钟
- (4) iOS 开发指南第四季——表视图：167 分钟
- (5) iOS 开发指南第五季——界面布局与屏幕适配：97 分钟
- (6) iOS 开发指南第六季——视图控制器与导航模式：170 分钟
- (7) iOS 开发指南第七季——iOS 常用设计模式：91 分钟
- (8) iOS 开发指南第八季——iOS 分层架构设计：105 分钟
- (9) iOS 开发指南第九季——数据持久化：130 分钟
- (10) iOS 开发指南第十季——数据交换格式：87 分钟
- (11) iOS 开发指南第十季——REST 风格 Web Service：106 分钟
- (12) iOS 开发指南第十二季——定位服务：110 分钟
- (13) iOS 开发指南第十三季——苹果地图应用：105 分钟

- (14) iOS 开发指南第十四季——iOS 测试驱动与单元测试: 95 分钟
- (15) iOS 开发指南第十五季——性能优化: 54 分钟
- (16) Swift 语言视频教程入门与提高(上): 342 分钟
- (17) Swift 语言视频教程入门与提高(下): 973 分钟
- (18) iOS8 开发视频教程 Swift 语言版 Part 1——第一个 iOS 应用程序: 171 分钟
- (19) iOS8 开发视频教程 Swift 语言版 Part 2——UIView 与视图: 306 分钟
- (20) iOS8 开发视频教程 Swift 语言版 Part 3——iOS 8 多分辨率屏幕适配: 192 分钟
- (21) iOS8 开发视频教程 Part 4——iOS 数据源协议、委托协议与高级视图: 151 分钟
- (22) iOS8 开发视频教程 Swift 语言版 Part 5——iOS 表视图: 312 分钟
- (23) iOS8 开发视频教程 Swift 语言版 Part 6——iOS 视图控制器与导航模式: 340 分钟
- (24) iOS8 开发视频教程 Swift 语言版 Part 7——iOS 常用设计模式: 125 分钟
- (25) OS 开发视频教程 Part 8——iPhone 与 iPad 应用开发的差异: 115 分钟
- (26) iOS8 开发视频教程 Swift 语言版 Part 9——iOS 分层架构设计: 203 分钟
- (27) iOS8 开发视频教程 Swift 语言版 Part 11——访问 Web Service: 414 分钟
- (28) iOS8 开发视频教程 Swift 语言版 Part 12——iOS 定位服务与地图应用: 269 分钟
- (29) iOS8 开发视频教程 Swift 语言版实战——2016 里约热内卢奥运会指南: 415 分钟
- (30) Swift 项目开发实战——基于分层架构的多版本 iPhone 计算器: 193 分钟
- (31) Objective C 编程基础视频课程: 537 分钟
- (32) iOS 多媒体技术精讲视频课程: 224 分钟
- (33) OS 图形图像与动画: 236 分钟
- (34) 项目实战——酒店预订系统 iPhone 客户端开发(ObjectiveC 版): 466 分钟
- (35) 项目实战——51CTO 学院 iPad 视频客户端开发: 349 分钟
- (36) iOS 中的下拉刷新及等待指示器: 38 分钟
- (37) iOS 进阶开发——Socket、Bonjour、对等网络、蓝牙通信: 113 分钟
- (38) iOS 进阶开发——调试程序: 164 分钟
- (39) iOS 测试: 183 分钟
- (40) iOS 进阶课程——iCloud 编程: 131 分钟
- (41) iOS 进阶开发——iOS 性能优化: 186 分钟
- (42) iOS 进阶课程——苹果的 WebService: 143 分钟
- (43) iOS 协同开发——iOS 代码版本控制 GitHub 代码托管服务: 211 分钟
- (44) iOS 进阶课程——Passbook 编程: 120 分钟
- (45) iOS 开发中如何使用广告——iAd-Admob: 215 分钟
- (46) iOS 的内购——应用内购买: 54 分钟
- (47) iOS 进阶课程——ASIHTTPRequest 框架: 47 分钟
- (48) iOS 进阶课程——Newsstand 编程: 103 分钟
- (49) iOS 进阶课程——iOS 推送通知: 91 分钟
- (50) iOS 架构设计: 161 分钟

说明: 上述课程为正在热销课程,定价 2600.00 元人民币,已订购教材的教师请与清华大学出版社盛东亮编辑联系(dongliang.sheng@qq.com)。工作人员会给您开通该课程三月内免费使用权限。学习地址为 <http://www.zhijieketang.com/classroom/3/courses>。

目录

CONTENTS

第 0 章 移动操作系统概论	1
0.1 操作系统的原理与概念	1
0.1.1 隐藏硬件细节	2
0.1.2 资源管理	2
0.1.3 操作系统的历史	2
0.2 操作系统的分类	4
0.2.1 批处理操作系统	4
0.2.2 分时操作系统	4
0.2.3 实时操作系统	5
0.2.4 个人计算机操作系统	5
0.2.5 网络操作系统	6
0.2.6 分布式操作系统	6
0.2.7 嵌入式操作系统	6
0.3 移动操作系统	6
0.3.1 移动设备的特征	6
0.3.2 主要的移动操作系统	7
0.3.3 移动操作系统的应用和发展	9
第 1 章 iOS 操作系统及 Swift 编程语言简介	10
1.1 苹果的硬件设备与操作系统	10
1.2 Swift 语言概述	10
1.3 iOS 概述	10
1.4 开发环境及开发工具	11
1.4.1 Xcode 开发工具	11
1.4.2 Xcode 安装	12
1.4.3 Xcode 卸载	13
1.4.4 Xcode 界面	14
本章小结	15
同步练习	15
第 2 章 Swift 语言程序设计(上)	16
2.1 第一个 Swift 程序	16
2.1.1 创建 Playground	16
2.1.2 创建 macOS 工程	20
2.2 代码解释	23

2.3	标识符	25
2.4	关键字	25
2.5	常量和变量	26
2.5.1	常量	26
2.5.2	变量	27
2.6	运算符	27
2.6.1	算术运算符	27
2.6.2	算术赋值运算符	28
2.6.3	关系运算符	29
2.6.4	逻辑运算符	31
2.6.5	位运算符	32
2.7	Swift 数据类型	33
2.7.1	整型	34
2.7.2	浮点型	35
2.7.3	布尔型	36
2.7.4	元组类型	36
2.7.5	数据类型转换	38
2.8	可选类型	39
2.8.1	可选类型概念	39
2.8.2	可选类型值拆包	39
2.8.3	可选绑定	40
2.9	字符	41
2.9.1	Unicode 编码	41
2.9.2	转义符	42
2.10	字符串	42
2.10.1	字符串创建	43
2.10.2	字符串拼接	43
2.10.3	字符串插入、删除和替换	44
2.10.4	字符串比较	45
2.11	控制语句	46
2.11.1	分支语句	46
2.11.2	循环语句	53
2.11.3	跳转语句	55
2.11.4	值绑定	60
2.12	集合	63
2.12.1	数组集合	63
2.12.2	字典集合	66
2.13	函数	69
2.13.1	定义函数	69
2.13.2	函数调用	70
2.13.3	函数类型	71
2.14	闭包	71
2.14.1	闭包的概念	71

2.14.2	使用闭包表达式	73
2.14.3	类型推断简化	73
2.14.4	隐藏 return 关键字	74
2.14.5	省略参数名	75
2.14.6	使用尾随闭包	75
2.14.7	捕获上下文中的变量和常量	76
本章小结		78
同步练习		78
第 3 章 Swift 语言程序设计(下)		82
3.1	Swift 中的面向对象类型	82
3.1.1	枚举	82
3.1.2	结构体与类	84
3.2	可选链	86
3.2.1	可选链概念	86
3.2.2	使用问号和感叹号	88
3.3	访问限定	89
3.3.1	访问范围	89
3.3.2	访问级别	89
3.3.3	使用访问级别最佳实践	92
3.4	属性	95
3.4.1	存储属性	95
3.4.2	计算属性	96
3.4.3	静态属性	98
3.5	方法	103
3.5.1	实例方法	103
3.5.2	静态方法	103
3.6	构造函数	105
3.6.1	默认构造函数	105
3.6.2	构造函数与存储属性初始化	107
3.6.3	使用参数标签	108
3.7	析构函数	109
3.8	类继承	110
3.8.1	继承概念	110
3.8.2	构造函数继承	112
3.8.3	重写	113
3.8.4	使用 final 关键字	117
3.8.5	类型检查	119
3.8.6	类型转换	121
3.8.7	使用 AnyObject 和 Any 类型	124
3.9	扩展	125
3.9.1	声明扩展	125
3.9.2	扩展计算属性	126
3.9.3	扩展方法	127

3.9.4 扩展构造函数	128
3.10 协议	130
3.10.1 协议概念	130
3.10.2 协议定义和遵从	130
3.10.3 协议方法	131
3.10.4 协议属性	133
本章小结	135
同步练习	135
第4章 iOS 应用开发概述	138
4.1 HelloiOS 工程	138
4.1.1 创建工程	139
4.1.2 Xcode 中的 iOS 工程模板	141
4.1.3 应用剖析	143
4.2 应用生命周期	145
4.3 iOS API 简介	146
4.3.1 API 概述	146
4.3.2 如何使用 API 帮助	149
本章小结	151
同步练习	151
第5章 iOS 应用界面技术	152
5.1 界面构建技术	152
5.1.1 使用故事板	152
5.1.2 使用 Xib 文件	154
5.1.3 使用代码	155
5.2 MVC 模式	155
5.2.1 MVC 模式概述	155
5.2.2 Cocoa Touch MVC 模式	156
5.3 视图控制器	156
5.3.1 视图控制器种类	156
5.3.2 视图控制器的生命周期	157
5.4 视图与 UIView	158
5.4.1 UIView 继承层次结构	158
5.4.2 视图分类	160
5.4.3 应用界面的构建层次	161
5.5 控件与动作事件	162
5.5.1 按钮	162
5.5.2 定义动作事件	164
5.6 视图与输出口	166
5.6.1 标签	167
5.6.2 定义输出口	167
5.7 视图与委托协议	169
5.7.1 委托设计模式	169
5.7.2 实例: TextField 委托协议	170

5.7.3 键盘的打开和关闭	173
5.7.4 键盘的种类	174
5.8 关闭和打开键盘通知	175
5.8.1 通知机制	175
5.8.2 实例：关闭和打开键盘	176
5.9 Web 视图	177
5.10 工具栏	181
5.11 导航栏	185
本章小结	189
同步练习	189
第 6 章 表视图	191
6.1 表视图中的概念	191
6.1.1 表视图组成	191
6.1.2 表视图的相关类	192
6.1.3 表视图的分类	192
6.1.4 单元格的组成和样式	195
6.1.5 数据源协议与委托协议	196
6.2 简单表视图	197
6.2.1 创建简单表视图	197
6.2.2 自定义单元格	203
6.3 分节表视图	208
6.3.1 添加索引	208
6.3.2 分组	211
本章小结	212
同步练习	212
第 7 章 界面布局与 Auto Layout 技术	214
7.1 iOS 界面布局 UI 设计模式	214
7.2 静态表与表单布局	216
7.3 Auto Layout 布局	220
7.3.1 Interface Builder 中管理 Auto Layout 约束	221
7.3.2 案例：Auto Layout 布局	221
本章小结	228
同步练习	228
第 8 章 屏幕适配与 Size Class 技术	229
8.1 iOS 屏幕的多样性	229
8.1.1 iOS 屏幕介绍	229
8.1.2 iOS 的 3 种分辨率	230
8.1.3 获得 iOS 设备屏幕信息	231
8.2 Size Class 与 iOS 多屏幕适配	233
8.2.1 Size Class 技术	233
8.2.2 Size Class 的九宫格	233
8.2.3 Size Class 的四个象限	234

8.2.4 Xcode 8 中使用 Size Class	236
8.2.5 案例：使用 Size Class	238
本章小结	241
同步练习	241
第9章 应用导航模式	242
9.1 导航概述	242
9.2 导航的“死胡同”——模态视图	243
9.3 平铺导航	252
9.3.1 平铺导航概述	252
9.3.2 使用资源目录管理图片	254
9.3.3 屏幕滚动视图的重要属性	256
9.3.4 分屏导航实现	258
9.4 标签导航	261
9.4.1 标签导航实例	261
9.4.2 标签导航实现	262
9.5 树形结构导航	265
9.5.1 树形结构导航实例	266
9.5.2 树形结构导航实现	267
本章小结	275
同步练习	275
第10章 多媒体开发	276
10.1 音频开发	276
10.1.1 音频文件简介	276
10.1.2 音频 API 简介	277
10.1.3 实例：音频播放	278
10.1.4 实例：音频录制	281
10.2 视频开发	287
10.2.1 视频文件简介	287
10.2.2 视频播放	288
本章小结	290
同步练习	290
第11章 图形图像开发	291
11.1 使用图像	292
11.1.1 创建图像	292
11.1.2 实例：创建 UIImage 图像对象	293
11.1.3 实例：从设备图片库选取或从照相机抓取	296
11.2 使用 Core Image 框架	301
11.2.1 Core Image 框架 API	301
11.2.2 滤镜	302
11.2.3 实例：旧色调和高斯模糊滤镜	303
本章小结	306
同步练习	307

第 12 章 数据存储	308
12.1 数据存储概述	308
12.1.1 沙箱目录	308
12.1.2 数据存储方式	309
12.2 分层架构设计	309
12.2.1 低耦合企业级系统架构设计	310
12.2.2 iOS 分层架构设计	310
12.3 实例: MyNotes 应用	312
12.3.1 采用纯 Swift 语言实现	312
12.3.2 采用 Swift 调用 Objective-C 混合搭配实现	317
12.4 属性列表	318
12.5 使用 SQLite 数据库	322
12.5.1 SQLite 数据类型	323
12.5.2 创建数据库	323
12.5.3 查询数据	325
12.5.4 修改数据	329
本章小结	331
同步练习	331
第 13 章 网络数据交换格式	333
13.1 XML 数据交换格式	334
13.1.1 XML 文档结构	334
13.1.2 XML 文档解析与框架性能	336
13.1.3 实例: MyNotes 应用 XML	337
13.2 JSON 数据交换格式	343
13.2.1 JSON 文档结构	343
13.2.2 JSON 数据编码/解码与框架性能	344
13.2.3 实例: MyNotes 应用 JSON 解码	345
本章小结	346
同步练习	346
第 14 章 网络通信	348
14.1 基于 Web Service 的网络通信技术	348
14.1.1 HTTP 协议	348
14.1.2 HTTPS 协议	349
14.1.3 苹果 ATS 限制	349
14.2 使用 URLSession	350
14.2.1 URLSession API	351
14.2.2 简单会话实现 GET 请求	352
14.2.3 默认会话实现 GET 请求	355
14.2.4 实现 POST 请求	356
14.2.5 下载数据	357
14.3 实例: 重构 MyNotes 案例	360
14.3.1 插入方法调用	360

14.3.2	修改方法调用	362
14.3.3	删除方法调用	362
本章小结		365
同步练习		365
第 15 章 综合案例：奥运会应用设计与开发		366
15.1	应用分析与设计	366
15.1.1	应用概述	366
15.1.2	需求分析	366
15.1.3	原型设计	366
15.1.4	数据库设计	366
15.1.5	架构设计	368
15.2	任务 1：创建工程	369
15.3	任务 2：信息系统层与持久层开发	369
15.3.1	迭代 2.1：编写数据库 DDL 脚本	369
15.3.2	迭代 2.2：插入初始数据到数据库	370
15.3.3	迭代 2.3：数据库版本控制	370
15.3.4	迭代 2.4：配置持久层	371
15.3.5	迭代 2.5：添加 SQLite3 库	371
15.3.6	迭代 2.6：编写实体类	371
15.3.7	迭代 2.7：编写 DAO 类	373
15.3.8	迭代 2.8：数据库帮助类 DBHelper	377
15.4	任务 3：业务逻辑层开发	379
15.4.1	迭代 3.1：编写比赛项目业务逻辑类	380
15.4.2	迭代 3.2：编写比赛日程业务逻辑类	380
15.5	任务 4：表示层开发	382
15.5.1	迭代 4.1：使用资源目录管理图片和图标资源	383
15.5.2	迭代 4.2：根据原型设计初步设计故事板	384
15.5.3	迭代 4.3：“首页”模块	385
15.5.4	迭代 4.5：“比赛项目”模块	385
15.5.5	迭代 4.6：“比赛日程”模块	391
15.5.6	迭代 4.7：“倒计时”模块表示层	393
15.5.7	迭代 4.8：“关于我们”模块表示层	395
15.6	任务 5：收工	396
15.6.1	迭代 5.1：添加图标	396
15.6.2	迭代 5.2：设计启动界面	397
本章小结		397
附录 同步练习参考答案		398

随着移动互联网的发展,移动智能设备越来越多,其中,移动操作系统至关重要。在学习本书之前,有必要介绍一下移动操作系统。

0.1 操作系统的原理与概念

消费类移动智能设备也属于计算机,也需要计算机操作系统,因此读者有必要先了解一下什么是计算机操作系统。

计算机系统主要分为硬件和软件,没有任何软件的计算机称为裸机,裸机不能做任何事情,有了软件,计算机可以处理数据、检索信息和保存数据等。软件又可以分为系统软件和应用软件。系统软件包括操作系统、编译器、编辑器和命令解释器。如图 0-1 所示,计算机的最底层是硬件,没有任何软件,即裸机。裸机之上是操作系统,操作系统屏蔽了底层硬件的复杂结构,为用户提供了一系列操作接口。在操作系统之上是其他系统软件,包括语言编译器、编辑器和命令解释器。系统软件之上是应用软件,例如 Microsoft Word、Microsoft Excel 和 Microsoft PowerPoint 等。



图 0-1 计算机系统的软件和硬件

那么什么是计算机操作系统呢?操作系统是计算机的一种系统软件,管理和控制计算机系统中的硬件及软件资源,为用户提供高效、可靠和安全的服 务。为了更好地理解操作系统的概念,下面从几个不同的角度进行介绍。

0.1.1 隐藏硬件细节

用户不需要了解硬件的技术特性和细节,更不能直接操控硬件。操作系统的引入隐藏了硬件细节,操作系统在裸机之上扩展,使得用户使用计算机更加方便。因此,操作系统的作用是为用户提供一台扩展的计算机,通常称为虚拟机。

用户使用计算机完成日常工作,例如财务人员使用计算机完成财务报销、打印财务报表等,他们不需要了解计算机硬件的内部细节,他们中很多人甚至不知道什么是中央处理器、什么是内存。虚拟机为用户隐藏了硬件细节,例如在保存财务报表到磁盘时,用户不需要知道文件保存到磁盘哪一个扇区,哪一个磁道中,只需要知道文件的保存路径以及文件名称是什么就可以了。

另外,对于应用程序开发人员,他们开发的应用软件,运行在虚拟机上,虚拟机也为他们隐藏了许多底层硬件特性,并提供了开发接口。

0.1.2 资源管理

一台计算机包含很多硬件和软件资源,而操作系统可以管理和控制这些资源,计算机的硬件资源包括处理器、存储器、时钟、磁盘、鼠标、键盘、显示器、网络适配器以及其他一些资源。软件资源包括执行程序和数据,它们往往以文件的形式存储在外存储器(磁盘或闪存)中。操作系统的任务是在程序访问这些资源时,协调各个程序和用户对这些资源的竞争,防止发生冲突,提高资源利用率,满足用户资源请求。

假设有两个程序同时试图对同一个文件进行写入操作,那么哪一个会写入成功呢?操作系统的任务之一就是当资源请求发生冲突时,能够有效地解决这些问题,而对于文件资源,操作系统采用了一种共享锁定机制,防止多个程序同时修改同一个文件。

0.1.3 操作系统的历史

计算机操作系统的发展与计算机体系结构的发展有着密切的关系。计算机的发展就是计算机体系结构的发展,计算机体系结构的进步推动了操作系统的发展和进步。下面按照计算机更新换代的历史介绍操作系统的发展情况。

通常按照计算机电器元件工艺的演变把计算机发展过程分为4个阶段。

1. 第一代计算机(1945—1955): 电子管时代,无操作系统

第二次世界大战结束后,哈佛、普林斯顿和宾夕法尼亚等大学以及一些公司和个人都成功地使用电子管制造出了计算机。当时的计算机非常庞大,需要使用数万个电子管,占地几个房间,但是运算速度不及现在的个人计算机。这些计算机没有操作系统,更没有程序设计语言,只能使用机器语言,程序员在一些插板上进行连线实现想要的控制功能,然后再将插板插入计算机,这样计算机就会根据插板的指令进行计算。直到20世纪50年代早期,出现了穿孔卡片,才不再使用插板,而是将程序写在卡片上,然后读入计算机进行计算。

2. 第二代计算机(1955—1965): 晶体管时代,采用批处理系统

20世纪50年代发明了晶体管,如图0-2所示,同样功能的晶体三极管体积大大缩小。晶体管替代了电子管,不仅体积变小了,稳定性和可靠性也大大地增强。

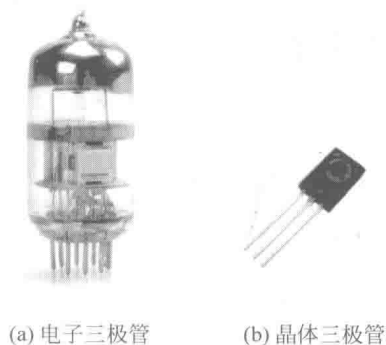


图 0-2 三极管

这个时期的计算机仍然是非常昂贵的,只有少数大公司、大学和重要的政府部门才有。为了防止误操作和有效地管理计算机,专门配置了操作员。这时已经有汇编和 FORTRAN 等语言了,程序员不直接操作计算机,程序员将要完成的一个作业(一个或一组程序)写在纸上,然后用穿孔机制成卡片,最后将这些卡片交给操作员。

为了提高计算机的执行效率,操作员把多个作业集中在一起,批量地处理。收集到一批作业后,将输入磁带装到磁带上,一个作业开始后,计算机从磁带上读取数据并运行,并将输出结果写入到输出磁带。一次作业完成后,操作系统自动地读入下一个作业并执行输出等操作。这一批作业完成后,操作员取下输出磁带,再到另外一台机器上进行脱机打印。第二代计算机主要用于科学计算。

3. 第三代计算机(1965—1980): 集成电路时代,多道程序设计

集成电路是通过一定的工艺,把一个电路中所需的晶体管、电阻和电容等元件及布线互连一起,制作成一小块或几小块晶片。集成电路使电子元件向微小型化、低功耗和可靠性方面迈进了一大步。

在批处理系统中,操作系统将一个作业从磁带读入到内存,然后执行并输出,这个过程中只能进行一个作业,称为单道运行。单道运行时,当处理器读取磁带或进行输入输出操作时,处理器会处于等待状态,因此浪费了很多处理器时间。多道运行可以解决这一问题,多道运行是在内存中同时存放多个作业,然后独立执行。在单处理器系统中,从表面上看,多道运行时载入系统的多个作业同时运行;但实际上还是单个作业轮流使用处理器,交替执行。

4. 第四代计算机(1980—至今): 大规模和超大规模集成电路,分时系统

随着大规模集成电路的发展,每平方厘米的晶片上可以集成上千个晶体管。计算机微型化成为可能,个人计算机时代到来了。

分时技术是多道运行的升级,就是将处理器运行的时间分成很短的时间片段,按照时间片段将处理器轮流分配给各个作业,使得每个作业都能有机会被执行,如果某个作业在分配的时间片段中不能完成,可以在下次时间片段中继续执行。所以从宏观上看,即使是单处理器计算机,每个作业都是同时执行的。