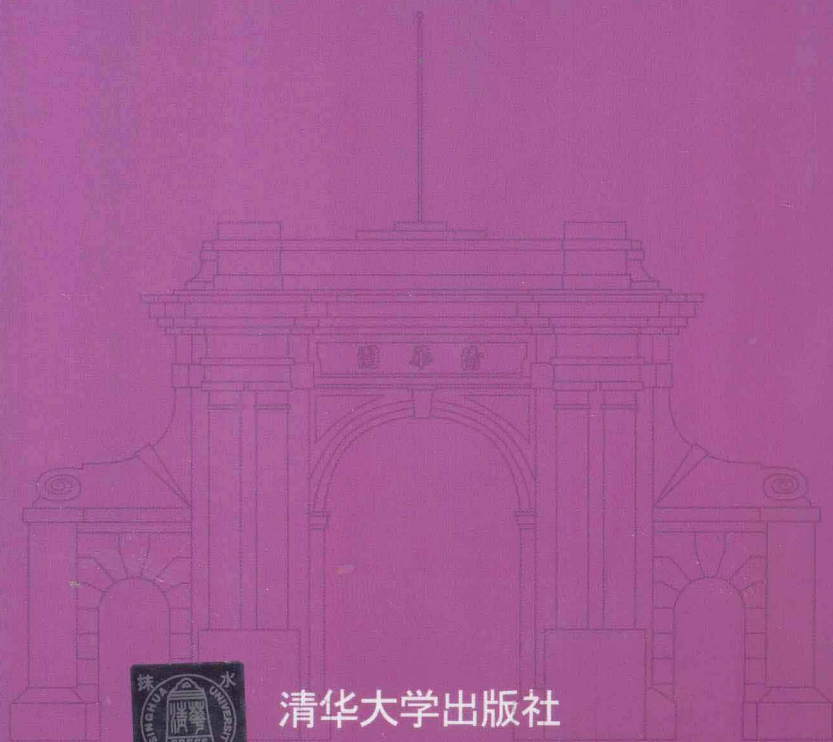


清华大学

计算机系列教材

陈渝 向勇 编著

操作系统实验指导



清华大学出版社

清华大学 计算机系列教材

陈渝 向勇 编著

操作系统实验指导

清华大学出版社
北京

内 容 简 介

本书是操作系统课程的实验教材,旨在帮助读者加强对操作系统原理与设计实现的理解,以分析、设计、改进和实现一个微型但全面的操作系统——ucore 为基本目标,通过增量式地完成 8 个基于 ucore 操作系统实验为操作系统实践环节,最终让读者了解并掌握操作系统的原理、设计与实现。

本书强调对于操作系统动手实践,是对操作系统实践教学的一次探索,可作为高等院校计算机专业操作系统课程的实验教材,也可作为各类操作系统教学的培训教材及自学参考资料。

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

操作系统实验指导 / 陈渝,向勇编著. —北京:清华大学出版社,2013.7

清华大学计算机系列教材

ISBN 978-7-302-32777-6

I. ①操… II. ①陈… ②向… III. ①操作系统—高等学校—教学参考资料 IV. ①TP316

中国版本图书馆 CIP 数据核字(2013)第 136214 号

责任编辑:白立军

封面设计:常雪影

责任校对:梁毅

责任印制:刘海龙

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 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:12.75

字 数:319 千字

版 次:2013 年 7 月第 1 版

印 次:2013 年 7 月第 1 次印刷

印 数:1~2000

定 价:25.00 元

产品编号:053017-01

序

“清华大学计算机系列教材”已经出版发行了 30 余种,包括计算机科学与技术专业的基础数学、专业技术基础和专业等课程的教材,覆盖了计算机科学与技术专业本科生和研究生的主要教学内容。这是一批至今发行数量很大并赢得广大读者赞誉的书籍,是近年来出版的大学计算机专业教材中影响比较大的一批精品。

本系列教材的作者都是我熟悉的教授与同事,他们长期在第一线担任相关课程的教学工作,是一批很受本科生和研究生欢迎的任课教师。编写高质量的计算机专业本科生(和研究生)教材,不仅需要作者具备丰富的教学经验和科研实践,还需要对相关领域科技发展前沿的正确把握和了解。正因为本系列教材的作者们具备了这些条件,才有了这批高质量优秀教材的产生。可以说,教材是他们长期辛勤工作的结晶。本系列教材出版发行以来,从其发行的数量、读者的反映、已经获得的国家级与省部级的奖励,以及在各个高等院校教学中所发挥的作用上,都可以看出本系列教材所产生的社会影响与效益。

计算机学科发展异常迅速,内容更新很快。作为教材,一方面要反映本领域基础性、普遍性的知识,保持内容的相对稳定性;另一方面,又需要紧跟科技的发展,及时地调整和更新内容。本系列教材都能按照自身的需要及时地做到这一点。如王爱英教授等编著的《计算机组成与结构》、戴梅萼教授等编著的《微型计算机技术及应用》都已经出版了第四版,严蔚敏教授的《数据结构》也出版了三版,使教材既保持了稳定性,又达到了先进性的要求。

本系列教材内容丰富,体系结构严谨,概念清晰,易学易懂,符合学生的认知规律,适合教学与自学,深受广大读者的欢迎。系列教材中多数配有丰富的习题集、习题解答、上机及实验指导和电子教案,便于学生理论联系实际地学习相关课程。

随着我国进一步的开放,我们需要扩大国际交流,加强学习国外的先进经验。在大学教材建设上,我们也应该注意学习和引进国外的先进教材。但是,“清华大学计算机系列教材”的出版发行实践以及它所取得的效果告诉我们,在当前形势下,编写符合国情的具有自主版权的高质量教材仍具有重大意义和价值。它与国外原版教材不仅不矛盾,而且是相辅相成的。本系列教材的出版还表明,针对某一学科培养的要求,在教育部等上级部门的指导下,有计划地组织任课教师编写系列教材,还能促进对该学科科学、合理的教学体系和内容的研究。

我希望今后有更多、更好的我国优秀教材出版。

清华大学计算机系教授,中国科学院院士

张钹

前 言

对于在校的学生和工程技术人员而言,能否有效地了解操作系统原理后面的具体设计实现呢?陆游说过:“纸上得来终觉浅,绝知此事要躬行”。我们在教学过程中,也深刻体会到这一点。我们认为,在了解基本的操作系统概念和原理的基础上,通过实际动手来一步一步分析、设计和实现一个微型化的操作系统,会深入了解操作系统的实现细节,并体会到概念原理和实际实现之间的紧密联系及巨大差异。

操作系统是一个复杂系统软件,涉及内容繁多,发展也很快,如 Linux、Windows 等,都是上百万行的源代码规模。开发人员开发这些操作系统软件的目的是用于实际计算机系统中,而不是用于教学,所以直接用这些操作系统来分析了解操作系统的实现和进行操作系统实验会比较复杂。而且目前部分操作系统教材的内容也越来越庞大和抽象,而面向操作系统设计实现的实验部分相对就少了很多。这两方面交织在一起,导致学生了解和掌握操作系统的实际细节很困难。

早期的 UNIX 操作系统实现和 MIT 教授 Frans Kaashoek 等基于 UNIX v6 设计的 xv6 操作系统给了我们启发:对一个计算机专业的本科生而言,在了解操作系统原理的基础上,设计实现一个操作系统有挑战,但是可行!我们对此进行了尝试与探索,以设计实现一个微型但全面的操作系统——ucore 为基本目标,以增量式递进开发方式完成各种基于 ucore 操作系统的实验为实践过程,以在此过程中逐步介绍的操作系统的概念和原理为实践指导,做到有“理”可循和有“码”可查,最终让读者了解和掌握操作系统的原理、设计与实现。目前的实验内容包含如下 8 个。

- (1) 启动操作系统的 bootloader:了解操作系统启动前的状态和要做的准备工作。
- (2) 物理内存管理子系统:理解硬件段/页模式和操作系统如何管理物理内存。
- (3) 虚拟内存管理子系统:理解页表机制、缺页故障处理以及内存替换算法。
- (4) 内核线程子系统:理解相对简单的内核态线程的动态管理过程。
- (5) 用户进程管理子系统:理解用户态进程动态管理过程以及系统调用过程。
- (6) 处理器调度子系统:理解操作系统的调度过程和调度算法。
- (7) 同步互斥与进程间通信子系统:理解进程间如何同步互斥以及进行信息交换和共享。
- (8) 文件系统:理解文件系统的具体实现,与进程管理和内存管理等关系。

其中每个开发步骤都是建立在上一个步骤之上的,就像搭积木,从一个一个小木块,最终搭出来一个小房子。在搭房子的过程中,完成从理解操作系统原理到实践操作系统设计与实现的探索过程。最新的代码和文档放在 http://www.github.com/chyyuu/ucore_lab 上。如果有同学和 OS 爱好者觉得这些实验难度不够,大家可参加更有挑战 and 乐趣的 ucore plus 实验,这些实验位于 http://www.github.com/chyyuu/ucore_plus 下。目前的代码和文档还有许多不完善和错误的地方需要改进,欢迎大家批评指正。

在实现基于 ucore 的操作系统实验过程中,我们参考和借鉴了 xv6、OS161 以及 Linux 的设计思路和实现代码,而且 Frans Kaashoek 博士也亲自给予了帮助与指导。国内多所高校的老师,包括陈向群、王雷、陈鹏、陈莉君、原仓周、蒲晓蓉等都给予了指导和帮助。操作系统课程的助教王乃峥、袁昕颢、茅俊杰、陈宇恒、曹聪、杨杨等完成了大量工作,在此表示衷心的感谢!

陈 渝 向 勇

2013 年 3 月 12 日

目 录

第 1 章 实验 0: 操作系统实验准备	1
1.1 实验目的	1
1.2 准备知识	1
1.2.1 了解 OS 实验	1
1.2.2 设置实验环境	2
1.2.3 了解编程开发调试的基本工具	14
1.2.4 基于硬件模拟器实现源码级调试	23
1.2.5 了解处理器硬件	31
1.2.6 了解 ucore 编程方法和通用数据结构	34
第 2 章 实验 1: 系统软件启动过程	41
2.1 实验目的	41
2.2 实验内容	41
2.2.1 练习	41
2.2.2 项目组成	45
2.3 从机器启动到操作系统运行的过程	48
2.3.1 BIOS 启动过程	48
2.3.2 bootloader 启动过程	48
2.3.3 操作系统启动过程	57
2.4 实验报告要求	66
辅助材料 A 关于 A20 Gate	66
辅助材料 B 启动后第一条执行的指令	68
第 3 章 实验 2: 物理内存管理	70
3.1 实验目的	70
3.2 实验内容	70
3.2.1 练习	70
3.2.2 项目组成	71
3.3 物理内存管理概述	73
3.3.1 实验执行流程概述	73
3.3.2 探测系统物理内存布局	75
3.3.3 以页为单位管理物理内存	75
3.3.4 物理内存页分配算法实现	78
3.3.5 实现分页机制	81
3.3.6 自映射机制	88
3.4 实验报告要求	90

辅助材料 A 探测物理内存分布和大小的方法	90
辅助材料 B 实现物理内存探测	91
辅助材料 C 链接地址、虚拟地址、物理地址、加载地址 以及 edata/end/text 的含义	92
第 4 章 实验 3: 虚拟内存管理	96
4.1 实验目的	96
4.2 实验内容	96
4.2.1 练习	96
4.2.2 项目组成	97
4.3 虚拟内存管理概述	98
4.3.1 基本原理概述	98
4.3.2 实验执行流程概述	99
4.3.3 关键数据结构和相关函数分析	100
4.4 Page Fault 异常处理	102
4.5 页面置换机制的实现	104
4.5.1 页替换算法	104
4.5.2 页面置换机制	105
4.6 实验报告要求	108
辅助材料 A: 正确输出的参考	109
第 5 章 实验 4: 内核线程管理	111
5.1 实验目的	111
5.2 实验内容	111
5.2.1 练习	111
5.2.2 项目组成	112
5.3 内核线程管理	114
5.3.1 实验执行流程概述	114
5.3.2 设计关键数据结构——进程控制块	115
5.3.3 创建并执行内核线程	117
5.4 实验报告要求	122
辅助材料 A 实验 4 的参考输出	123
辅助材料 B “原理”进程的属性与特征解析	124
第 6 章 实验 5: 用户进程管理	127
6.1 实验目的	127
6.2 实验内容	127
6.2.1 练习	127
6.2.2 项目组成	128
6.3 用户进程管理	130
6.3.1 实验执行流程概述	130
6.3.2 创建用户进程	131

6.3.3	进程退出和等待进程	136
6.3.4	系统调用实现	137
6.4	实验报告要求	141
辅助材料 A	“原理”用户进程的特征	141
第 7 章	实验 6: 调度器	145
7.1	实验目的	145
7.2	实验内容	145
7.2.1	练习	145
7.2.2	项目组成	146
7.3	调度框架和调度算法设计与实现	147
7.3.1	实验执行流程概述	147
7.3.2	计时器的原理和实现	147
7.3.3	进程状态	148
7.3.4	进程调度实现	149
7.3.5	调度框架和调度算法	150
7.3.6	Stride Scheduling	154
7.4	实验报告要求	158
辅助材料 A	执行 priority 大致的显示输出	158
第 8 章	实验 7: 同步互斥	160
8.1	实验目的	160
8.2	实验内容	160
8.2.1	练习	160
8.2.2	项目组成	161
8.3	同步互斥的设计与实现	162
8.3.1	实验执行流程概述	162
8.3.2	同步互斥的底层支撑	163
8.3.3	信号量	165
8.3.4	管程和条件变量	167
8.4	实验报告要求	171
辅助材料 A	执行 make run-matrix 大致的显示输出	171
第 9 章	实验 8: 文件系统	173
9.1	实验目的	173
9.2	实验内容	173
9.2.1	练习	173
9.2.2	项目组成	173
9.3	文件系统的设计与实现	176
9.3.1	ucore 文件系统总体介绍	176
9.3.2	通用文件系统访问接口	179
9.3.3	Simple FS 文件系统	179

9.3.4	文件系统抽象层——VFS	183
9.3.5	设备层文件 I/O 层	185
9.3.6	实验执行流程概述.....	189
9.3.7	文件操作实现.....	190
9.4	实验报告要求	193

第 1 章 实验 0：操作系统实验准备

1.1 实验目的

- (1) 了解操作系统开发的实验环境。
- (2) 熟悉命令行方式的编译、调试工程。
- (3) 掌握基于硬件模拟器的调试技术。
- (4) 熟悉 C 语言编程和指针的概念。
- (5) 了解 x86 汇编语言。

1.2 准备知识

1.2.1 了解 OS 实验

编写一个操作系统程序难吗？别被现在上百万行的 Linux 和 Windows 操作系统吓倒。当年 Thompson 在他夫人带着小孩度假留他一人在家时，编写了 UNIX；当年 Linus 还是一个 21 岁大学生时弄出了 Linux 的雏形。站在这些巨人的肩膀上，我们能否也尝试一下做“巨人”的滋味呢？

MIT 的 Frans Kaashoed 教授等在 2006 年左右参考 PDP-11 上的 UNIX Version 6 写了一个可在 x86 上跑的 xv6 操作系统（基于 MIT License），用于学生学习操作系统。Harvard 大学的 David A. Holland 等也设计了 OS161 操作系统用于操作系统实验教学。我们可以站在他们的肩膀上，参考他们的设计思路、方法和源代码，尝试着一步一步完成一个从“空空如也”到“五脏俱全”的“麻雀”操作系统——ucore，此“麻雀”OS 包含软件启动、中断处理、物理内存管理、虚存管理、进程管理、处理器调度、同步互斥、进程间通信、文件系统等主要操作系统内核功能，每个实验包含的内核代码量（C+asm+注释）在 300~10 000 行左右，充分体现了“小而全”的指导思想。

ucore 的运行环境可以是真实的 x86 计算机，不过考虑到调试和开发的方便，我们可采用 x86 硬件模拟器，比如 QEMU、BOCHS、VirtualBox、VMware Player 等。ucore 的开发环境主要是 GCC 中的 gcc、gas、ld 和 MAKE 等工具，也可采用集成了这些工具的 IDE 开发环境 Eclipse-CDT 等。在分析源代码上，可以采用 Scitools 提供的 understand 软件（跨平台），Windows 环境上的 Source Insight 软件，或者基于 emacs+ctags、vim+ctags 等，都可以比较方便地在一堆文件中查找变量、函数定义、调用/访问关系等。软件开发的版本管理可以采用 GIT、SVN 等。比较文件和目录的不同可发现不同实验中的差异性和进行文件合并操作，可使用 meld、kdiff3、UltraCompare 等软件。调试（Deubg）实验有助于发现设计中的错误，可采用 gdb（配合 qemu）等调试工具软件。整个实验的运行环境和开发环境既可以

在 Linux 使用,又可以在 Windows 中使用。推荐使用 Linux 环境。

如何一步一步来实现 ucore 呢? 根据一个操作系统的设计实现过程,可以有如下的实验步骤。

(1) 启动操作系统的 bootloader,用于了解操作系统启动前的状态和要做的准备工作,了解运行操作系统的硬件支持,操作系统如何加载到内存中,理解外设中断和陷阱中断等。

(2) 物理内存管理子系统,用于理解 x86 分段/分页模式,了解操作系统如何管理物理内存。

(3) 虚拟内存管理子系统,通过页表机制和换入换出(Swap)机制,以及故障中断和缺页故障处理等,实现基于页的内存替换算法。

(4) 内核线程子系统,用于了解如何创建相对与用户进程更加简单的内核态线程,如对内核线程进行动态管理等。

(5) 用户进程管理子系统,用于了解用户态进程创建、执行、切换和结束的动态管理过程,了解在用户态通过系统调用得到内核态的内核服务的过程。

(6) 处理器调度子系统,用于理解操作系统的调度过程和调度算法。

(7) 同步互斥与进程间通信子系统,了解进程间如何进行信息交换和共享,并了解同步互斥的具体实现以及对系统性能的影响,研究死锁产生的原因,以及如何避免死锁。

(8) 文件系统,了解文件系统的具体实现,与进程管理等关系,了解缓存对操作系统 I/O 访问的性能改进,了解虚拟文件系统(VFS)、Buffer Cache 和 Disk Driver 之间的关系。

其中每个开发步骤都是建立在上一个步骤之上,就像搭积木,从一个一个小木块,最终搭出来一个小房子。在搭房子的过程中,完成从理解操作系统原理到实践操作系统设计与实现的探索过程。这个房子最终的建筑架构和建设进度如图 1-1 所示。

如果完成上述实验后还想完成更大的挑战实验,那么可以参加 ucore 的研发项目,我们可以完成 ucore 的网络协议栈,增加图形系统,增加编程语言支持(比如目前的 golang、python 等),在 ARM 嵌入式系统上运行,支持虚拟机功能等。这些项目已经有同学参与,欢迎其他有兴趣的同学加入!

接下来将介绍实验环境的设置、Linux 系统的安装、Linux 命令行的使用方法、各种实验工具的使用方法以及 Intel 80386 硬件的重要特征等。这些内容足以写成另外几本书,这里主要是介绍与实验相关的内容并进行了大量的精简,部分内容来源于 Internet(如 Ubuntu forum 网站、qemu 网站、GNU 网站等)和 Intel 的 CPU 手册,由于内容繁多,无法给出具体的参考署名,这里对相关作者一并表示感谢。

1.2.2 设置实验环境

我们参考了 MIT 的 xv6、Harvard 的 OS161 和 Linux 等设计了 ucore OS 实验,所有 OS 实验需在 Linux 下运行。对于经验不足的同学,推荐参考“通过虚拟机使用 Linux 实验环境”一节用虚拟机方式进行实验。

1. 开发 OS 实验的简单步骤

在 github 网站(https://github.com/chyyuu/ucore_pub)可下载我们提供的 lab1 ~



图 1-1 ucore 系统结构图

lab8 实验相关软件和文档,大致经过以下过程就可以完成使用。

- (1) 下载并解压软件包。
- (2) 进入各个 OS 实验工程目录,如执行“cd Code//Lab1”可查看目录下 Lab1 的源码,

执行“cd doc”查看各个实验相关文档。

(3) 根据实验要求阅读源码并修改代码(用各种代码分析工具和文本编辑器)。

(4) 并编译源码,例如执行: make。

(5) 如编译不过则返回步骤(3)。

(6) 如编译通过则可查看运行情况,如执行: make qemu。

(7) 如想测试个人的实验完成是否基本正确,则可执行: make grade。

(8) 如果实现基本正确可查看运行情况(即看到步骤(6)的输出存在不是 OK 的情况)则返回步骤(3)。

(9) 如果实现基本正确(即看到步骤(6)的输出都是 OK)则生成实验提交软件包,例如,执行: make handin。

(10) 把生成的使用提交软件包和实验报告上传或发电子邮件给助教和老师。

另外,可以通过 make debug 或 make debugnox 命令实现通过 gdb 远程调试 OS 实验工程。

2. 通过虚拟机使用 Linux 实验环境(推荐: 最容易的实验环境安装方法)

这是最简单的一种通过虚拟机方式使用 Linux 并完成 OS 各个实验的方法, Linux 操作系统和各种实验所需的开发软件都已经安装到虚拟硬盘文件中了, 只需安装 virtual Box 虚拟机软件就可以开始进行实验了。配置的大体步骤如下。

(1) 安装 VirtualBox 虚拟机软件(有 Windows 版本和其他 OS 版本, 可到 <http://www.virtualbox.org/wiki/Downloads> 下载)。

(2) 在网上下载一个已经安装好各种所需编辑/开发/调试/运行软件的 Linux 实验环境的 VirtualBox 虚拟硬盘文件(oslabs_for_student_2012.zip, 包含一个虚拟磁盘镜像文件和两个配置描述文件, 下载此文件的网址参见 https://github.com/chyyuu/ucore_lab 下的 README 中的描述)。

(3) 用 2345 好压软件(有 Windows 版本, 可到 <http://www.haozip.com> 下载。一般软件解压不了 xz 格式的压缩文件, 也可以用其他支持解压 zip 和 xz 压缩格式的软件)先解压到 C 盘(也可以是其他盘符路径)的 vms 目录下, 即:

```
C:\vms\ubuntu-12.04.vbox.xz
C:\vms\ubuntu-12.04.vmdk.vmdk.xz
C:\vms\ubuntu-12.04.vmdk-flat.vmdk.xz
```

在分别用好压软件或其他能够解压 xz 压缩格式的软件进一步解压上述三个文件, 形成

```
C:\vms\ubuntu-12.04.vbox
C:\vms\ubuntu-12.04.vmdk.vmdk
C:\vms\ubuntu-12.04.vmdk-flat.vmdk
```

解压后这三个文件所占用的硬盘空间为 12GB 左右。在 VirtualBox 中加载 ubuntu-12.04.vbox, 就可以启动并运行 Linux 实验环境了。

启动到提示输入用户名时, 请输入:

```
chy
```

当提示输入口令时,只需按空格键和 Enter 键即可。然后就进入开发环境中了。实验内容位于 ucore_lab 目录下。可以通过如下命令获得放在 github 上的整个实验的最新代码和文档:

```
$git clone https://github.com/chyyuu/ucore_lab.git
```

并可通过如下命令获得以后更新后的代码和文档:

```
$git pull
```

如需要进一步了解一下 git 的基本使用方法,这可以通过网络搜索获得很多这方面的信息。

3. 安装 Linux 实验环境(适合自己安装 Linux 系统的同学)

这里主要以 Ubuntu Linux 12.04(32 位)作为整个实验的系统软件环境。首先需要安装 Ubuntu Linux 12.04,这里主要介绍一种比较容易的 WUBI Linux 的安装方式。

WUBI 方式安装(最容易的 Linux 安装方法)

WUBI 是一个专门针对 Windows 用户的 UBUNTU Linux 安装工具,读者需要做的只是单击几下鼠标而已。不需要改变分区设置,不需要启动文件,不需要 Live CD。使用 WUBI 可很方便地安装或卸载 Ubuntu,如果读者从来没有安装过 UBUNTU Linux,WUBI 很适合初学者第一次安装 UBUNTU Linux。具体方法如下。

(1) 去 OS Course FTP 或官方网站 <http://releases.ubuntu.com/12.04/ubuntu-12.04-desktop-i386.iso> 下载一个 ubuntu-12.04-desktop-i386 的 ISO 文件。

(2) 通过 winrar 等工具将下载来的 ISO 文件中的 wubi.exe 解压出来,放在任意一个分区的根目录下。这里推荐预留了一个至少大小为 8GB 的 NTFS 分区,单击 wubi.exe 安装文件,这时会弹出对话框。

注意:在 ubuntu 12.04 中,“在 Windows 内安装”的那个选项被禁用了,只能通过以下指令开启(假定 X 为光驱盘符):X:\wubi.exe-force-wubi。

(3) 设置好分区将要安装的分区、语言、分配的系统大小、用户名和密码(务必记住)之后,单击“安装”按钮,这时如果正在安装的计算机已经联网了,会自动从镜像网站上下载 ISO 文件。这里采用绕过 WUBI 下载镜像 ISO 的方法安装 Ubuntu 12.04,会节省大量时间。避免下载 ISO 文件的这一步非常关键。在进行这一步之前请将网线断开,然后将提前下载来的 ubuntu-12.04-desktop-i386.iso 文件复制到 WUBI 所创建的 Ubuntu 目录下的 install 文件夹中,重新运行 wubi.exe。这次再也不会提示下载 ISO 文件了。几秒钟后,WUBI 就会提示你重新启动系统。注意,此时 Ubuntu 并没有安装在硬盘上,必须重新启动才开始进行 Ubuntu 12.04 的安装。

(4) 单击“完成”按钮,选择重启计算机。计算机重启后,在启动选项中选择 Ubuntu,出现“press 'Esc' to...”时,不用理会,这时 Ubuntu 滚动条出现在屏幕上。此时,才正式开始安装 Ubuntu 12.04 至硬盘分区某一目录下。接下来什么也不用做,只需等待。当提示正式安装完成后,重新启动计算机系统,可以发现在启动选项中有“Ubuntu”和“Windows”。可以根据读者的情况进行选择。

4. 使用 Linux

在实验过程中,需要了解基于命令行方式的编译、调试、运行操作系统的实验方法。为此,需要了解基本的 Linux 命令行使用。

(1) 命令模式的基本结构和概念。Ubuntu 是当前易于使用和操作的 Linux 发行版。Linux 的命令的操作模式功能可以实现各种功能。简单地说,命令行就是基于字符命令的用户界面,也被称为文本操作模式。绝大多数情况下,用户通过输入一行或多行命令直接与计算机互动。

(2) 如何进入命令模式。假设 Ubuntu Linux 启动后进入图形界面,单击左上角,在提示行输入并回车,从而可以在此软件界面中进行命令行操作。

打开 `gnome-terminal` 程序后可能会注意到类似下面的界面:

```
chy@ chyhome-PC:~$ls
file1.txt file2.txt file3.txt tools
```

“`chy@ chyhome-PC:~$`”这些字符串被称为命令终端提示符,它表示计算机已经就绪,正在等待着用户输入操作指令。以作者的屏幕画面为例,`chy` 是当前所登录的用户名,`chyhome-PC` 是这台计算机的主机名,`~`表示当前目录。此时输入任何指令按 `Enter` 键之后该指令将会提交到计算机运行,比如可以输入命令: `ls` 再按下 `Enter` 键:

```
ls [Enter]
```

注意: `[Enter]`是指输入完 `ls` 后按下 `Enter` 键,而不是输入这个单词,`ls` 这个命令将会列出当前所在目录里的所有文件和子目录列表。

下面介绍 Bash Shell 程序的基本使用方法,它是 Ubuntu 默认的外壳程序。

1) 常用指令

(1) 查询文件列表: `ls`。

```
chy@chyhome-PC:~$ls
file1.txt file2.txt file3.txt tools
```

`ls` 命令默认状态下将按首字母升序列出当前文件夹下面的所有内容,但这样直接运行所得到的信息是比较少的,通常它可以结合以下这些参数运行以查询更多的信息。

① `ls /`: 将列出根目录“/”下的文件清单。如果给定一个参数,则命令行会把该参数当作命令行的工作目录。换句话说,命令行不再以当前目录为工作目录。

② `ls -l`: 将列出一个更详细的文件清单。

③ `ls -a`: 将列出包括隐藏文件(以“.”开头的文件)在内的所有文件。

④ `ls -lh`: 将以 KB/MB/GB 的形式给出文件大小,而不是以纯粹的 Bytes。

(2) 查询当前所在目录: `pwd`。

```
chy@chyhome-PC:~$pwd
/home/chy
```

(3) 进入其他目录: `cd`。

```
chy@chyhome-PC:~$pwd
/home/chy
chy@chyhome-PC:~$cd /root/
chy@chyhome-PC:~$pwd
/root
```

上面的例子中,当前目录原来是/home/chy,执行 cd /root/之后再运行 pwd 可以发现,当前目录已经改为/root了。

(4) 在屏幕上输出字符: echo。

```
chy@chyhome-PC:~$ echo "Hello World"
Hello World
```

这是一个很有用的命令,它可以在屏幕上输入指定的字符串(" "中的内容),可用于信息提示和输出相关内容等要求。

(5) 显示文件内容: cat。

```
chy@chyhome-PC:~$ cat file1.txt
Roses are red.
Violets are blue,
and you have the bird-flue!
```

也可以使用 less 或 more 命令来显示比较大的文本文件内容。

(6) 复制文件: cp。

```
chy@chyhome-PC:~$ cp file1.txt file1_copy.txt
chy@chyhome-PC:~$ cat file1_copy.txt
Roses are red.
Violets are blue,
and you have the bird-flue!
```

这个命令可以复制一个文件的内容到另一个文件中。

(7) 移动文件: mv。

```
chy@chyhome-PC:~$ ls
file1.txt
file2.txt
chy@chyhome-PC:~$ mv file1.txt new_file.txt
chy@chyhome-PC:~$ ls
file2.txt
new_file.txt
```

这个命令可以简单理解为一个文件改名字。

注意:在命令行模式进行操作时,系统基本上不会给丰富的提示信息,当然,绝大多数命令可以通过加上一个参数-v 来要求系统给出执行命令的反馈信息。

```
chy@chyhome-PC:~$ mv -v file1.txt new_file.txt
'file1.txt' -> 'new_file.txt'
```

(8) 建立一个内容为空的文本文件: touch。

```
chy@chyhome-PC:~$ ls
file1.txt
chy@chyhome-PC:~$ touch tempfile.txt
chy@chyhome-PC:~$ ls
```