

黑客道的九个段位是对“3*3”（或者 32）教学计划进度的描述。它把黑客道讲授的三门语言各自分为三个阶段，这样就得到了九个不同而渐进的层次：

1. 8086 汇编 汇编语言 王爽著
2. C 语言入门
3. C 指针的编程艺术
4. 80386 保护模式的汇编
5. Linux 内核算法分析
6. Scheme 语言入门
7. 编写 Scheme 解释器
8. 编写 C 编译器
9. 编写 BIOS

完成了某个阶段学习的学员即可升段，进入后一个阶段的深造。

“黑客道”中的九段是如何评定的？

“黑客道”学员的九个段位的评定完全可以具体化和流程化，即与“黑客道”教学流程的九个阶段对应起来相挂钩。

黑客道的段位评判标准非常讲究可操作性，一切以学员自己开发提交的软件作品出发，尽量做到客观公正：

* 初段：掌握了 Intel x86 上实模式（8086/8088）的编程技术者

评判标准：能否理解计算机的体系结构、利用汇编语言编写一些简单的算法。

* 二段：通过了 C 语言编程入门者

评判标准：能否熟练运用基本的 C 句法，编写一些小型的 UNIX 实用程序的克隆版本。

* 三段：掌握了 C 指针变量的实质和运用（各种基本数据结构和通用算法）者

评判标准：能否利用 C 语言开发一种新型算法。

* 四段：掌握了 Intel 80386 保护模式汇编技术

评判标准：能否采用 C 和汇编代码混合编程，编写一种设备驱动程序。

* 五段：掌握了 Linux 内核核心算法者。

评判标准：能否从裸机开始，从源代码安装 GNU/Linux 操作系统；能否修改、定制 Linux（或者 FreeBSD）内核设计中的算法（例如，能否在内核中增加一个新的文件系统）。

* 六段：理解计算本质者，打牢哲理和数理基础，掌握了 Scheme 语言。

评判标准：能否利用 Scheme 开发 CGI 程序。

* 七段：可以在 x86 硬件上实现 Scheme 求值器

评判标准：能否利用 C 和汇编在 x86 上写出符合 R5RS 规范的 Scheme 解释器。

* 八段：能利用 Scheme 和汇编代码写出 C 编译器者。

评判标准：能否在 x86 上编写出符合 C99 规范的 C 编译器。

* 九段：可以开发 BIOS 程序者。

评判标准：能否自行设计印刷电路板，利用各种芯片和芯片组构造新的计算机系统，并开发出该系统的 BIOS。

上面的每一个段位的评定都是容易实现的，主要的评判标准在于学员能否提出自己开发的相应软件作品。黑客道学员的段位在洪峰评定后可以应学员要求公布在本网站上。

黑客道培训课程简介

更新时间：2007 年 3 月 20 日

汇编语言

第一阶段：Intel x86 上的汇编语言(8086/8088)。（黑客道初段）

1. 各种进位的数字系统与转换（二进制、八进制、十进制、十六进制）
2. 汇编器的两大基本任务
3. 8086 的 14 个寄存器和寻址机制
4. 一道程序的组成部分
5. 代码段的寻址
6. 数据段的寻址
7. 堆栈段的操作和应用
8. 构造控制结构
9. 把汇编代码模块化
10. 中断与跳转
11. 字符串的操作
12. 浮点计算的实现
13. 多媒体数据的处理
14. 文件的操作

15. 外部设备的控制与数据交换

16. 从汇编代码中调用系统调用

时间：一个月。

主教材：《x86 汇编语言入门》。

参考资料：《Intel 80386 处理器手册》

第一阶段：C 语言编程入门。（黑客道二段）

1. 基本数据类型
2. 数据类型的转换与赋值语句
3. 通过键盘和屏幕与计算机交互
4. 两类分支控制结构
5. 四种循环控制结构
6. 数组与字符串
7. 指针和数组
8. 函数
9. 变量与函数的作用域
10. 各种运算符和结合级的优先级
11. 数据的抽象：结构、共用体与枚举
12. 位运算与位域操作
13. 文件的读写
14. 编译预处理、C 编译器的选项

15. C 的标准库与库函数

16. C 程序的调试

时间：一个月。

主教材：《解读 C99》。

第二阶段：C 指针编程的艺术。（黑客道三段）

指针变量是 C 语言的灵魂，没有理解指针变量的人，实际上没有掌握 C 语言。洪峰讲解这一专题时，通常采用了 Scheme 与 C 对比的方法，使学员可以很快进入角色，掌握 C 语言的核心内容。

学员结束本课程的学习后，还应该掌握完整的 GNU Building 系统。

1. 什么是指针与指针变量？
2. 字符数组与字符串：指针与数组的关系
3. 指针操作之一：赋值
4. 指针操作之二：检索
5. 指针操作之三：交换
6. 利用指针构造数据结构（之一：向量、双端队列、列表）
7. 利用指针构造数据结构（之二：集合、多集、哈希、字典）
8. 利用指针构造数据结构（之三：堆栈、队列、优先队列）
9. 指针变量与位运算
10. 指针与内存的管理
11. 回调函数（callback functions）及其应用

12. 深入解剖 printf 函数
13. 解读其他 C99 的标准库函数与 GNU C Library 函数
14. 算法的效率分析：指针变量在常用算法中的应用
15. 高级专题：C++ STL 中的智能指针
16. 总复习：Unix 终端设备的驱动程序分析

时间：两个月。

主教材：《C 指针编程的艺术》。

辅导材料：《解读 C99》、《Scheme 之道》。

第二阶段：x86 体系上保护模式汇编。（黑客道四段）

1. 80386 体系设计的总体思想
2. 80386 的寻址机制：分段和分页保护机制
3. 关于新增的段寄存器
4. 通用寄存器用途与宽度扩展
5. 32 位的标志寄存器
6. 系统状态与状态寄存器的编程
7. 系统段的结构与编程
8. 代码段的结构
9. 数据段与堆栈段
10. 中断门与陷阱门
11. 可编程中断控制器的编程

12. 任务门与调用门
13. GDT 与 LDT 的编程
14. 多任务的切换
15. 80386 汇编代码的调试
16. 总复习：现代操作系统内核的设计

时间：一个月。

主教材：《x86 汇编语言入门》。

辅导材料：《Intel 80386 处理器手册》、《计算机体系结构》。

第三阶段：C 语言与 UNIX 内核中的算法实现。（黑客道五段）

本课程分析 Linux 2.6.20 内核中的算法。Linux 内核主要代码是采用 C 语言编写的，其中的各种算法是分析 C 语言应用的极好素材。同时，黑客道学员通过分析 Linux 内核，可以很快地熟悉这一平台，为今后的应用程序开发打下良好的基础。熟悉了 Linux 的学员，可以把得到的经验推广应用到 FreeBSD 内核的学习上去。

1. UNIX 系统的总体设计思想
2. x86 的保护模式与实模式
3. 系统的启动和关机
4. 内核空间与用户空间
5. 程序执行环境分析
6. 系统调用内幕

7. 系统时钟与内核定时
8. 信号
9. 进程管理算法
10. 文件系统算法
11. 内存管理算法
12. 进程间通信的算法
13. 设备驱动程序的开发（包括 C 语言与汇编语言混合编程）
14. 网络编程模块
15. 内核中的动态模块
16. 内核源代码包的 Makefile 文件分析与内核定制

时间：两个月。

主教材：《C 指针编程的艺术》。

辅导教材：《Scheme 之道》、《x86 汇编语言入门》。

Scheme 语言

Scheme 入门。（黑客道六段）

Scheme 是编程语言王国的女王。黑客道的入门阶段教授这一功能超级强大的算法语言，黑客道学员修炼入段的学习内容已经模块化，分为以下十个单元。

1. 基本数据类型与类型转换
(Primitive Data Types and Conversions)
2. 符号表达式和 lambda 表达式

(S-exp and lambda expressions)

3. 词法定界和块结构

(Lexical scoping and block structure)

4. 递归和尾递归

(Recursion and Tail-recursive functions)

5. 输入和输出，系统接口

(Input and Output, System Interface)

6. 高阶算子和准引用

(High-order operators, quasi-quotation)

7. 宏、结构、OOP

(Macros, Structure, and OOP)

8. 惰性求值和流

(Lazy-evaluation and Streams)

9. 连续、当前连续、连续传递风格和跳转

(Continuation, call/cc, CPS, and Jumps)

10. 非确定性计算与人工智能程序设计

(Non-deterministic computing and other AI topics)

时间：一个月。

在完成上面的十个单元后，学员应该完成以下六个单元的练习作为项目实习：

1. 建立 CGI 编程环境

2. 读写 CGI 的环境变量

3. Web Form. 数据的解码
4. 动态生成网页
5. 数据的保持与更新
6. 总复习：构造一个完整的 CGI 应用程序

时间：一个月。

预备知识：学员应该掌握了 GNU Emacs 编辑器的基本用法，并且可以在 GNU/Linux 或者 FreeBSD 系统上进行基本的操作。英语水平应该达到可以熟练阅读和写作的程度。

主教材：《Scheme 之道》。

辅导材料：《Scheme 语言百问》、《泛系导论》、《泛系数学导论》。

构造 Scheme 语言解释器。（黑客道七段）

学习这一课程的学员需要先学习和掌握 C 语言，x86 汇编语言以及 x86 体系硬件平台。这一课程教学一般要等到学习了 C 和汇编语言之后进行。

1. 总论：Scheme 求值器的组成
2. 基本数据类型的构造
3. 一些基本的例程
4. 数据路径与控制器
5. C 运行时环境的构造
6. C 代码块和汇编代码块的接口
7. 偶对与链表
8. 特殊表的实现

9. 连续作为第一等对象
10. 小结：构造所有基本的 Scheme 对象
11. 堆栈 (stack) 与栈框 (stack frame) 的实现
12. 利用寄存器读写结构化的数据
13. 高阶算子：apply 与 eval 构成的元循环
14. 正则序的引入和实现
15. 内存垃圾收集与算法实现
16. 总复习：构造 Scheme 解释器

时间：三个月。

主教材：《Scheme 语言百问》。

辅导材料：《Scheme 之道》、《C 指针编程的艺术》、《x86 汇编语言入门》。

利用 Scheme 语言和汇编代码编写 C 编译器。（黑客道八段）

编译原理对于任何计算机专业学生的重要性是众所周知的。不懂编译原理者，实际上没有学会计算机技术的核心内容，编译原理课程也是计算机专业教学中难啃的“硬骨头”。

这门课程是洪峰所有教授课程中非常经典和精彩的内容，洪峰教授这门课程

时，采用三兼顾的方法，融合哲理、数理和技理于一体，指导学员采用

Scheme 语言和汇编代码编写一个实际可以运行的 C 编译器，实际可操作性非常高，是编译原理教学实践方面的巨大进步和质的飞跃。

1. 数学、语言、逻辑
2. 三大基本的数学结构
3. 图论基础、乔姆斯基的结构语言理论
4. 符号表、出错表的构造
5. 正则表达式
6. 状态、操作、目标
7. 自动机、有限自动机、非确定有限自动机
8. 词法分析
9. 句法分析（从上到下）
10. 句法分析（从下到上）
11. 语义分析
12. 运行时系统的结构
13. 中间代码生成
14. 代码优化技术
15. 目标代码生成
16. 总复习：C 编译器的构造

时间：四个月。

主教材：《Scheme 之道》。

辅导材料：《泛系数学导论》、《C 指针编程的艺术》、《x86 汇编语言入门》、《解读 C99 》。

学习这一课程的学员需要先学习和掌握 Scheme 语言和 C 语言，以及 x86 体系硬件平台。

第三阶段：信号的分析与综合。（黑客道九段）

这门课程主要针对来自计算机硬件专业的系统程序员开设，学员应该具备模拟电路、数字电路和计算机体系结构方面的基础。

1. 物理定律、传感器、计算机外部设备
2. 电路元件与泛导方程
3. A/D 与 D/A 中的数学建模
4. 系统函数、频率响应、数字模拟
5. 时钟与尺度、总线与泛通
6. 拉氏变换、模拟系统、微分方程
7. z 变换、数字系统、递归方程
8. 卷积
9. 傅立叶级数与傅立叶积分
10. 流的处理、信号的并发
11. 模拟与数字信号的综合
12. 硬件描述语言、FPGA 编程、编译器
13. 中断的性质、设计及其重构
14. 寄存器的设计与组织

15. 系统的启动和停机

16. 总复习：BIOS 的设计

时间：两个月。

主教材：《信号的分析与综合》

辅 导材料：《模拟电路入门》、《数字电路入门》、《Verilog 入门》、《泛
系数学导论》、《收敛级数与发散级数》、《差分方程与时序分析》、

《Fourier 级数讲义》、《小波分析讲义》、《计算机体系结构》、《MIPS
体系结构图解》、《ARM 体系结构图解》。

C 语言

C 语言是系统程序开发的首选，是黑客道教学课程中的重要环节。在设计 C 语言时，许多设计特性均考虑到了简化编译器编程的要求，这一工程实践本身就是“强化简化抓关键”的极好案例。

本课程要求学员会使用 GNU/Linux 系统，会使用 GNU Emacs 编辑器开发程序。课程结束时，学员应该掌握 C 程序的开发和调试技术。