

赖于课程大纲的要求、个人喜好、学生的背景和能力。图中的课程从左往右越来越强调以程序员的角度来看待系统。以下是简单的描述。

- **ORG**：一门以非传统风格讲述传统主题的计算机组成原理课程。传统的主题包括逻辑设计、处理器体系结构、汇编语言和存储器系统，然而这里更多地强调了对程序员的影响。例如，要反过来考虑数据表示对 C 语言程序的数据类型和操作的影响。又例如，对汇编代码的讲解是基于 C 语言编译器产生的机器代码，而不是手工编写的汇编代码。
- **ORG+**：一门特别强调硬件对应用程序性能影响的 ORG 课程。和 ORG 课程相比，学生要更多地学习代码优化和改进 C 语言程序的内存性能。
- **ICS**：基本的 ICS 课程，旨在培养一类程序员，他们能够理解硬件、操作系统和编译系统对应用程序的性能和正确性的影响。和 ORG+ 课程的一个显著不同是，本课程不涉及低层次的处理器体系结构。相反，程序员只同现代乱序处理器的高级模型打交道。ICS 课程非常适合安排到一个 10 周的小学期，如果期望步调更从容一些，也可以延长到一个 15 周的学期。
- **ICS+**：在基本的 ICS 课程基础上，额外论述一些系统编程的问题，比如系统级 I/O、网络编程和并发编程。这是卡内基-梅隆大学的一门一学期时长的课程，会讲述本书中除了低级处理器体系结构以外的所有章节。
- **SP**：一门系统编程课程。和 ICS+ 课程相似，但是剔除了浮点和性能优化的内容，更加强调系统编程，包括进程控制、动态链接、系统级 I/O、网络编程和并发编程。指导教师可能会想从其他渠道对某些高级主题做些补充，比如守护进程(daemon)、终端控制和 Unix IPC(进程间通信)。

图 2 要表达的主要信息是本书给了学生和指导教师多种选择。如果你希望学生更多地

章号	主题	课程			
		ORG	ORG+	ICS	ICS+ SP
1	系统漫游	●	●	●	●
2	数据表示	●	●	●	○ (d)
3	机器语言	●	●	●	●
4	处理器体系结构	●	●		
5	代码优化		●	●	●
6	存储器层次结构	○ (a)	●	●	○ (a)
7	链接			○ (c)	○ (c)
8	异常控制流			●	●
9	虚拟内存	○ (b)	●	●	●
10	系统级 I/O			●	●
11	网络编程			●	●
12	并发编程			●	●

图 2 五类基于本书的课程

注：符号○表示覆盖部分章节，其中：(a)只有硬件；(b)无动态存储分配；(c)无动态链接；(d)无浮点数。

ICS+是卡内基-梅隆的 15-213 课程。