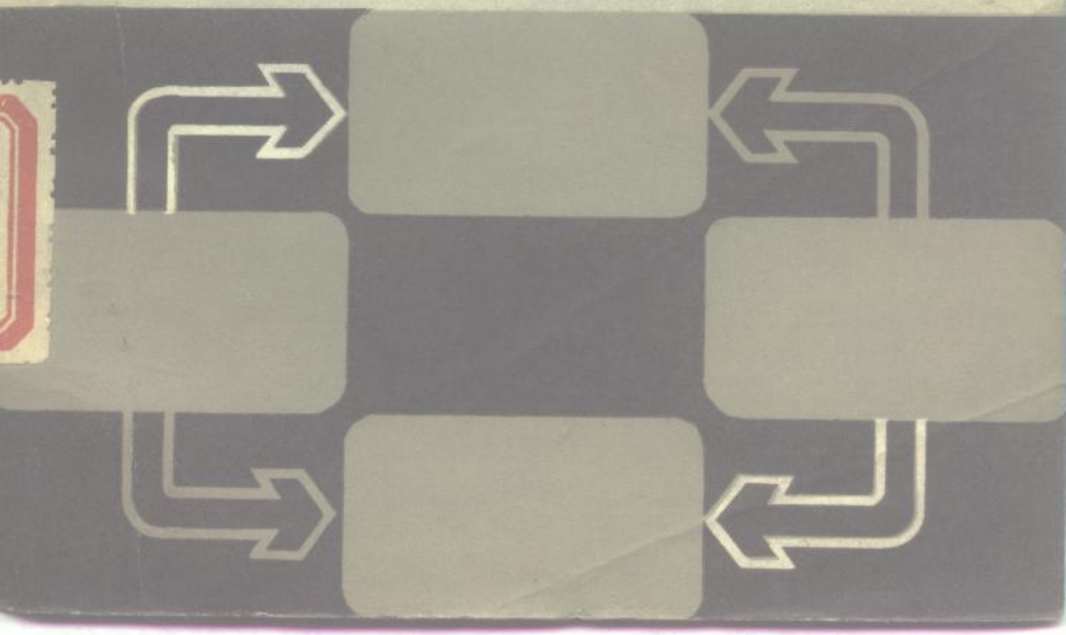
软件移植

SOFTWARE·PORTABILITY·

P. J. 布朗 等 著

科学出版社



软件移植

P. J. 布朗 等 著

朱关铭 章幼义 瞿泽方 译

何国森 校

科学出版社

内 容 简 介

本书较全面地介绍了软件移植的基本原理、主要技术以及一些重要的研究成果。

本书首先讲述了移植的基本概念，随后介绍了编译程序的移植、抽象机硬件的实现和有关软件移植实用化的一些研究，最后叙述了软件的保护以及软件移植的一些研究成果和实例。

本书可供计算机软件工作者和有关专业人员学习参考。

P. J. Brown etc.
SOFTWARE PORTABILITY
Cambridge University Press 1977.

软 件 移 植

P. J. 布朗 等 著
朱关铭 章幼义 瞿泽方 译
何国森 校
责任编辑 杨家福 刘晓融
科学出版社 出版
北京朝阳门内大街 137 号
中国科学院印刷厂印刷
新华书店北京发行所发行 各地新华书店经售

1982 年 12 月第 一 版 开本：787×1092 1/32
1982 年 12 月第一次印刷 印张：1⅝ 5/8
印数：0001—6,000 字数：285,000

统一书号：15031·457
本社书号：2896·15—8
定价：1.95 元

序　言

本书是以1976年3月26日至4月9日在坎特伯雷市肯特大学举办的软件可移植性讨论会的高级教程为基础的。该讨论会是在欧洲经济共同体(EEC)科学和技术研究委员会(CREST)的信息训练小组的赞助下，由科学研究委员会主办的。其教程先由始终参加讲课的讲授人员提供一份完整的讲授大纲，然后根据特约论文增补充实。

该讨论会的教程指导：

E. B. 斯普拉特，P. J. 布朗

英国坎特伯雷肯特大学计算实验室

课程的讲授者：

P. J. 布朗

肯特大学计算实验室

M. 格里菲斯

法国南锡大学

R. E. 格里斯沃尔德

美国亚利桑那大学计算机科学系

H. W. 劳森

瑞典林彻平大学系统技术学院

B. 尼布利特

肯特大学斯旺西学院计算机科学系

M. 理查兹

剑桥大学计算机实验室

W. M. 威特
美国科罗拉多大学电机工程系
B. A. 威奇曼
英国国家物理实验所数值分析和计算部

教程特约论文的作者：

R. 奥尔伍德
英国拉夫巴勒技术大学民用工程系
D. M. 布兰登
美国高级计算机技术
B. 福特
牛津数值算法小组
G. R. 弗兰克
曼彻斯特大学计算机科学系
J. R. 格罗根
英国国际计算机有限公司
J. 英格利斯，P. J. H. 金
伦敦大学比克贝克学院计算机科学系

JS101/1/1

目　录

第Ⅳ部分 实 用 化

第Ⅴ部分 法 律 方 面

第Ⅵ部分 实例研究

第Ⅶ部分　研究和未来

第Ⅷ部分　两项研究

第Ⅰ部分　引　　言

这篇引言叙述了本书的写作动机和结构.

第Ⅰ.A章　本书的结构

E. B. 斯普拉特　　P. J. 布朗

(英国　坎特伯雷　肯特大学)

一本关于软件移植的教程是很有价值的,其理由很多,最主要的是:

(a) **经济的理由**. 当前巨额的资金被花于把程序从一台计算机搬到另一台计算机.

(b) **硬件的发展**. 当前硬件发展的速度极快,大部分硬件都正在不断地用新的取代老的.

(c) **网络**. 地区间的、全国性的以及国际间的网络的出现,增大了对可移植性,尤其是对数据的可移植性的需要.

由于这些原因,科学研究委员会(the Science Research Council)倡导把全世界有关这个题目的研究成果集中起来,编写成教程. 1976 年 1 月,八位作者汇集在一起,筹划编写这本教程,并打算在 1976 年春季把该教程提交给一个为期两周的讨论会征集意见,然后修改润色,最后发表. 那个会开得很好,新的教程也极为成功,本书就是在那个会上经过共同努力所取得的最后成果. 我们希望这本书既可以作为软件移植的入门读物,又可以是一本有用的参考书.

本教程的作者们很清楚地知道,对于不同的人来说,软件

移植有不同的含意．也就是说，FORTRAN 语言的程序编写者有一种理解，抽象机的设计者有第二种理解，保存着大量数据文件的保险公司有第三种理解，硬件的设计者有第四种理解，而法学家有第五种理解．把这些考虑综合起来就形成了这样一本内容丰富的教程．

这本教程还包括一些有关可移植性的实际方案的研究．这类内容的大部分是由特邀作者提供的．

总的来说，这本教程分为七个部分，这篇引言就是其中的第一部分．第二部分介绍了可移植性的基本概念．第三部分讲的是工具，它的范围从高级语言之间的翻译一直到抽象机的硬件实现．第四部分研究软件移植的实用化．在第二部分和第四部分，有许多章节虽然是从基本内容讲起的，可是最后都谈到了作者本人在这些课题上的最新工作成果．第五部分讲的是法律方面的问题，这个问题对于移植软件已越来越显得重要了．第六部分是一些实例的研究．第七部分叙述了当前两项重要的研究计划．最后的第八部分介绍了两项可移植性研究的结果，一项是由英国和法国的科学研究委员会完成的，另一项是由欧洲经济共同体（EEC）完成的．这一部分，以及第五部分中的法律问题和第六部分中的实例研究，在内容上是独立的，可单独阅读而不牵涉其它的部分．

这本书整个材料的组织吸取了 Bill Waite 的建议，而对于叙述方式的规定则主要根据 Ralph Griswold 的意见．最初参加讨论这本教程的人提出了大量有意义的问题，对出版这本书的准备工作有很大的好处．另外，从以前出版的具有类似性质的书中，特别是保存在慕尼黑的许多关于编译程序构造和软件工程的书中，也吸取了不少有益的东西．

第Ⅱ部分　基本概念

这部分介绍了可移植性的理论的概况，并且比较了目前实现移植的一些有用的方法.

第Ⅱ.A章　理　　论

W. M. 威特

（美国　科罗拉多大学）

建立一种理论的目的在于提供一个共同的概念，可以用它来解释和推测某种现象的各个方面. 因此，可移植性的理论就应该让我们能够解释为什么某些程序比其它的更容易移植，以及应该让我们能够预测在移植某个给定的程序时会产生些什么困难. 不幸的是，就象大多数工程学科一样，对于软件的可移植性也不存在完美无缺的、无所不包的理论.

缺乏完整的理论并不等于说我们对构成可移植程序的基础的概念一无所知. 可移植性是与程序的语义以及它们如何进行表达紧密相联系的，因此有理由设想把程序设计语言的语义方面的理论结果应用到可移植性问题上去. 在这一章里我将讲述如何应用其中的一些结果来设计判定方法，它对决定一个软件的可移植性是极为重要的.

对软件移植有兴趣的人主要可分为两大类：一些人希望用现有的工具来写可移植的程序，另一些人希望创造新的工具. 在这一章里我将几乎完全谈前一类人关心的问题. 我告诉他们一定要很好地选择合适的工具，并仔细地检查它们.

基本的语言概念知识将帮助他们把注意集中在最关键的地方；我将设法指出这些关键的地方，并说明要去解决什么问题．[语义概念的完整处理参见(Pratt, 1975)，它已超出了本书的范围．]应用这些概念来设计中间语言将在第Ⅶ.A章里讨论．

1．原　　语

一种语言的原语就是不能够用那种语言的概念来解释的基本单元．每一种语言都有这种基本单元：平面几何中的“点”，英语中的“是(to be)”，FORTRAN语言中的“实型数据”．原语只能用直觉、经验、信念以及不包含这个原语的某种形式系统来加以定义．例如，FORTRAN语言中的“实型数据”就定义为“对实数值的近似处理”[美国国家标准协会(ANSI)，1966]．这个定义就借助了我们的经验(“近似处理”)，也借助了形式数学(“实数值”)．

从根本上说，一种语言的语句的意义决定于它所包含的原语的意义．当我们在一台新机器上实现一个程序的时候，我们也就根据这台机器的概念提供了这个程序所包含的原语的定义．如果整个程序的“意义”要保持不变，那末各原语的“意义”首先就必须保持不变．这一点可能容易做到，也可能不易做到；它既取决于原语的原始“意义”，也取决于目标计算机所提供的便利．在这里我将着重叙述在各种程序设计语言中都要出现的原语，并说明它们的定义是如何影响程序的可移植性的．

1.1．基本对象

我们先从一组不加限制的值开始，值也就是能够加以运

算的抽象实体．**模态**根据值上可能的运算，把值加以分类．对象就是值的一个具体例子；如果某个值是一种语言的原语，那末这个值就称为**基本对象**．让我们来考察能定义原语值的性质的方法，以及这些定义对可移植性有何影响．

基本模态 **Boolean** 通常用**枚举**的办法来加以定义(Naur, 1963)：

"<逻辑值>::=**true**(真)|**false**(假)"

Character 是另一类到处碰到的表示值的有限集合的模态．为了能够用各种目标计算机的硬件字符集来表示字符值，大部分的语言定义都为这些字符值给出了最少的说明．可移植程序的编写者必须使程序不决定于字符值的性质，因为字符值并不是由语言定义来明确保证的．例如COBOL[美国国家标准协会(ANSI)，1974]规定了有 51 个字符的字符集，但是它并不保证这些字符在目标计算机上都是可用的．

PASCAL 是用两份文件来作定义的，一份是"用户手册"，一份是"报告"(Jensen and Wirth,1974)．"报告"简单地陈述了字符值"决定于具体的实现"；而"用户手册"则进一步宣称，下述最低限度的假设保持与实现无关：

"字符集包括

(1) 按字母次序排列的大写拉丁字母 A···Z 的集合

(2) 按数字次序连接着的十进制数字 0···9 的集合

(3) 空格符．"

我的经验认为这些假设是有道理的，必须涉及字符的可移植程序的编写者，大多数情况都可以相对可靠地作这些假设．(要特别注意在三组字符之间没有定义它们的相互关系，并且要注意字母集合并不需要是紧致的．)

无限集合，例如那些由模态 **integer** 和 **real** 所描述的无限集合在计算机里必须用有限的集合来近似表示．这些近似表