国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第13卷第2期 1991年6月 Vol. 13 No. 2

面向对象计算机系统若干问题的研究

朱海滨

（电子计算机系）

摘 要 现有面向对象的软件环境大都建立在传统的 Von Neumann 型计算机结构上，其执行效率很低。传统系统是无法支持面向对象风格高效实现的。很有必要开发新型的体系结构以支持这种良好的程序设计风格。本文从介绍面向对象程序设计的基本思想入手，建立面向对象思想与计算机系统的联系，阐述有关研究问题并给出初步的解决办法。

关键词 计算机系统，对象，面向对象，面向对象程序设计

分类号 TP33

面向对象的程序设计思想被认为是80年代的结构程序设计，以该思想为基础的程序设计语言也相继问世，并显示出强大的生命力。Smalltak-80，POOL，Actor，Orient-84 等都是此类语言的优秀代表。现有的面向对象软件环境大都建立在传统的 von Neumann 型计算机结构上，其执行效率很低。从面向对象风格的特点来看，传统的系统结构是无法支持面向对象风格的固有并发性等特征的，因而大大限制了系统效率的提高。很有必要开发新型的体系结构以支持这种良好的程序设计风格。现在国际上已经出现了许多支持特定面向对象环境的计算机系统，如支持 Smalltalk-80 语言的基于 RISC 技术的 SOAR 系统、基于 32 位微处理器的 Sword32 系统、还有 SWAMP 系统；支持 POOL 语言的并行结构等等。本文旨在建立面向对象思想与计算机系统之间的联系，阐述有关的研究问题并给出初步的解决办法。

1 面向对象程序设计的基本思想

面向对象的程序设计方法以信息隐蔽和抽象数据类型概念为基础，把系统中所有资源，如数据、模块以及系统都看成对象；每个对象把一个数据类型和一组过程封装在一起，这组过程可以对这一数据类型进行处理，并在定义对象时可以规定外界对其操作的权限。这一方法直接而自然，可以使设计人员把主要精力放在系统一级上，而对细节问题可以较少地关心。

1989年7月5日收稿

一般认为，面向对象＝数据抽象＋信息隐蔽＋继承性。

可以给出如下描述性定义：

定义 1 对象：对象定义为一个三元组＜MS，DS，MI＞，其中，

MS 是对象受理的操作集合；

DS 是对象的存贮或数据结构；

MI 是对象受理的操作名集（即消息名集，也称对外接口）。

定义 2 消息：消息定义为一个二元组＜消息名，变元组＞，其中，

消息名∈∪MI；

变元组::＝＜变元，变无，…，变元＞；

变元::＝对象。

定义 3 类：类定义为一个四元组＜INH，DD，OI，ITF＞，其中，

INH 是类的继承性描述；

DD 是数据结构描述；

OI 是操作集合的具体实现；

ITF 是统一的对外接口。

从上面的定义可以看出，对象是对操作表示和数据表示两种功能的抽象．当 DS→Φ 时，对象→一组纯过程；当 MS→Φ 时，对象→一个纯数据结构。如果真是这样，对象就成为废品一堆，无任何作用。所以，对象使得数据及相应的操作不可分，实现了数据封装的功效。

类是对一种对象的抽象，它将该种对象所具有的共同特征(包括操作特征和存贮特征)集中起来，由该种对象所共享。在系统构成上，则形成了一个具有特定功能的模块和一种代码共享的手段。

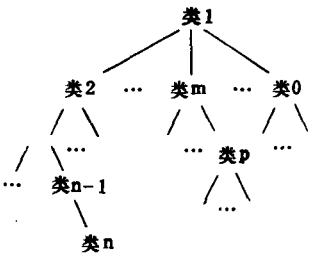

图 1

定义 4 继承：类实现中的继承机制是一种独特的性质，它使得子类可以继承其前辈类的特征和能力。类的继承具有传递性：IF (C2 INH C1) & (C3 INH C2) THEN C3 INH C1 其中 INH 表示“继承于”(如图 1 所示，其中类 m、类 o、对类 p 构成多重继承关系，即，类 p 多重继承于类 m 和类 o)。

为了清晰起见，类可以重新定义如下：

类 n:: ＝＜$\bigcup_{i=1}^{n}$ 类 i 本身的数据结构描述，$\bigcup_{i=1}^{n}$ 类 i 本身的操作实现，

$\bigcup_{i=1}^{n}$ 类 i 本身的对外接口＞

面向对象的概念具有以下特征：

(1) 固有并发性：对象是相对独立的，知道如何工作（How to do）的实体；消息发送是启动对象工作的唯一途径。各个对象的工作可以相互独立，因此具有天然的并发性。

(2) 数据流与控制流的统一：对象操作的启动是由其它对象发送消息来控制的，消息的发送又总是携带某些对象的，因此，控制信息和数据信息是一起到来的，从而统一了数据流和控制流的思想。

(3) 动态连接：即对象的功能执行是在消息传送时确定的，这样，提高了程序设计的灵活性，可以用专门的硬件来提高其执行效率。

(4) 局部存贮与分布式计算：即每个对象通过数据抽象和信息隐蔽将其内容和状态置于自身独立的存贮结构中，并且，对象的处理也是自治的，整个系统的运算和处理是分布式的。

2 面向对象系统中的问题

2.1 基本拓扑结构的考虑

通过以上的分析，要支持对象功能的并发执行，必须建立多个处理单元的硬件环境。可以设想采用主辅机结构，主机是通用计算机，辅机由多个具有专门硬件的处理单元组成，每个单元具有一般的运算功能、局部存贮器和专用功能部件，并配以适当的互联网络（如图 2）。

2.2 专用硬件的设置

为说明清楚起见，先给出几个定义：

定义 5 方法：在面向对象的程序中是指实现对象功能的程序段。

定义 6 源对象：指发出消息的对象。

定义 7 目的对象：指接收某一源对象发出的消息的对象。

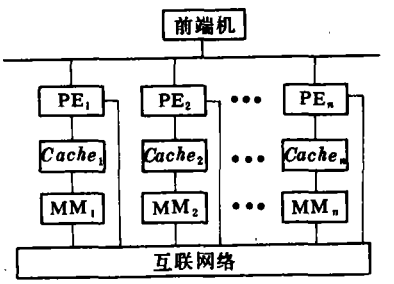

图 2 基本的拓扑结构

定义 8 语景：在面向对象系统中是指对象在处理一个方法时所必需的工作环境。它也是一种特殊的对象，其状态指示对象的工作情况。

(1) 方法查找：面向对象系统的大量工作是查找对象方法，因此，特别需要有专门硬件机制（比如高速相联比较部件）支持高速的方法查找，使对象快速地响应消息。

方法查找与继承性表示是紧密相关的。继承性表示的好坏也直接影响着效率的高低。在软件实现中一般采用指针链的形式；而在 VLSI 高度发展的今天，可以采用表格存贮方式进一步加快方法的查找；如果是采用相联存贮器，则可以将继承性表示与方法查找中比较部件结合在一起，在硬件支持下，可以较容易地高效实现多重继承。

(2) 消息的发送与接收：由于面向对象系统的执行中，消息发送是激活对象的主要手段，所以有必要设置专用部件支持消息的快速发送与接收。

消息的传递一般具有以下四个基本阶段：

1）发送：必须保证源对象所发出的消息能够到达目的对象；

2）接收：必须保证目的对象能够收到源对象发给它的消息；

3）受理：目的对象正确地处理源对象发给它的消息；

4）回答：目的对象根据源对象的要求决定是否给以答复。

由于受理和回答两步的工作具有很大的灵活性，所以，这里的硬件只能支持消息的发送和接收，而受理和回答则由操作系统和应用程序负责。

(3) 支持消息处理的硬件机制 .

语景是对象的基本工作环境。每当对象要处理一个外部发送来的消息时，都要先为其建立语景。语景作为一种特殊对象，要准确地反映对象的工作情况，应设有以下内容(如图 3 所示)：

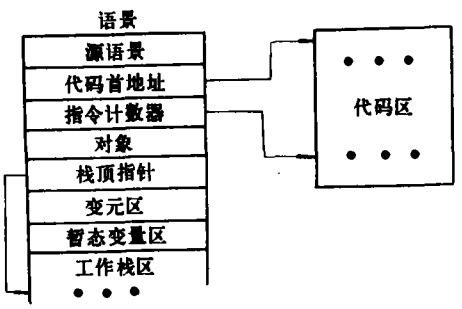

图 3 对象的工作环境

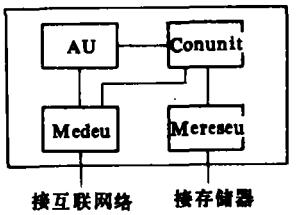

图 4 处理单元功能框图

AU 是支持一般计算功能的运算部件；

Conunit 是寄存当前语景的大量寄存器阵列；

Medeu 是处理消息发送与接收的部件；

Mereseu 是消息到代码的映射部件，即方法查找部件。

1）源语景：用于指出引发该方法语景的方法语景，以备返回时用；

2）指令计数器：记录引发一个新语景时代码的位置；

3）栈顶指针：记录引发一个新语景时工作栈的栈顶；

4）代码首地址：用于指出该语景对应代码段；

5）对象：指出该语景对应的主体对象；

6）暂态变量区：用于存放对应代码所需的暂态变量；

7）变元区：存放引发该语景的消息变元；

8）工作栈区：是后进先出（LIFO）栈，用于对象的运算工作。

以上的语景需要大量的寄存器(或高速缓冲栈)的支持，另外，为了高速起见，建立语景和语景切换应该有专门的指令支持，如支持成批数据的传送功能等。

根据以上的讨论，可以初步给出每个处理单元的功能框图（如图 4)。

2.3 Cache 的设置问题

针对于面向对象程序执行特征，应分别设置两种不同的高速缓冲部件：其一是对象 Cache，用于支持对象的快速识别，确定对象所在的处理单元；其二是消息 Cache，用于支

持快速的响应消息。在设置 Cache 中，同样会遇到不一致的问题，但如果采用分布主存，一个对象只存在于一个单元的存贮器中，则可以避免这一问题。

2.4 互联网络

在面向对象的系统中，互联网络也是一个重要的环节. 互联时应特别注意消息的正确传递。最直观的是采用交叉开关，这样可以使消息与对象的联系比较直接，从而达到快速的目的。当然在实现中，应当考虑性能价格比的合理性，可以采用其它更为合适的结构。

2.5 消息的处理与对象的状态

处理机的个数是有限的，不可能在每个消息一到来就马上得到响应，需要根据处理机的调度情况来处理。另外，在一个消息到来时有可能目的对象正在处理上一个消息。因此，该消息还不能马上得到响应，所以，为每个对象配备一个消息队列是必要的，也就是说，对象的状态要包括消息队列。

根据执行情况的不同，一个对象所处的基本状态有休眠、就绪和执行三种状态，另外，还有可能出错，其转换如图 5 所示。

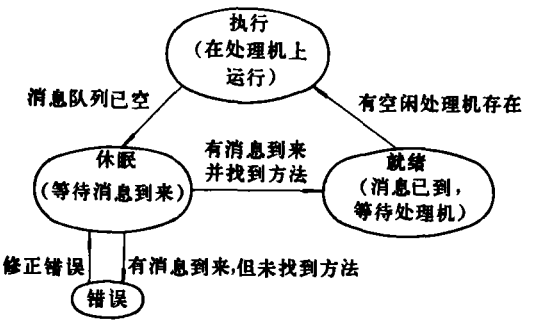

图 5 对象状态的转换

消息的处理和回答的实际处理应由应用程序确定，系统所作的工作主要就是对象运行状态和消息队列的管理。

2.6 处理机的调度问题

最大限度地发挥各个处理单元的能力，是面向对象系统的难点和重点。对象到处理单元的适当映射显得非常重要，特别是在分布主存的体系结构中，对象映射的适当与否将直接影响到整个系统的执行效率。比如，对象在处理过程中应最大程度地在所处处理单元对应的存贮器中查找方法，否则，不同处理单元之间的消息传递将会大大影响系统的执行效率。对象在处理消息时，按指数增长不断启动新的对象，这样就使许多不相关的对象可以同时运行。这也就是为什么要设置快速语景切换部件的原因。

3 结束语

以上就与开发支持面向对象程序设计的系统有关的问题作了探讨，并给出了初步的解决办法，希望能对开发面向对象的系统有所促进。笔者认为，以上所提出的问题是需要首先解决的问题，并且其中的每一个问题都有待于进一步研究和付诸实现。笔者愿意与对此问题感兴趣的同仁进行讨论和交换意见，以使问题能得到更好的解决。

参 考 文 献

[1] Adele Goldberg and David Robson, Smalltalk-80: The language and Its Implementation. Addison-Wesley, Reading, Mass., 1983

[2] Geoffery A. Pascoe. The Elements of Object-Oriented Programing. BYTE, Feb., 1986

[3] Alan Snyder. Encapsulation and Inheritance in Object-Oriented Programing Languages. OOPSLA'86 Proceedings

[4] 朱海滨. 对 Smalltalk-80 系统的分析及其核心环境的实现. 硕士学位论文，国防科技大学，1988

[5] Takuo Watanabe and Akinori Yonezawa. Reflection in an Object-Oriented Concurrent Language. OOPLSA'88 Proceedings

[6] Edward H. Bensley, et al. An Execution Model for Distributed Object-Oriented Computation. OOPSLA'88 Proceedings

On the Research of Some Issues in Object-Oriented Computer Systems

Zhu Haibin

(Department of Computer Science)

Abstract

The classical von Neumann architecture has its limits and can not make the object-oriented programs run effectively. So it is necessary to develop a new kind of architecture to express this style's advantages. Some relations between the object-oriented programming style and the computer system are demonstrated after the idea of object-oriented programming is illustrated and defined. From the relations, some problems are discussed and primary solutions about the problems are given.

Key words computer system, object, object-oriented, object-oriented programming