



[12] 发明专利说明书

专利号 ZL 99805004.0

[45] 授权公告日 2005 年 7 月 27 日

[11] 授权公告号 CN 1212563C

[22] 申请日 1999.4.8 [21] 申请号 99805004.0

[30] 优先权

[32] 1998. 4. 15 [33] US [31] 09/060,702

[86] 国际申请 PCT/US1999/007690 1999.4.8

[87] 国际公布 WO1999/053395 英 1999.10.21

[85] 进入国家阶段日期 2000.10.12

[71] 专利权人 微软公司

地址 美国华盛顿州雷德蒙德

[72] 发明人 詹姆斯·爱德华·沃尔什

本杰民·阿伦·拉迪克-古尔德

审查员 王京霞

[74] 专利代理机构 北京银龙知识产权代理有限公司

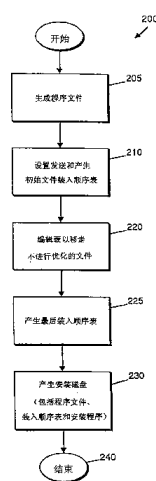
代理人 皋吉甫

权利要求书 4 页 说明书 21 页 附图 5 页

[54] 发明名称 改进程序发送时间的系统和方法

[57] 摘要

以改进程序发送时间的方式在硬盘驱动器上安装文件。在完成编程后，发送程序，并监视与磁盘加强操作相关的磁盘活动，以便确定在程序或命令发送中从磁盘上读取文件部分的顺序。该数据用于产生装入顺序表，该装入顺序表指示在发送中读取的文件各个部分的顺序。安装盘包括文件和装入顺序表。在安装过程中，安装程序从装入顺序表读数据，并写文件部分，使其以由装入顺序表描述的顺序存储在硬盘驱动器上的邻近簇中。计算机然后以从邻近磁盘簇的正确顺序从磁盘上读发送相关数据，这减少和消除由于为了读发送相关数据而使磁头在非邻近簇之间移动的磁盘存取造成的时间浪费。



1. 一种从安装盘向磁盘安装程序代码的方法，包括步骤：

产生包括发送相关部分的多个文件；

确定程序发送期间读出的发送相关部分的顺序；

产生指示读发送相关部分的顺序的装入顺序表；以及

在安装盘上存储多个文件、装入顺序表和安装程序；

安装程序包括计算机可执行指令，用于完成以下步骤：

识别磁盘上的至少一组邻近自由簇；以及

在所述至少一组邻近自由簇中，从安装盘安装多个文件的发送相关部分，所述发送相关部分最初以所述装入顺序表指示的顺序存储在磁盘上。

2. 如权利要求 1 所述的方法，识别至少一组邻近自由簇的步骤包括：

从存储在磁盘上的数据读一系列可用的自由簇；以及

识别具有大量邻近自由簇的邻近自由簇组。

3. 如权利要求 1 所述的方法，其中多个文件包含非发送相关部分，在安装所述多个文件的所述发送相关部分的步骤后，将所述非发送相关部分存储在磁盘上。

4. 如权利要求 1 所述的方法，其中识别在磁盘上的至少一组连续自由簇的步骤包括：

决定识别的邻近自由簇但是否是磁盘上的第一自由簇组，以及

如果不是，就写入一个临时文件到需要填入的自由簇上，以得到识别的邻近自由簇组。

5. 如权利要求 1 所述的方法，其中识别在磁盘上的至少一组邻近自由簇的步骤包括：

决定识别的邻近自由簇是否是磁盘上的第一自由簇组，以及
如果不是，就写入一个临时文件到需要填入的自由簇上，以得到识别的邻近自由簇组。

6. 如权利要求 1 所述的方法，安装多个文件的发送相关部分的步骤包括：

从储存在安装盘中的装入顺序表中读多个项，每一个项识别多个文件中的一个和要读的文件部分；

对每一个从顺序表中读的项，从安装盘读出通过该项识别的文件部分；

对每一个从顺序表中读的项，将读出的文件部分写入磁盘上的邻近自由簇组。

7. 如权利要求 6 所述的方法，其中装入顺序表识别在程序发送中读多个发送相关文件部分的顺序。

8. 如权利要求 6 所述的方法，其中装入顺序表中的每个项包含文件识别符以及从识别文件读的数据位置。

9. 如权利要求 6 所述的方法，其中装入顺序表中的每个项包含文件名、命名文件的偏置，以及从命名文件读的数据量。

10. 一种在逻辑盘上安装和优化程序代码的方法，包括步骤：

产生包括与磁盘加强操作相关部分的多个文件；

在逻辑盘上安装多个文件之前，确定在磁盘加强操作中读磁盘加强的顺序；

产生装入顺序表，所述装入顺序表指示在磁盘加强操作中读磁盘加强部分的顺序；以及

在安装盘上存储多个文件、装入顺序表和安装程序；

从安装盘向该逻辑盘安装多个文件和装入顺序表；
在逻辑盘上安装后，通过完成以下步骤优化磁盘加强部分：
识别逻辑盘上的至少一组邻近自由簇；以及
以装入顺序表中指示的顺序对磁盘加强部分重新排序。

11. 如权利要求 10 所述的方法，其中装入顺序表包含多个项，每个项包含多个文件中的一个的名称以及要重新排序的文件部分的名称。

12. 如权利要求 10 所述的方法，其中识别在磁盘上的至少一组邻近自由簇的步骤包括：

决定识别的邻近自由簇是否是磁盘上的第一自由簇组，以及
如果不是，就写入一个临时文件到需要填入的自由簇上，以得到识别的邻近自由簇组。

13. 如权利要求 10 所述的方法，其中通过从文件的开始端的偏置和从文件读到的数据量中识别所述部分。

14. 如权利要求 13 所述的方法，其中对磁盘加强部分重新排序的步骤包含步骤：

读装入顺序表中的项，该项识别多个文件中的一个以及要读取的文件部分；

读从识别的多个文件中的一个文件的文件部分；
在逻辑盘上写所述的文件部分。

15. 从安装盘将程序码安装到磁盘上的方法，其中，操作系统力图将文件保持在邻近簇中，包括步骤：

产生多个包括发送相关部分的文件，其中，发送相关部分以非邻近簇方式写入；

决定发送相关部分读入的次序；

产生负载顺序表，表示发送相关部分读入的次序；和

在安装盘上储存多个文件，负载顺序表和安装程序；

安装程序包括计算机可执行的执行步骤指令，包括

识别至少一组磁盘上的邻近自由簇；

写入一个临时文件以占据至少邻近自由簇的一个自由簇，以及迫

使每个文件的发放相关部分以负载表指明的次序写进非邻近簇，；

以及

在至少一组邻近自由簇内安装多个文件的发送相关部分，该发送相关部分最初是由负载表指明的次序储存在磁盘上。

16. 根据权利要求 15 的方法，其中，在至少一组邻近自由簇上安装多个文件的发放相关部分，包括：

从储存在安装盘的负载次序表中读多个项，每一个项识别多个文件中的一个以及读出的文件中的一部分；

对每一个从顺序表中读的项，从安装盘读出通过该项识别的文件部分；

对每一个从顺序表中读的项，将读出的文件部分写入磁盘上的邻近自由簇组。

改进程序发送时间的系统和方法

本发明涉及在硬盘驱动器上存储数据，特别涉及在程序启动和其它磁盘增强操作时以减少从磁盘读数据所需时间的方式存储数据。

在用于存储计算机数据的磁盘驱动器的上下文中，术语“碎片”指的是文件在磁盘分区或逻辑盘上的非邻近排列。碎片文件可以指在逻辑盘的不同区域上散布的相同磁盘文件部分，或者指整个文件的散布，从而使文件之间存在无用空间。当从逻辑盘删除旧文件和添加新文件时，常常出现碎片。当删除文件时，以前使用的空间就成为可用空间，计算机操作系统在逻辑盘的该空区域内写入新的文件。但是，如果新文件太大而在该空区域内写不下，操作系统可通过在几个空区域内存储该文件的部分而将该文件变成碎片，这些空区域散布在整个逻辑盘上。同样，如果产生的文件太大，超过了与存储该文件的空间相邻的邻近自由空间的大小，则产生的文件便变成碎片。操作系统保持文件每部分存储于何处的记录，因而当从该逻辑盘上读取该文件时，可以以正确的顺序恢复该文件。

在从开始至结束“线性”读取文件的情况下，碎片降低了磁盘的存取速度并降低了整个磁盘操作的特性。提供应用程序以在形成碎片的磁盘上整理碎片。除重新对文件排序以便将每个文件存储在磁盘的邻近部分上外，这些程序通常对文件排序使得这些文件都存储在逻辑盘的开始端。这样在磁盘末端的形成了合并空间，并认为这是有益的，因为操作系统不需要为了寻找存储新文件的足够空间而进行长时间搜索，新文件更可能存储在邻近簇（cluster）中（即非碎片的），而不是散布在整个文件的大量较小的簇组中。

碎片整理是有好处的，它提高了从开始端至末端线性读取文件的特性。但是，当执行一些操作系统时，计算机以一些可预测的序列从几个不同文件恢复数据块，常常只从每个文件读取几个块。例如，发送字处理程序可以导致装入可主执行的动态连接库的第一部分、可主执行的优

先文件的第二部分等。当选择一些程序命令（如“打开文件”）时，会发生类似的磁盘操作。这些类型的磁盘加强操作可以导致大量磁盘头从一个物理簇移动到另一个物理簇。因为现代微处理器与磁盘驱动器的平均搜索时间相比是极快的，为了发送典型程序必需从磁盘上读取许多不同文件，发送程序花费了大量时间，使微处理器处于等待磁盘头从一个簇移动到另一个簇的空闲状态。

根据这些观察，当执行磁盘加强操作（如发送程序或发送与一些命令相关的代码），以便以在操作中的存取顺序来对在逻辑盘分区（以及物理磁盘）上的磁盘加强操作中使用的数据块排序。这会导致程序文件设计成“碎片”，并通过减少从逻辑盘装入所需数据的无序磁盘存取的数量而提高特性。

为了优化应用程序发送过程，现有技术公开了该类型应用文件的重构和碎片整理。例如，IBM 技术公开公报中顺序号为 TDB01960112 的文件公开了一种配置文件(profile)引导文件系统的内容。为取代根据关于文件参考模式的静态假定而排序的文件，当程序运行并时，系统动态地确定文件存取模式，并分配磁盘空间以最大化程序运行的每个硬件/软件结构的文件系统特性。为了实现上述功能，当启动程序时，监视文件系统活动，并使存取的磁盘扇区记录包含配置文件数据的文件。以预定的时间间隔，分析配置文件数据以便确定是否可能有更优文件分配结构。如果是，则操作者可以运行重构程序，该重构程序根据配置文件数据指示的磁盘存取模式给文件分配磁盘空间。程序会映射出从配置文件数据到其相关文件的磁盘块存取地址，并在磁盘上将频繁存取的块物理的放置在一起。所获得的磁盘块分配结构优化特定计算机配置的公用存取模式。

微软公司还宣布“WINDOWS 98”操作系统具有这样的特性，即使其在应用程序发送和重新排序存储文件中监视磁盘活动，从而以顺序形式对在启动中使用的数据块排序。

尽管现有技术公开了优化用于程序启动的文件的方法，但是这些技术使在程序安装在计算机上并随后由用户发送后执行的。该过程具有几个缺点。例如，直到程序安装和发送以及已经监视到启动过程后才能进行

优化。在程序启动过程中监视磁盘存取过程会大大降低启动速度。在监视了启动过程后，计算机必须运行费时优化例行程序，该程序防止用户在优化过程中使用计算机，或需要用户离开计算机，以便运行几小时这样的例行程序。最后，该过程只适于监视和优化启动过程，而不适于其它磁盘加强操作，如装入与一些命令（如“打开”）有关的代码。

在操作系统软件上实现优化特性，用户为利用这些优化特性本身，必须获得和安装操作系统合适版本的拷贝。在很多情况下，用户不愿在采用新性能的操作系统软件升级上花费钱或时间，或者是在公司中用户自己不能作决定。因此使大量用户使用老版本操作系统，而不能获得在应用程序启动上的改进特性。

因此，本领域需要改进的系统和方法，目的是以在应用或命令发送时进行存取的顺序在逻辑盘上存储数据。该方法应使其在安装到逻辑盘上后不需监视应用程序文件，并且也不要求用户申请和安装新版本的操作系统软件。

本发明满足上述需要，通过对程序安装或以后的重构，使发送相关的部分以尽可能接近在典型发送中读取的顺序存储，该部分是在发送程序或执行命令时读取的。在安装中或安装后使用的该软件包含必要的程序文件和装入顺序表（load sequence list），其识别读文件的与发送相关部分的顺序。在安装和以后的重排序中，文件的发送相关部分以装入顺序表中指示的顺序排序，发送相关部分最好存储在逻辑盘上的邻近簇中。剩下的文件安装在磁盘上的其它自由空间中。当用户运行程序时，由于计算机能以正确顺序从逻辑盘上的邻近簇中读取启动文件，因而会减少启动时间，由此缩短由于从非邻近簇读取数据而造成的延迟。

一般说来，本发明提供计算机可读取安装媒体，并以加速启动或发送程序或其它磁盘加强操作过程的方式来安装程序或在以后对程序重新排序。安装媒体包含具有发送部分的多个文件和装入顺序表。装入顺序表指示要在逻辑盘上排序的文件发送相关部分的顺序。该安装媒体还包含计算机可执行的指令，用于执行安装或重新排序步骤。安装或重新排序包含识别在逻辑盘上的至少一组邻近自由簇。多个文件的发送相关部分

安装在一组邻近自由簇中，使发送相关部分以装入顺序表指示的顺序存储在硬盘驱动器上。

另一方面，本发明提供一种从安装盘向硬盘驱动器安装程序代码的方法。该方法包含从存储在安装盘上的装入顺序表读取项(entry)的步骤。每个项识别与程序代码有关的文件部分。对于从装入顺序表读取的每个项，从安装盘上读取由项识别的文件部分，并将其写到硬盘驱动器的一组邻近自由簇上。该文件部分以对应装入顺序表中的项的顺序存储在硬盘驱动器上。

另一方面，本发明提供一种产生程序代码并将其从安装盘安装在逻辑盘上的方法。该方法包含产生包含发送相关部分的文件和确定在程序发送或其它磁盘加强操作中对发送相关部分的读取顺序。产生装入顺序表以反映在程序发送中读取发送相关部分的顺序。该文件、装入顺序表和安装程序存储在安装盘中。该安装程序包含计算机可执行指令，用于识别逻辑盘上的至少一组邻近自由簇，并安装在一组或多组邻近自由簇中的多个文件的发送相关部分。发送相关部分以装入顺序表中指示的顺序存储在逻辑盘上。

另一方面，本发明提供一种安装和优化逻辑盘上的程序代码的方法。该方法包含步骤：产生包含与磁盘加强操作有关的部分的多个文件和确定在磁盘加强操作中读取磁盘加强部分的顺序。产生装入顺序表以指示在磁盘加强操作中读取磁盘加强部分的顺序。多个文件和装入顺序表安装在逻辑盘上。在安装后，通过识别逻辑盘上至少一组邻近自由簇并对自由簇中的磁盘加强部分重新排序而优化磁盘加强部分。磁盘加强部分以装入顺序表中指示的顺序存储在逻辑盘上。

根据对下述实施例的详细说明和参考附图与权利要求，可以更清楚理解本发明的各个方面。

图1是为本发明一个实施例提供操作环境的个人计算机的方框图；

图2是流程图，显示了获得在逻辑盘上安装文件并以在程序发送中装入的顺序来存储文件的安装盘的举例性方法；

图3是流程图，显示了从安装盘向逻辑盘安装文件并以在程序发送中

装入的顺序来存储文件的举例性方法；

图 4 是显示举例性装入顺序表的内容的表格；

图 5 是显示启动相关的三个文件的部分以及这些部分装入顺序的表格；

图 6 是显示将文件部分写到逻辑盘上的自由簇的举例性方法的表格；

图 7 是显示将文件部分写到逻辑盘上的自由簇的举例性方法的表格；

图 8 是流程图，显示另一种在将文件安装到逻辑驱动器上后的优化文件部分的方法。

本发明目的是提供用于在硬盘驱动器上安装程序的改进的系统和方法。在一个举例性实施例中，应用程序相关文件安装在逻辑盘分区上（即物理磁盘驱动器的逻辑子集），使发送相关部分以在发送中读取的顺序存储，该发送相关部分是在发送程序或选择命令时读取的。

安装软件包含必要文件和装入顺序表，装入顺序表识别在发送中装入的文件的发送相关部分的顺序。在安装过程中，在发送中使用的文件部分以装入顺序表中指示的顺序安装，发送相关部分最好存储在逻辑盘上的邻近簇中。剩下的文件安装在逻辑盘上的其它自由空间中。另外，可以以普通方式安装程序文件，随后以装入顺序表所指示的顺序对发送相关部分重新排序。当用户发送程序或选择命令时，由于计算机能以正确顺序从逻辑盘上的邻近簇中读取发送相关部分文件，因而缩短了发送时间，由此减少了由于从非邻近簇读取数据而延迟的时间。

下面参照附图说明本发明的的各方面和举例性的操作环境，其中相同标号表示各图中相同的部分。

图 1 和下述说明试图提供对实施本发明的合适计算机环境的简洁而概括的说明。在安装程序的概括上下文中描述的本发明用于在与个人计算机相连的硬盘驱动器上安装应用程序，本领域的技术人员会知道本发明还可结合各种程序模块来实现，这些程序模块包含应用程序模块、操作系统模块、实用程序模块和类似的程序模块。一般地，程序模块包含例程序、程序、元件和数据结构等，用于完成特定任务或处理特定数据类型。此外，本领域的技术人员能理解，本发明可以和其它计算机系

统配置一起使用，这些计算机系统配置包含手持式装置、多处理器系统、基于微处理器或可编程的用户电子设备、微型计算机、主计算机和类似装置。本发明还可用于散布式计算环境，通过由通讯网络连接的远程处理器来完成任务。在散布式计算环境中，程序模块可以位于本地和远程内存存储器上。

参照图 1，实现本发明的系统实施例包含普通个人计算机 20，个人计算机 20 包含处理器 21、系统内存 22 和将系统内存与处理器 21 相连的系统总线 23。系统内存 22 包含只读存储器(ROM)24 和随机存取存储器(RAM) 25。基本输入/输出系统 26 (BIOS) 存储在 ROM24 内，BIOS 26 包含帮助在个人计算机 20 内的各组成之间传送信息的基本例行程序，如在启动中。个人计算机 20 还包含硬盘驱动器 27、可换式磁盘驱动器 28 和光盘驱动器 30，可换式磁盘驱动器 28 可在可换式磁盘 29 上读写，光盘驱动器 30 用于读 CD-ROM 盘 31 或在另一光媒体上读写。硬盘驱动器 27、可换磁盘式驱动器 28 和光盘驱动器 30 分别通过硬盘驱动器接口 32、可换磁盘式驱动器接口 33 和光盘驱动器接口 34 连接到系统总线 33 上。驱动器和其相关的计算机可读媒体向个人计算机 20 提供非易失性存储器。本领域的技术人员会理解单硬盘驱动器 27 包含一个或多个“分区”或“逻辑盘”，它们是物理驱动器的逻辑子集。尽管上述的计算机可读的媒体指的是硬盘驱动器、可换式磁盘和 CD-ROM 盘，但是本领域的技术人员应该理解还可在实施例操作系统中采用其它类型的计算机可读媒体，如磁卡、快速内存卡、数字视频盘、伯努里盒式盘和类似的部件。

可将多个程序模块存储在驱动器上和 RAM 25 中，包含操作系统 35 (如微软公司的“WINDOWS 95”操作系统)、一组应用程序 36 (如“MICROSOFT OFFICE”程序组)、一个或多个其它应用程序 37 和数据 38。操作系统软件 and 应用程序一般从 CD-ROM 31 或软盘 29 安装在硬盘驱动器 27 上。多数情况下，安装过程由存储在 CD-ROM 或软盘上的安装程序以及操作系统或应用程序相关文件来控制。

用户可通过键盘 40 和指点器如鼠标 42 向个人计算机 20 输入命令或信息。其它输入装置 (未示) 可包含麦克风、控制杆、游戏垫、卫星盘、

扫描仪或类似装置。这些和其它输入装置通常通过串行端口接口 46 与处理器 21 连接，串行端口接口 46 与系统总线连接，但是也可通过其它接口连接，如游戏端口或通用串行总线（USB）。还可通过接口如视频适配器 48 将监视器 47 或其它类型显示器连接于系统总线 23。除监视器外，个人计算机一般包含其它外围输出装置（未示），如扬声器或打印机。

个人计算机 20 可在采用逻辑连接到一个或多个远程计算机如远程计算机 9 的网络环境中工作。远程计算机 49 可以是邮件服务器（包含一个或多个消息存储）、文件服务器（包含一个或多个文件存储）、路由器、对等器或其它公共网络节点，一般包含许多或所有上述与个人计算机 20 相关的部件，尽管图 1 中只给出了内存存储器 50。图 1 所示的逻辑连接包含局域网（LAN）51 和广域网（WAN）52。该网络环境广泛使用在办公室、企事业范围的计算机网络、内部网络和互联网中。

当使用在 LAN 网络环境中，个人计算机 20 通过网络接口 53 连接到 LAN 51 上。当使用在 WAN 网络环境中，个人计算机 20 一般包含调制解调器 54 或其它在 WAN 52 如互联网上建立通讯的装置。调制解调器 54 可以是内置的或外置的调制解调器，可通过串行端口接口 46 连接到系统总线 23 上。在网络环境中，显示的关于个人计算机 20 的程序模块或其组成部分可以存储在远程内存存储器上。应理解图示的网络连接只是举例性的，还可使用在计算机之间建立通信连接的其它装置。

普通磁盘驱动器格式化

在说明本发明的特征之前，说明数据在普通硬盘存储器中存储的方式是很有帮助的。

硬盘存储器允许计算机改变存储于其中的数据，因而可以在磁盘上读写数据。物理磁盘驱动器包含一个或多个分区或逻辑盘，作为物理驱动器的逻辑子集。数据通常以独立文件存储在逻辑盘上，对每个文件进行单独存取。每个文件存储在磁盘上的位置被识别和存储在数据结构中，使计算机快速存取逻辑盘上的任何文件。

当对使用的硬盘驱动器格式化时，硬盘上的存储区域被划分成同心圆称为磁道。磁盘上磁道的数量取决于磁盘的大小。磁盘上的每个磁道被

分成相同数量的扇区。一般地，扇区呈片排列，使得在磁盘外缘的扇区占用的线空间多于靠近磁盘内边缘的扇区。但是，每个扇区用于存储相同量的数据，一般为 512 字节。

每个磁道上的扇区组成簇。扇区组成簇是由操作系统完成的，因此不是物理划分。磁盘上的每个磁道包含相同数量的扇区。根据磁盘的大小，簇通常包含 1 至 64 个扇区，每个扇区一般存储 512 字节的数据。每个簇是独立存取的。

计算机的操作系统在逻辑盘上的一个或多个簇中存储每个文件。大文件需要几个簇以存储与文件相关的所有数据，很小的文件可以存储在单个簇中。由于簇不存储来自一个以上文件的数据，因此如果文件不包含足够填充簇的所有扇区的数据，则一个或多个簇的扇区会是空的。因此，如果存储在每个簇中的数据量相对于多数文件较小的话，则最有效使用磁盘空间。

硬盘驱动器一般包含层叠的几个硬铝磁盘，其上覆盖有磁性材料。每个盘的上表面和下表面可存储数据。通过磁盘表面的磁化区域而在每个磁盘上对数据编码。通过读/写头恢复数据或将数据添加到磁盘上。磁盘驱动器的每个磁头包含连接到旋转臂上的小电磁铁，该电磁铁非常接近磁盘表面。向每张盘的每个记录表面提供一个磁头。该臂绕轴旋转并在磁盘表面上的径向路径上来回移动。磁头驱动器使臂旋转以便控制磁头的移动。当在磁盘上写数据时，磁头位于磁盘上要写入数据的合适区域上。加在磁头的电流产生磁场，磁化磁头附近的小磁盘区域。小的磁化区域代表数字比特。同样地，当在磁盘驱动器上读数据时，磁头位于磁盘上的合适区域上，在磁头上感应出电流。磁头上感应出的电流然后解码成数字数据。

磁盘驱动器的盘与单个轴连接。该轴与电机相连，以一致的恒速旋转磁盘。尽管磁盘的扇区可占用不同量的空间，但是存储在每个扇区的数据量是相同的。这使磁盘以恒速旋转，从而取出相同量的数据，而与磁盘上扇区的所在位置无关。

当磁盘上的新簇被存取时，在磁头实际开始读或写数据之前，必须进

行两个机械操作。第一，磁头从当前位置径向移动到包含目标簇的磁道位置。“寻道时间”指臂支撑磁头以克服惯性和摩擦作用和进行移动所需时间。“寻道时间”还包含“固定时间”，指的是磁头在移动后固定在静止位置所需时间。第二，磁头必须等待包含所需数据的簇旋转到磁头下面。该时间指的是“旋转等待时间”。由于磁盘以恒速旋转，因此磁头必须等待需要簇经过磁头下面的最长时间是需要盘完成一个循环的时间。因此，主要由于到达磁盘正确区域的机械动作需要，使新簇的每次存取产生固有延迟。

操作系统必须保持存储在逻辑盘上的文件位置记录。有几种数据结构用于文件存取。逻辑盘上的每个扇区具有基于逻辑盘上扇区位置的唯一识别数字。同样的，每个簇具有唯一簇数字。识别数字是这样分配的，即对相临扇区和簇连续编数字。用于确定逻辑盘的哪部分用于文件数据存储的基本数据结构是文件分配表（FAT）。其它文件系统包含高性能文件系统（HPFS）和 NT 文件系统（NTFS）。尽管在 FAT 文件系统中说明了本发明，本发明还应用于其它类型的文件系统。

在 FAT 文件系统中，FAT 存储在靠近逻辑盘或分区的开始端，并包含逻辑盘上每个簇的项。簇通过其簇数连续列在 FAT 中。每个簇的 FAT 项包含文件下一部分的簇的数字。包含文件最后数据的簇的 FAT 项包含文件的末端（EOF）项。因此，存储在逻辑盘上的每个文件在 FAT 中，表示为一个或多个簇链。FAT 指示是否每个簇已分配给了一个文件，但是不直接告诉计算机的操作系统所给的簇属于哪个文件。

如上所述，如果每个簇很小（如 4KB），则可最有效地使用磁盘空间。但是，磁盘分区的 FAT 大小决定可以包含在逻辑盘上的簇的最大数字，从而决定簇的大小。例如，在一个 FAT 16 文件系统中，FAT 使用 16-bit 数字表示每个簇。这就决定了磁盘上的簇最大数字为 65,536（即，64K）。数字通过 FAT 的大小划分磁盘的大小，以确定每个簇的最小量，如表 I 所示：

<u>磁盘大小</u>	<u>最小簇大小（比特）</u>
达到 32MB	512
〉 32MB 至 62MB	1K
〉 64MB 至 128MB	2K
〉 128MB 至 256MB	4K
〉 256MB 至 512MB	8K
〉 512MB 至 1GB	16K
〉 1GB 至 2GB	32K

表 I

FAT 32 文件系统采用 28-bit 数字表示每个簇，并能保持 2^{28} 个簇的磁道，在非常大的逻辑盘上允许有 4K 字节或更小的簇。在“WINDOWS 95”操作系统以及后来的“WINDOWS”操作系统如“WINDOWS 98”操作系统的 OSR2（OEM 服务版本 2）版本中提供 FAT 32。

称为目录的数据结构包含存储在逻辑盘上的所有文件的列表和子目录（有一些作为正规文件与目录比特置位（set）存储）。根目录不同于子目录，根目录没有父目录，并且一般先于用于存储文件数据的扇区存储在接近逻辑盘的开始端。该目录具有包含簇数字的每个文件的项，其中存储文件的第一部分。通过存储每个文件的开始簇数字，目录使每个文件依赖于 FAT。

为在逻辑盘上存取文件，计算机的操作系统读取文件的目录项，以确定对应于存储文件数据的第一簇的 FAT 项。操作系统然后读取开始于文件第一 FAT 项的整串 FAT 项。利用第一簇的位置和来自 FAT 的簇串，操作系统可以确定属于文件的每个簇，并相应存取每个簇。

当向逻辑盘添加数据或从逻辑盘上删除数据时，磁盘会变成碎片。碎片指在逻辑盘上的非相邻簇存储单个文件的组成部分。当在逻辑盘上写文件时，操作系统在由 FAT 的自由簇列表识别的空白空间内存储数据。用来存储数据的空白空间可以包括几个非相邻的簇。

为改进发送时间而安装程序或对程序重新排序

图 2 至 7 显示了以改进程序发送时间的方式安装应用程序相关文件的

方法实施例。本领域的技术人员会理解本发明可以结合各种应用程序、实用程序和操作系统相关程序使用，包含由华盛顿雷德蒙德的微软公司公开的“MICROSOFT OFFICE”应用程序组。

与现有技术一样，本发明必须确定在程序发送时从逻辑盘读取的文件部分的顺序。但是，与现有技术在逻辑盘上安装好文件后确定文件和对文件重新排序不同，本发明是当文件最初安装逻辑盘上时，就以接近最优化顺序排列文件。一般说来，通过确定发送中文件各部分存取顺序、在安装盘上产生装入顺序表和以装入顺序表指示的顺序安装的发送相关文件部分来实现上述过程。另外，在将文件安装在逻辑盘上后，可以使用装入顺序表对文件重新排序。装入顺序表可以通过互联网或其它发布装置装在安装盘上。

图 2 和 3 是流程图，显示了产生安装盘和以在程序发送中装入的顺序在逻辑盘上安装文件的举例性方法。图 2 的流程图显示了产生包含应用程序相关文件、一个或多个装入顺序表和安装程序的安装盘的方法 200。图 3 的流程图显示了从安装盘向逻辑盘安装程序以及将其以在发送中读取的顺序存储的方法 300。

首先参照图 2，图中显示了产生安装盘的方法 200。方法 200 的第一步是产生应用程序相关文件（步骤 205）。在该举例性方法中，以本领域的技术人员熟悉的普通方式实现该步骤，该步骤包含产生与程序（如应用程序）相关的所有文件。这些文件可包含可执行文件、动态链接库、数据文件、优选文件和类似文件。

在产生这些文件后，方法 200 运行到步骤 210。此时，文件安装在了计算机上，并发送程序。发送过程由测试代码监视或“测试”，其决定在发送中存取文件的哪部分。测试代码产生装入顺序表，该装入顺序表指示读取了哪个文件部分以及其读取的顺序。

图 4 显示了装入顺序表的格式。表中的每个项识别存取的文件、文件中的偏置和读取的数据量。以字节测量偏置和长度，该偏置和长度与文件的开始端相关。偏置和长度限定了读取部分的开始和长度。该步骤形成程序发送中接触的所有文件的表、每个读取的文件的位置以及当文件

发送时从磁盘读取的数据量。

方法 200 从步骤 210 运行到步骤 220。本领域的技术人员会理解初始装入顺序表包含不是程序本身的组成部分的文件，该初始装入顺序表在步骤 210 产生，并包含在引导中接触的所有文件的记录。例如，当发送应用程序时，可以要求计算机读各系统文件或其它共享文件的部分，这些文件的产生与应用程序本身无关。在步骤 220，方法 200 分析初始文件顺序表的内容并移去不是最优化的或在安装程序时重新安装在用户计算机上的文件。该排除的文件将包含不能或不应该由安装程序改变的文件以及可能很频繁改变的文件（如优选文件）。本领域的技术人员会理解可以以不同方式实现该步骤，包含使用防止其包含在初始装入顺序表中的排除表，或在安装时处理装入顺序表以忽略某些文件。

在编辑装入顺序表以移去不是最优化的文件后，方法 200 运行到步骤 225，产生一个或多个最后装入顺序表。本领域的技术人员会理解装入文件的顺序可以受各种与将安装程序的计算机相关的因素影响。例如，操作系统的特殊版本（包含主版本和小的更新）和与计算机相关的簇大小与在逻辑盘上安装文件部分的最佳顺序有一些关系。因此，每个方案可以产生不同装入顺序表。例如，与操作系统的特殊版本一起使用的装入顺序表可为另一机器配置添加附加文件或从优化装入顺序表删除文件。同样地，可以提供用于各个簇大小的装入顺序表。另外，可以提供和小簇（如 4KB）一起使用的单个装入顺序表，在文件排序时由其获得其它簇大小的装入顺序表。在图 4 所示的实施例中，最后装入顺序表具有相同的格式。

在步骤 230，产生实际的安装盘（如 CD-ROM 或软盘）。安装盘包含应用程序相关文件、装入顺序表和安装程序。当安装盘插入用户计算机上的驱动器时，用户会发送安装程序以在计算机逻辑盘上安装文件。根据用户的输入和从计算机读取的配置信息，安装程序确定需要安装哪个文件以及文件应该安装的顺序，并从安装盘上读文件和以正确顺序写磁盘。

图 3 显示了方法 300，方法 300 用于从安装盘向逻辑盘安装文件，并

以在程序发送中读取的顺序存储发送相关程序部分。在一实施例中，由安装盘上包含的安装程序执行方法 300，安装盘可以是插入盘驱动器 30 的 CD-ROM（图 1）。

当用户发送安装程序时，开始方法 300 并运行到步骤 305。在步骤 305，安装程序询问计算机并获得适于操作系统安装在计算机上的数据、硬盘驱动器的特性等。

在步骤 310，安装程序提示用户并获得适于特殊程序或用户想要安装的程序的信息。进行安装前，安装程序还要求用户输入他或她的名字和系列编号。

在步骤 315，安装程序使用在步骤 310 和 315 获得的信息以确定来自安装盘的哪个文件应该安装在硬盘驱动器上。

在步骤 320，安装程序采用在步骤 310 和 315 获得的信息以确定应该使用哪个装入顺序表来确定应该安装和存储在逻辑盘上的发送相关文件部分的顺序。如前所述，可以提供不同的装入顺序表以适于不同版本的操作系统、不同硬盘驱动器簇大小等。

图 5 显示四个应用程序文件的部分（文件 A, B, C, D）和简单装入顺序表的实例，该装入顺序表识别发送相关文件部分和当程序发送时其装入的顺序。在该实例中，应用程序可以包含文件 A、B、C 和 D。当然，应用程序可以包含许多附加文件。每个文件包含多个部分（如 A1-A8、B1-B6、C1-C5 和 D1-D5）。但是，只有文件 A、B 和 C 包含在应用程序发送时装入的部分。这些发送相关部分包含部分 A1-A4、B1-B3、C1-C3。文件 A、B、C 剩下的部分和整个文件 D 不在发送中装入。本领域的技术人员会理解每个文件（如 A1-A4）的发送相关部分不需要是邻近的或优先于非发送相关文件部分（如 A5-A8）。图 3 的最后线显示了发送装入顺序，其说明了在发送中发送相关文件 A、B 和 C 部分装入的顺序。在该实施例中，发送装入顺序是 A1、B3、A3、B2、B1、C2、C1、A2、C3、A4。

剩下的安装过程（步骤 325、330 和 335）目的是以上述发送装入顺序在逻辑盘上的邻近簇中安装的发送相关文件部分。这样，安装过程可描述成具有两个目标：（1）以正确顺序写入发送相关文件部分以及（2）

保证发送相关文件部分存储在逻辑盘上的邻近扇区中。通过装入顺序表对以正确顺序写发送相关文件部分的目标编址。满足存储在邻近簇中的文件部分由于用于大多数个人计算机上 FAT 文件系统的各特性的要求。

一般，FAT 文件系统以“簇自由表”顺序将文件写入逻辑盘。换句话说，当文件写到硬盘驱动器上时，文件的第一簇写在 FAT 中识别的第一自由簇，文件的第二簇写在 FAT 中识别的第二自由簇，等等。在许多情况下，自由簇不是邻近的。因此，当安装程序时，需要识别包含足够邻近自由簇的逻辑盘部分，以便存储程序发送相关文件部分。可以以簇自由表顺序写剩下的文件部分。

在步骤 325 中，安装程序识别将用于存储发送相关文件部分的邻近自由簇块（在逻辑盘上）。本发明考虑用几种方法来识别足以存储发送部分文件的可用磁盘空间。

第一种方法是假定产生的文件存储在逻辑盘的开始端，以及假定在末端有足够邻近自由空间量。尽管该假定是不现实的，但是在安装过程中，可以建议用户使用普通碎片整理实用程序来整理逻辑磁盘的碎片，以实现该假定。在整理了磁盘碎片后，应用程序可以安装在磁盘末端的自由空间中，使发送相关文件部分存储在邻近簇中。本领域的技术人员会理解该方法是不够理想的，因为在安装前对逻辑盘整理碎片要花费时间，并且一些用户会不愿意或不能完成建议的碎片整理。

第二种方法是假定在磁盘的末端有大自由空间，但是在磁盘开始端有一些非邻近自由簇。在该方法中，最后安装来自装入顺序表的发送相关文件部分。这会使剩下的应用程序文件（即非发送相关部分）首先安装并以簇自由表顺序保存在磁盘上。在多数情况下，存储的大量文件数据应该导致非发送相关文件部分存入非邻近自由簇和发送相关文件部分最后安装在邻近自由簇中，该邻近自由簇保留在磁盘的末端。

第三种识别足够邻近自由簇的方法包含读整个自由簇表并追踪任何邻近自由簇组的大小。然后以大小排序，以便发现的最大邻近自由簇组。识别和标记用于存储发送相关文件部分的最大自由簇组。在识别了发送文件的目标簇后，过程利用自由簇的初始表，并开始向逻辑盘写入数据。

如果目标簇不是表中的第一自由簇，为了取得理想的目标簇而将临时文件写入需要存储的自由簇。在发送文件写入邻近目标簇后，删除临时文件，这些簇返回簇自由表。下面更详细地说明临时文件的使用。

第四种方法类似于第三种，但是减少了需要写成临时文件的数据数量。可能第三种方法会需要许多要写到磁盘早期部分的多个小的簇组数据。为了使活动最少化，第四种方法采用最早期邻近簇组或组集合来大大改进程序发送时间。

在已经识别了用于安装发送相关文件部分的足够邻近自由簇后，安装程序运行到步骤 330，并以正确顺序将发送相关文件部分写入自由空间。本领域的技术人员会理解，需要以装入顺序表指示的顺序将发送文件部分写入自由空间的特殊步骤会依赖于各种因素。这些因素中最重要的可能是计算机特殊操作系统，该特殊系统提供工具，用以将文件写入逻辑磁盘，并限制多少文件可以写入硬驱动器。下面说明适于与“WINDOWS 95”操作系统一起使用的两种方法。

本领域的技术人员会理解在逻辑盘上写发送文件的最简单方案是当自由空间是邻近的并位于逻辑盘的末端以及操作系统不试图强迫文件成邻近的时候。在该简单和也许不现实的方案中，可以以下述方式实现在逻辑盘上写文件的过程。第一，安装程序打开 CD-ROM（用于读）和逻辑盘（用于写）上的必要文件。接着，从 CD-ROM 上读在装入顺序表中列出的第一簇并将其写入硬盘驱动器。有些情况下，必须刷新（flush）输出文件，以保证数据已实际写在磁盘上。重复这些步骤，直到装入顺序表中的所有簇都写入逻辑盘了。此时，安装程序可以运行到步骤 335，在步骤 335，从 CD-ROM 上读剩下的文件部分（即非发送相关部分）并将其写入逻辑盘。当写入所有文件时，读、写的文件关闭，操作系统更新 FAT 文件系统。方法 300 接着在步骤 340 结束并完成安装。

在“WINDOWS 95”操作系统的上下文中说明了将单独文件写入非邻近簇（步骤 330）。“WINDOWS 95”操作系统试图将文件保存在邻近簇中。因此，当多个簇中写入单个文件时，操作系统试图将其放入邻近簇中。在本发明的一个实施例中，通过用临时文件占用硬盘驱动器上的空间来

克服操作系统的这一特征，并迫使每个文件的发送相关部分以期望的顺序写入非邻近簇。本领域的技术人员会理解操作系统和文件系统文件可写入顺序强制的要求。最重要的是，文件的第一簇是以单调递增顺序分配的。换句话说，在文件的第二簇前分配文件的第一簇，等等，而与写入文件的信息的顺序无关。

图 6 显示了临时文件的使用的实例，反映在安装过程的不同阶段中在逻辑盘上的 12 个簇的内容。在该实施例中，图 5 的 10 个文件部分将写入图 6 所示的 12 个簇中。在安装开始前，12 个簇（任意标为数字 0—11）是自由簇表中的邻近簇。如果邻近簇 0—11 不是自由簇表中的第一簇，安装程序将临时文件写入簇 0 前的簇。这使簇 0 成为自由簇表中的第一自由簇。

安装程序会从合适的装入顺序表中读取期望的簇顺序、从 CD-ROM 中读簇并将其写入逻辑磁盘的簇 0—9。由于操作系统要求以单调递增顺序写每个文件的簇，将文件写入逻辑磁盘的过程会需要多次通过自由空间，并包含使用临时文件以保证数据写入期望的簇。

现在参照图 6，在通过 1 中，安装程序读装入顺序表并确定簇 A1 应该首先装入。因为这是文件 A1 中的第一簇，因此可将其立即写入簇 0。

接着，安装程序确定簇 B3 应该写入簇 1，但是，还不能保存簇 B3，因为簇 B1 和 B2 还没有写入硬盘驱动器。因此，安装程序将临时文件 t1 写入簇 1。

在写完临时文件 t1 后，安装程序确定应该写簇 A3。但是不能保存簇 A3，因为簇 A2 还没有写入硬盘驱动器。因此，安装程序将临时文件 t2 写入簇 2。

然后将临时文件 t3 写入簇 3，因为直到簇 B1 已经写入硬盘驱动器后才能保存簇 B2。

在写入临时文件 t3 后，安装程序确定应该将簇 B1 写入簇 4。因为这是文件 B 中的第一簇。因此可以将其写入硬盘驱动器的簇 4 中。

在簇 B1 后，安装程序将临时文件 t4 入簇 5，因为直到簇 B1 已经写入硬盘驱动器后才能保存簇 B2。

接着，安装程序确定应该将簇 C1 写入簇 6。因为簇 C1 是文件 C 中的第一簇，因此将其写入簇 6。

接着，安装程序确定应该将簇 A2 写入簇 7。因为已经写了簇 A1，因此簇 A1 可以写入簇 7。

在将簇 A2 写入硬盘驱动器后，安装程序确定簇 C3 应该写入簇 8。但是，由于没有存储簇 C2，因此还不能写簇 C3。安装程序将临时文件 t5 写入簇 8。

接着，安装程序确定应该将簇 A4 写入簇 9。但是，由于簇 A3 还没有写入硬盘驱动器，因此安装程序将临时文件 t6 写入簇 9。

在安装程序已经将数据写入所有将用于存储发送相关文件部分的簇后，安装程序将大临时文件 t7 写入磁盘上剩下的自由空间。

在通过 2 中，安装程序继续以单调递增的顺序将发送相关文件写入硬盘驱动器的过程。

在通过 2 中，簇 B3 不能写入硬盘驱动器，因为簇 B2 还没有被保存。

由于簇 A1 和 A2 已经保存，则簇 A3 可以写入簇 2。因此安装程序删除临时文件 t2 并将 A3 写入簇 2。

由于簇 B1 已经保存，则簇 B2 可以写入簇 3。安装程序删除临时文件 t3 并将 B2 写入簇 3。

由于簇 C1 已经保存，则簇 C2 可以写入簇 5。安装程序删除临时文件 t5 并将簇 C2 写入簇 5。

由于簇 C1 已经保存，则簇 C3 可以写入簇 8。安装程序删除临时文件 t5 并将 C3 写入簇 8。

最后，由于簇 A1-A3 已经保存，则簇 A4 可以写入簇 9。安装程序删除临时文件 t6 并将 A4 写入簇 9。

在通过 3 中，由于簇 B1 和 B2 已经保存，因此簇 B3 可以最后写入簇 1。安装程序删除临时文件 t1 并将簇 B3 写入簇 1。

在写完所有发送相关文件部分后，安装程序删除临时文件 t7 和任何写入簇 0 以前的簇的临时文件。这使得空着的簇返回自由簇表。此时，从 CD-ROM 读取剩下的程序部分（即非发送相关文件部分）并将其写入

硬盘驱动器。这些文件将以簇自由表顺序保存在硬盘驱动器中。

有些版本的“WINDOWS 95”操作系统采用文件分配策略，假定如果打开两个文件并且程序正写入两个文件，程序会在文件之间需要一些自由空间。图 7 显示了第二种方法，用于将发送文件部分写入硬盘簇 0—9，尽管这是操作系统的特征。第二种方法非常类似于图 6 所示的方法，但是需要额外通过自由簇以在每个自由簇中产生临时文件。

在图 7 的实例中，图 5 的 10 个文件部分将写入硬驱动器的簇 0—9。在安装开始前，12 个簇（数字 0—11）是在自由簇表中的邻近簇。如果邻近簇 0—11 不是自由簇表中的第一簇，安装程序将临时文件写入簇 0 之前的簇。这使簇 0 成为自由簇表中的第一自由簇。

在通过 1 中，安装程序将独立临时文件（t1-t10）写入每个簇，每个簇将用于存储在装入顺序表中识别的文件簇。将另一临时文件 t11 写入跟着簇 9 的所有自由簇。这保证操作系统不会在由安装程序写的簇之间留出空间。

通过 2—4 与图 6 的通过 1—3 同样进行，并用于以单调递增顺序写文件簇。但是，在通过 2 中，由于簇 A1、B1、C1 和 A1 可以分别写入簇 1、4、6 和 7 中，因此必须删除临时文件 t1、t5、t7 和 t8。

在所有发送相关文件部分已经写入逻辑盘后，安装程序删除临时文件 t11 和任何写入簇 0 之前的簇的临时文件。这使空着的簇返回自由簇表。此时，从 CD-ROM 中读剩下的文件部分（即非发送相关文件部分）并将其写入逻辑盘。这些文件将以簇自由表顺序保存在逻辑盘上。

上述说明描述了本发明的几个实施例。这些实施例一般包含一种识别哪个发送相关程序部分应该安装在逻辑盘上的邻近簇的装入顺序表，该装入顺序表用于识别足够量的邻近自由簇的方法，以及用于以期望的顺序将程序部分写入邻近簇。

本发明具有几个附加特征，可以合并入本发明的实施例。

可以将本发明的方法扩展，不仅包含新文件的发送相关部分，而且包含产生逻辑盘上的文件和新程序的附加部分。

例如，如果系统文件和其它预先安装的文件包含在装入顺序表中，则

本发明可用于优化这些文件。为了完成这些优化，进行如上所述的安装过程，但还会包含打开要读的文件和打开要写的新临时文件。将产生文件的相关簇和其它发送相关程序部分一起写入邻近自由簇。在发送相关部分都写入逻辑盘上的邻近簇后，将删除初始产生文件，对临时文件重新命名成具有与删除的初始文件相同的名字。

如上所述，本发明的原理可用于测试（instrument）和安装（或重新排序）其它命令、程序的磁盘加强部分。例如，可以测试执行例行程序如 OPEN、SAVE、LOAD 和 PRINT 的代码，结果包含在安装盘上的装入顺序文件中。然后，当安装新程序时，安装程序使用该信息并将这些部分代码写入逻辑盘的邻近部分，或在安装后对代码的相关部分重新排序。

如上所述，在安装过程的有些阶段，临时文件可以用于占用逻辑盘上的几乎所有自由空间。在完成安装过程后，一般由安装程序删除这些临时文件。但是，为了保证移去临时文件并向计算机提供空着的空间，可以对由操作系统执行的后-再引导（post-reboot）或“调整”过程编程，以便检查出现的一个或多个临时文件并删除发现的文件。安装过程的后-再引导部分可以用于删除和重命名任何由于使用在安装过程中而不能删除的优化系统文件。

图 8 是流程图，显示采用本发明原理优化涉及启动的代码部分和在该代码已经安装在逻辑驱动器上后的其它磁盘加强操作的另一种方法 800。与图 3 的方法相同，方法 800 采用一个或多个装入顺序表和程序文件的安装盘。但是，在该实施例，程序文件是以普通方式与复制的装入顺序表一起首先安装在逻辑驱动器。在安装后，有时调整程序可以使用装入顺序表以优化形式对磁盘加强数据重新排序。

在步骤 805，方法 800 包含以普通方式在逻辑驱动器上安装程序文件。此时，复制的装入顺序表还复制到逻辑驱动器上。

在安装好程序后，可以使用单独程序对与磁盘加强操作如程序启动相关的程序部分重新排序。可在各个时间运行该“调整”或重新排序程序。例如，当安装好程序后计算机重新启动时，可以自动运行调整程序。每逢用户选择完成他或她的计算机的维护活动时，还可人工控制调整程序。

步骤 810 是调整或重新排序过程的第一步。在步骤 810, 调整程序查询计算机并获得匹配安装在计算机上的操作系统以及硬盘驱动器特性的数据。

在步骤 815, 调整程序使用在步骤 810 获得的数据以确定应该使用哪个装入顺序表来确定应该在逻辑盘上重新排序的磁盘加强文件部分的顺序。如前所述, 可以提供不同装入顺序表以适应不同版本的操作系统、不同硬盘驱动器簇大小等等。

本领域的技术人员会理解, 程序组如“MICROSOFT OFFICE”程序组包含一个以上应用程序。通过提供每个应用程序的装入顺序表可以将本发明应用于一组程序的安装。如果要支持多个版本的操作系统, 本发明可以提供每个支持系统的每个应用程序的装入顺序表。当“优化”程序时(在安装中或安装后), 可应用的装入顺序表可以以预定顺序集中在一起。如果发送相关文件部分包含在一个以上的装入顺序表中, 可为了包括该部分可以从除第一装入顺序表的所有装入顺序表中移去该部分。

在步骤 820, 调整程序识别用于存储磁盘加强文件部分的邻近自由簇(在逻辑盘上)块。如上所述, 本发明采用几种方法用于识别足够存储发送文件部分的可用磁盘空间。

在为安装发送相关文件部分而识别了足够邻近自由簇后, 安装程序运行到步骤 825, 并以正确顺序在自由空间中对磁盘加强文件部分重新排序。

简单地说, 图 8 的方法使文件在安装时或紧接着重新排序, 还在安装后的任何时候对其调整或固定。通过采用装入顺序表, 完成该调整而不使用现有技术中使用的监视功能。

最后, 上述的过程使用由操作系统提供的读、写功能。该方法具有很强功能并实际减小了安装过程毁坏用户硬盘驱动器上的数据的可能性。但是, 本领域的技术人员会理解, 通过使用低级功能在簇级写数据, 可以将数据写在硬盘驱动器上。该方法要求安装程序将文件部分写入正确簇, 然后返回和修补(patch) FAT。尽管在本发明的上下文中可以采用该方法, 但是有使硬盘上的数据以一些方式遭受破坏的危险。

详细描述的综述

从上述说明中会理解本发明提供一种以改进程序发送时间的方式安装文件或文件重新排序的改进的系统和方法。本发明的主要优点在于，在通过提供一个或多个装入顺序表安装好程序后，不需要监视启动过程。

上述的实施例监视程序启动过程，并确定在磁盘加强操作如发送程序或选择某些程序命令中从磁盘读取文件部分的顺序。该数据用于产生装入顺序表，该装入顺序表指示在发送中读取文件各个部分的顺序。在安装中，安装程序从装入顺序表读数据，并以上述顺序将发送相关文件部分写在逻辑盘上的邻近簇中。另外，在程序安装后，根据装入顺序表对发送相关文件部分重新排序。在完成上述过程后，计算机从逻辑盘上以正确顺序从邻近磁盘簇读发送相关数据，这减少和消除了由于为读取发送数据而使磁盘磁头必须在非邻近簇之间移动的磁盘存取所造成的时间浪费。

本发明还可以在一个或多个根据和实施图2至8所示的特征的程序模块中实现。没有说明为实现上述各个过程的特殊编程语言，因为考虑到上述的和附图所示的操作、步骤和过程足以公开了供本领域技术人员实施本发明的内容。此外，由多种可以实施本发明的计算机和操作系统，因此不能提供适于这些不同系统的详细计算机程序。特定计算机的每个用户会知道最满足用户需要和目的的语言和工具。

此外，尽管在“WINDOWS 95”操作系统下安装应用程序的安装程序的上下文中主要说明了本发明，本领域的技术人员会理解本发明适用于各种类型的程序模块和运行各种操作系统并使用各种文件系统的系统。例如，本发明可用于安装应用程序、实用程序、操作系统程序模块和类似程序。如上所述，本发明还可用于对与多种磁盘加强操作相关的代码相关的程序数据排序，该磁盘加强操作包括发送应用程序、发送与程序命令（如 FILE OPEN）和类似的操作。

上面说明了关于特殊实施例的本发明，这些实施例只是用于说明本发明而不是限定本发明的。本领域的技术人员会明白有适合于本发明而不背离本发明的精神和范围的可替换实施例。因此，本发明的范围不是由上述说明限定的，而是由附加权利要求限定的。

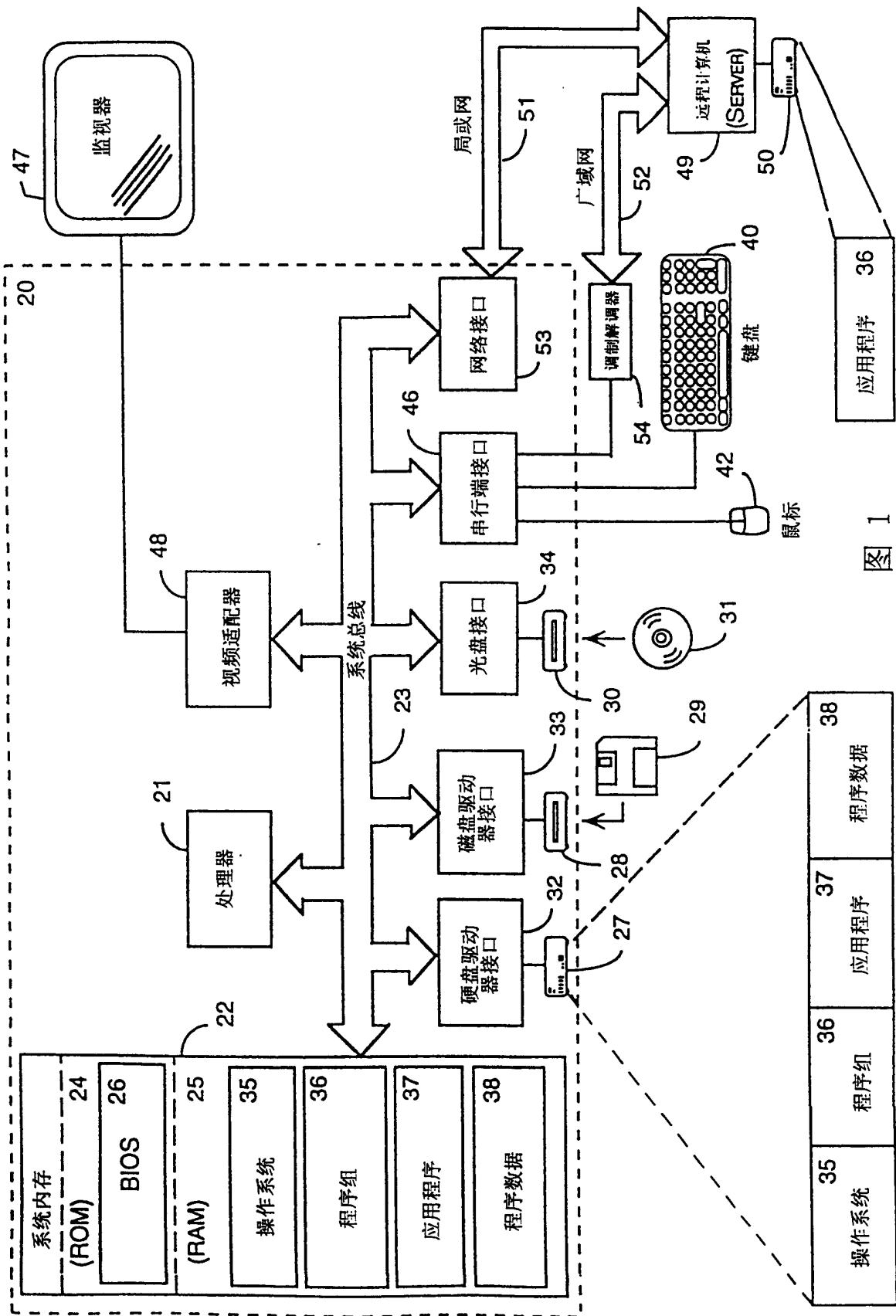


图 1

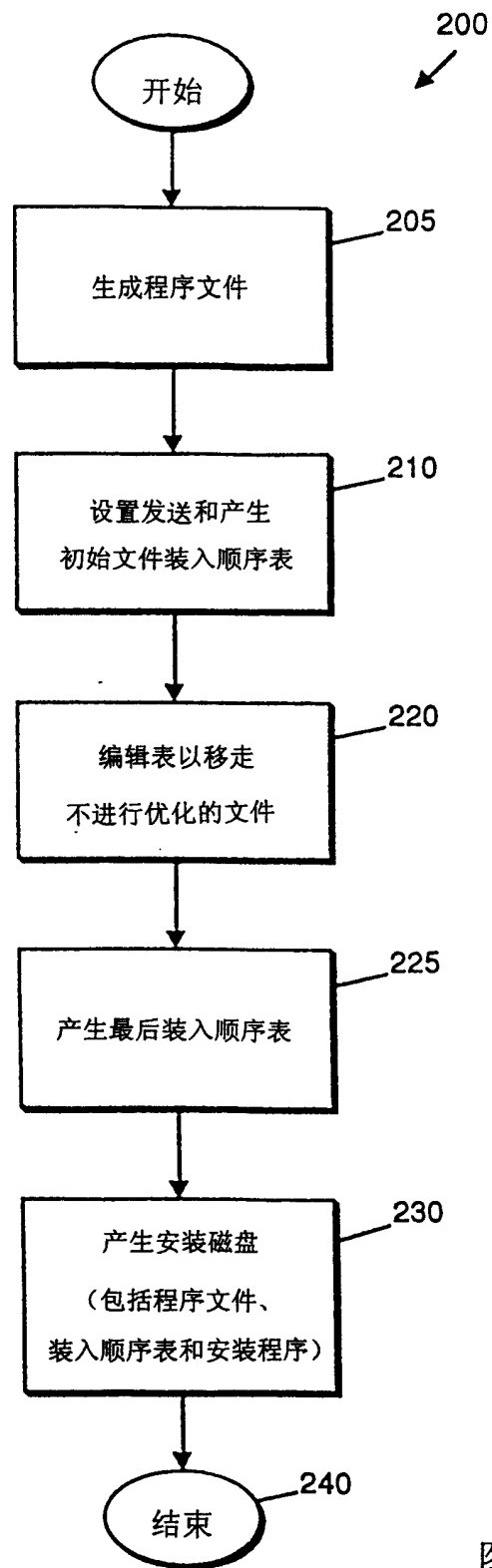


图 2

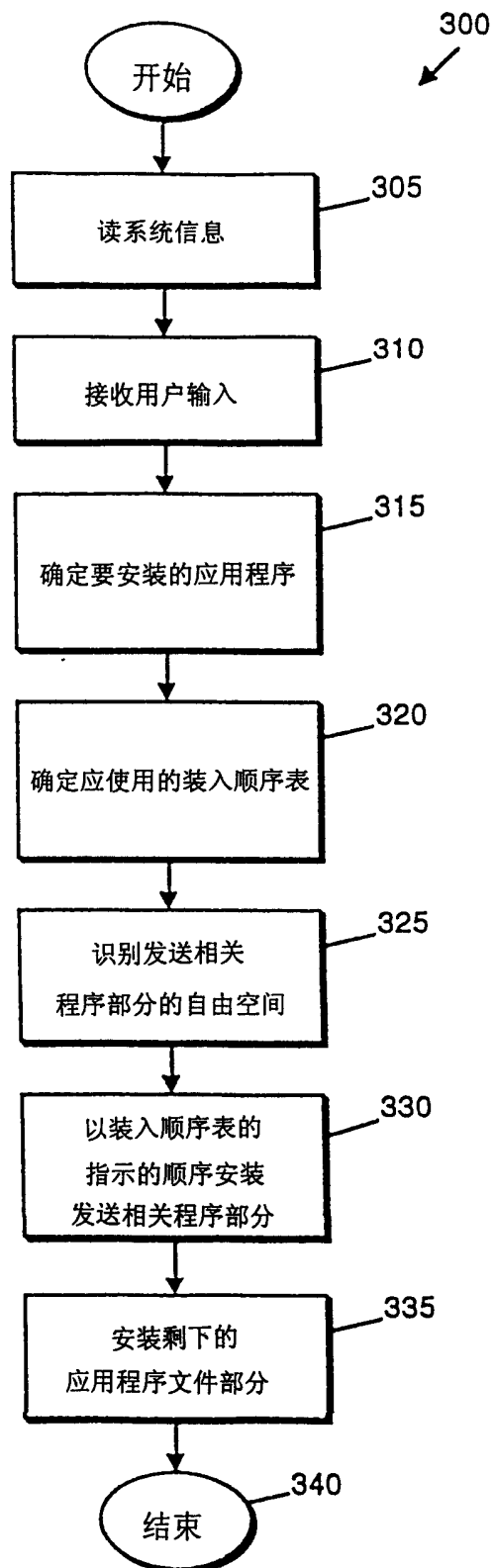


图 3

装入顺序表

文件ID	偏置	长度
c14d8220	80	1c0
023c0000	a00	e0
...	...	...

图 4

发送相关部分				非发送相关部分				
文件A部分:	A1	A2	A3	A4	A5	A6	A7	A8
文件B部分:	B1	B2	B3		B4	B5	B6	
文件C部分:	C1	C2	C3		C4	C5		
文件D部分:					D1	D2	D3	D4 D5
发射顺序: A1 B3 A3 B2 B1 C2 C1 A2 C3 A4								

图 5

簇											
	0	1	2	3	4	5	6	7	8	9	10 11
安装前	[----- Free -----]										
通过1	A1	[t1]	[t2]	[t3]	B1	[t4]	C1	A2	[t5]	[t6]	[--- t7 ---]
通过2	A1	[t1]	A3	B2	B1	C2	C1	A2	C3	A4	[--- t7 ---]
通过3	A1	B3	A3	B2	B1	C2	C1	A2	C3	A4	[- Free -]

图 6

簇											
	0	1	2	3	4	5	6	7	8	9	10 11
安装前	[----- Free -----]										
通过1	[t1]	[t2]	[t3]	[t4]	[t5]	[t6]	[t7]	[t8]	[t9]	[t10]	[-- t11 --]
通过2	A1	[t2]	[t3]	[t4]	B1	[t6]	C1	A2	[t9]	[t10]	[-- t11 --]
通过3	A1	[t2]	A3	B2	B1	C2	C1	A2	C3	A4	[-- t11 --]
通过4	A1	B3	A3	B2	B1	C2	C1	A2	C3	A4	[- Free -]

图 7

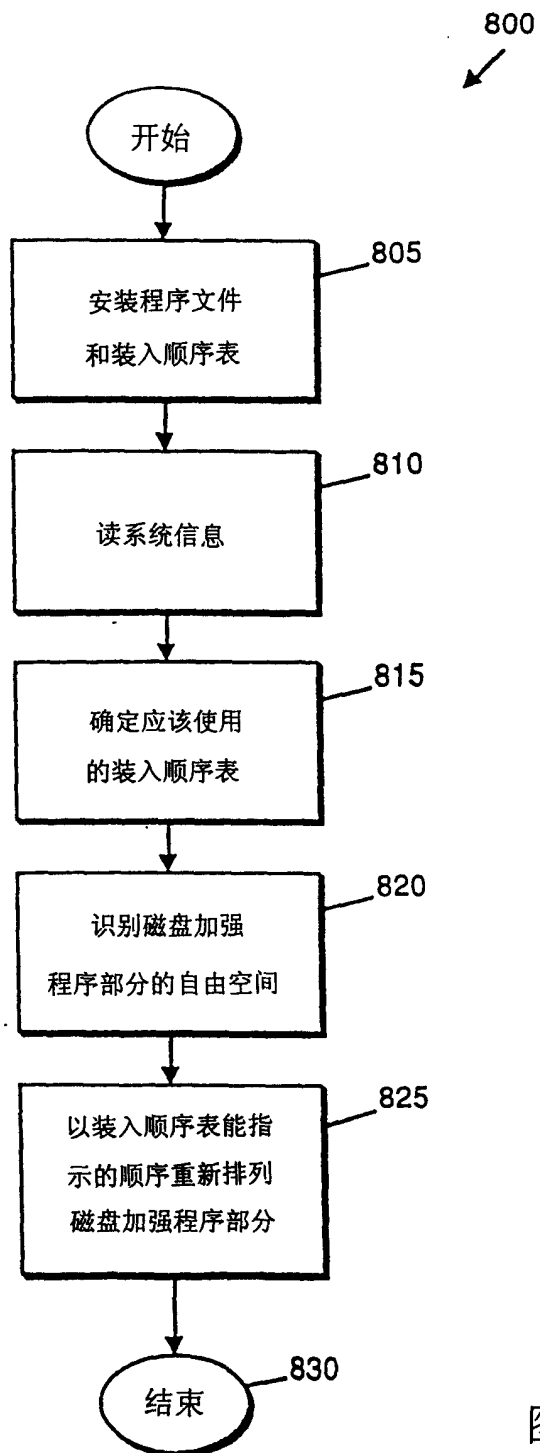


图 8