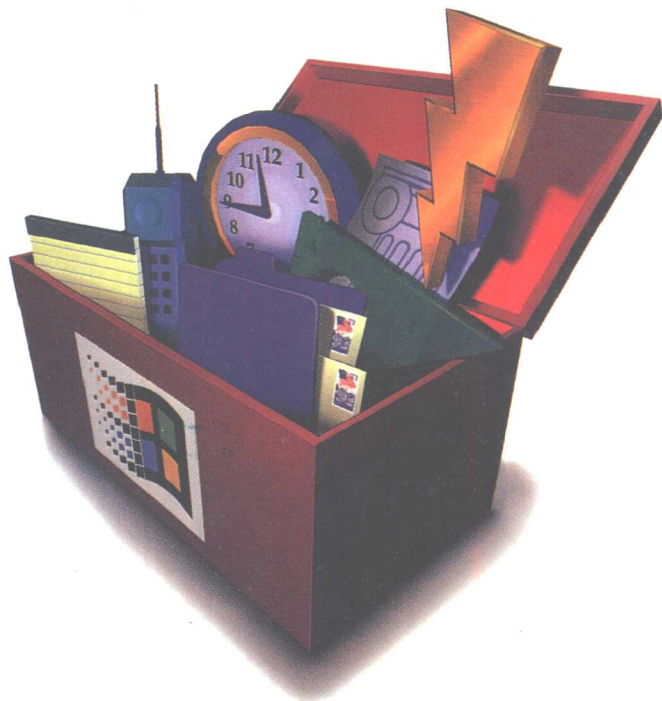


# Windows NT™

## 优化

[美] Microsoft公司 Russ Blake 著  
国家智能计算机研究开发中心 译  
北京曙光计算机公司



# Microsoft® WINDOWS NT



## 资源手册 (第三卷)

北京大学出版社

# Windows NT 资源手册

(第三卷)

## Windows NT 优化

[美] Microsoft 公司 Buss Blake 著

国家智能计算机研究开发中心  
北京曙光计算机公司 译

北京大学出版社

## 内 容 简 介

Windows NT 即 Windows 新技术(New Technology),是美国 Microsoft 公司的新产品。它在安全性、支持多操作系统并存、支持网络环境以及支持多媒体等方面具有明显的技术优势和独到的技术特色。《Windows NT 资源手册》一书是一套内容极为丰富的,介绍 Windows NT 庞大资源宝库的技术资料。该书共分三卷。第一卷全面介绍了 Windows NT 各类资源的基本概念、功能、原理及使用的要点,还介绍了 Windows NT 与现有其他操作系统的关系以及如何在 Windows NT 环境下使用现有其他操作系统的问题;第二卷对因系统配置不当、用户及代表用户运行的应用程序的操作不当而产生的出错消息以及报告系统运行状态的各种消息作了有指导意义的解释;第三卷进一步对可以实时动态监视系统性能的各种监测手段(特别是各种性能监视器、计算器)作了更加深入的介绍和剖析,对在各种复杂情况下优化使用 Windows NT 系统提出了具有重要参考价值的建议措施。

这套书不同于一般的用户手册或程序员手册。它不仅让读者知其然,还让读者知其所以然,从原理上让读者有所收益。这样,这套书就不仅仅是开出一些头疼医头、脚疼医脚的药方,而是力图给读者以智慧上的启迪,让读者在不知不觉间成为使用自如的 Windows NT 的高手。对于 Windows NT 这样一座博大精深的技术宝库来说,有了这套书,就等于有了一幅脉络清晰的藏宝图,就可以居高临下、知难而进,凭借这套书提供的丰富的 Windows NT 资源,攻克一个又一个使用 Windows NT 时遇到的技术难关。

这套书适合于在大专院校、科研机构、高技术产业和各行各业学习与工作的广大 Windows NT 用户阅读。

---

---

本书版权为美国 Microsoft 公司所有,1985—1993。(Copyright ©1985—1993 by Microsoft Corporation. All rights reserved.)

本书中文简体版版权于 1994 年由 Microsoft 公司转让予中国国家智能计算机研究开发中心,由北京大学出版社独家出版。· 版权所有,翻制必究。

---

---

## 作者简介

作者 Russ Blake 毕业于 Antioch 大学(BA, 哲学), 并在 Madison 的 Winsconsin 大学取得了计算机科学硕士学位。自从 70 年代早期他在 HP 公司参加了 HP3000 开发小组以来, 就被卷入了操作系统和性能监视工具的设计之中。他为 Tandem 计算机发明和创造了 Xray 性能监视器。在机器人学及其实时性能方面进行了一些研究工作之后, 他以软件产品设计工作主任的身份加入了 Sun Microsystems。最近五年来, 他又成为主管 Microsoft 公司高级操作系统性能的经理。他参与并领导了 Windows NT 操作性能监视器的设计, 并开发了其他一些用于调整 Windows NT 操作系统及 Windows NT 应用程序主要性能的工具。

# 绪 言

25年前我第一次遇到计算机性能问题。我的一位大学教授想在后院为他的孩子做一个跳绳。他要了解绳子的强度应是多少,问能否提供帮助。由于渴望炫耀自己的编程专长,于是很乐意地接受了下来。教授说明了孩子的体重和绳子需要的长度后,我便急忙奔向分时终端。写好程序,在纸带上穿好孔,不久答案就交给了教授,我对自己的成功感到非常愉快。

几周之后,另一位管理分时计算机的教授在顺便来访时,问起是什么原因致使本月的计算机维护费用超过了600美元。我知道我们要承担联机时间和计算时间的费用,但这个账单似乎令人难以置信。是什么原因导致了这种结果呢?几天之后我醒悟了。这是由于我那计算跳绳张力的小程序!它是从0到X不断重复的,这里X是跳绳的倾角。每当我使X增加,程序都回到0重新开始。对程序做了这样一个非常简单的改动后,其效率就百倍地提高了。600美元,哎。

所以说我的第一个性能测量工具是一个账单。有时账单还是最有效的工具,因为差的计算机性能要花费许多钱。并且解决性能问题时买错了硬件还要浪费许多钱。但性能差的最大代价是它降低了使用这些系统和程序的所有用户的工作效率。如果我们能适当调整自己国家所有的程序和计算机,那么国债就能还清了。

我希望能帮助大家避免600美元的账单,很可能今天的数字要比600美元大得多。本书描述了Microsoft® Windows NT™操作系统及其应用程序的动态特性,以及这种特性如何影响Windows NT及其应用程序的性能。本书对计算机领域的很多方面都非常有用,包括:

- 企业信息系统和网络的管理者,他们在服务器和桌面计算机上评估、设计、配置和管理着Windows NT
- 企业部门管理者,他们必须维护Windows NT的部门服务器
- 企业信息系统的人事部门,他们设计、实现、维护着配置在Windows NT上的重要的应用
- 独立软件卖主,他们要把16位应用移植到Windows NT上,或者为Windows NT开发32位应用程序
- Windows NT用户,他们关心着能提高计算机性能的附加硬件

考虑到读者背景不同而且所拥有的系统也不同,所以我们设法仔细地定义我们所沿用的术语。如果我们赢得了这场游戏,那么就需要一个标准的游戏场地。希望它在精通计算机科学的人们面前不至于显得过于笨拙。但无论如何,读者应该理解:像所有的系统一样,Windows NT 有它自己的术语。

读完本书,将能够:

- 决定配置 Windows NT 的硬件需求
- 查找网络、服务器和台式计算机的瓶颈
- 准确地确定购买哪些硬件能最有效地提高生产率
- 获得各种活动如何影响计算机硬件性能的知识
- 完成性能规划,以决定服务器和桌面计算机将来的设备需求
- 了解 Windows NT 重要性能的特征
- 确定各种设计折衷对最优地执行程序产生的影响
- 估计应用程序的硬件需求
- 确定应用中的瓶颈,如果可能的话,再取消它们

本书的专题性很强,既有价值,又有一定的责任。它针对的是 Windows NT 3.1 的初始版,并为 Windows NT 的性能管理提供了非常重要的细节。但这些细节也可能在后续版本中有所变化,例如在 Windows NT 的最近版本中,有些术语已被重新定义,有些计数器的名字也做了改变,而且我们这里提到的一些技巧已不再适用。我们尽力提供尽可能多的工具,以便在新版本中出现某些变化时,还能够清晰地描述它们。

改变 Windows NT 并不是调整众多的系统参数的问题。取而代之的是利用一组新的强大的工具来分析系统性能。每一个工具都将一步步引导用户直到杀死瓶颈恶魔。

随书提供的软盘(或 CD-ROM)中,有一个人人为负载生成程序以及一些其他有用的工具。人为负载生成程序用于进行系统中被控制的性能实验,在本书附录 C 中做了描述。其他实用程序都有联机文档。在本书中,我们是在涉及到相关主题时,讨论这些工具的,但用户应该浏览一下软盘中提供的这些工具。它们不沉,所以不如来回搬动搬动。

有些专题在这本书中没做描述。例如,最小化 Windows NT 所需要的磁盘空间。像曾经提到过的一样,我们没有试图详尽论述改变附录中列出的参数所带来的影响,这主要是因为它们从来不需要改变,除非情况非常反常。同时我们没有详细比较不同硬件平台上的 Windows NT 性能。但我们能够让用户在自己的环境中做出这种性能比较。

本书是集体智慧的结晶。它建立在强大的软件基础之上。这个基础的根基是以下几个卓有贡献的小组：设计小组（由 Dave Cutler 领导），测试小组（由 Moshe Dunie 和 Ken Gregg 领导），以及文档小组（由 Chris Brown 领导）。我设计并帮助实现了性能监视器，但数量巨大的代码编写工作是由 Hon-Wah Chan 和 Mike Moskowicz 完成的，Bob Watson 也做出了重要贡献，Christos Tsollis 以及所有为 Windows NT 设计计数器的 NT 开发小组成员都给予了很大的帮助。Windows NT 还支持一个强大的应用程序性能调整工作集，它们有的随本书提供，有的由 Windows NT 软件开发包 (SDK) 和设备驱动程序包 (DDK) 提供。这些工具由 Reza Baghai (主要程序员) 实现，协助工作的有 Paramesh Vaidyanathan, Lee Smith, Tom Zakrajsek, Mark Leaming, Mark Lucovsky, Lou Perazoli, Mark Enstrom, 以及 Phillip Marino。另外我要特别感谢本书的编辑人员。如果本书可读性很强，这要归功于无数个不知疲倦的日日夜夜，否则就是由于我不可救药地在一开始就伤害了此书。

所有这些做出贡献的人们都进行了长时间艰辛的劳动，从而才有今天这些最为先进的瓶颈检测和性能规划的工具，并用它们来消除瓶颈。这是一项充满困难，但又必须有人来做的工作。

Russ Blake

1993 年夏

# 目录

绪言 .....	IX
<b>第一章 如何优化 Windows NT .....</b>	<b>1</b>
1.1 可调整的 Windows NT .....	1
1.2 检测瓶颈 .....	2
1.3 性能规划 .....	3
1.4 优化应用程序 .....	3
1.5 性能监视器与其他优秀工具 .....	4
1.6 性能监视器是一个新的应用类型 .....	5
<b>第二章 性能监视的艺术 .....</b>	<b>7</b>
2.1 计算机体系结构 101 .....	8
2.2 瓶颈定义 .....	12
2.3 计数器的作用 .....	13
2.4 为什么不是总能得到有关性能的答案 .....	14
2.4.1 性能监视器是如何监视计算机的 .....	15
2.5 性能监视器概述 .....	16
2.5.1 性能计数器的结构 .....	20
2.5.2 选定计算机 .....	20
2.5.3 选定对象 .....	24
2.5.4 选定计数器 .....	25
2.5.5 选定实例 .....	28
2.5.6 定制显示格式 .....	29
2.6 性能监视器的四个视窗 .....	32
2.6.1 图表视窗 .....	33
2.6.2 报告视窗 .....	38
2.6.3 警报视窗 .....	40
2.6.4 日志视窗 .....	43
2.6.5 装入并观察日志文件 .....	47



2.6.6	保存设置	53
2.6.7	输出性能数据	55
2.7	性能监视器的局限性	55
2.7.1	为什么不能观察到某些磁盘数据	55
2.7.2	为什么处理器队列总是空的	56
2.7.3	打印性能数据的方法	56
2.7.4	GUI 批处理器	57
2.7.5	TCP 与 SNMP	57
2.7.6	重要热键	57
<b>第三章</b>	<b>检测处理器瓶颈</b>	<b>59</b>
3.1	瓶颈的可转移性	60
3.2	启动：创建总设置文件	62
3.3	绘制应面曲线	66
3.4	分析处理器性能	66
3.4.1	性能监视能随意进行吗(不能!)	70
3.4.2	Windows NT 上的处理器调度	71
3.4.3	神秘的锯齿队列长度	73
3.4.4	用户态与特权方式	76
3.4.5	多进程与线程	80
3.4.6	低利用率瓶颈	86
3.4.7	图形系统如何使用处理器	88
3.4.8	16 位 Windows NT 应用对处理器的使用	90
3.4.9	16 位 MS-DOS 应用对处理器的使用	96
3.4.10	谁启动了所有这些进程?	99
3.5	摆脱处理器瓶颈	101
3.6	监视多个处理器	102
<b>第四章</b>	<b>检测磁盘瓶颈</b>	<b>111</b>
4.1	采集磁盘性能统计数据	111
4.2	磁盘繁忙值得高兴	113
4.3	挖掘高的磁盘吞吐率	122
4.4	挖掘更高的磁盘吞吐率	124
4.5	消除磁盘瓶颈	131

---

4.6 考察磁盘阵列 .....	132
<b>第五章 检测内存瓶颈</b> .....	<b>139</b>
5.1 Windows NT 虚存系统是怎样工作的 .....	139
5.1.1 配置 Windows NT 的可用内存 .....	141
5.2 内存活动和页面调度示例 .....	142
5.3 多进程页面调度 .....	152
5.4 监视非页式池 .....	156
5.5 内存短缺对磁盘的影响 .....	157
5.6 什么是内存蚕食 .....	159
<b>第六章 检测 Cache 瓶颈</b> .....	<b>161</b>
6.1 文件系统 Cache 概览 .....	161
6.2 基本 Cache 实验 .....	163
6.3 顺序读写 .....	168
6.4 随机读写 .....	174
6.5 映射文件至内存 .....	179
<b>第七章 检测网络瓶颈</b> .....	<b>185</b>
7.1 网络吞吐量的统计与分析 .....	186
7.2 网络瓶颈的简化模型 .....	192
7.2.1 神秘的遗失时间 .....	193
7.3 进行网络瓶颈检测 .....	195
7.4 比较颠倒后的平台 .....	198
7.5 增加客户至测试服务器 .....	200
7.6 服务器磁盘活动 .....	206
7.7 从服务器向客户拷贝目录 .....	210
7.8 监视 TCP/IP 性能 .....	215
7.9 监视 NWLink 性能 .....	221
<b>第八章 性能规划</b> .....	<b>227</b>
8.1 监视多个服务器 .....	227
8.2 归档和存储性能数据 .....	229
8.3 分析变化趋势 .....	230
8.4 监视桌面计算机 .....	231

<b>第九章 编写高性能的 Windows NT 应用程序</b> .....	233
9.1 内存管理 .....	233
9.2 明智地使用内核 .....	235
9.3 把握图形 .....	236
9.3.1 图形的成批处理 .....	238
9.4 设备描述表的管理 .....	239
9.5 异步输入和窗口管理程序 .....	240
9.6 对 RISC 计算机的考虑 .....	240
9.7 选择 API 集 .....	241
<b>第十章 调整 Windows NT 应用程序</b> .....	245
10.1 首先运行性能监视器 .....	245
10.2 Windows API 统计程序 .....	249
10.2.1 设置统计环境 .....	250
10.2.2 应用程序的统计分析 .....	251
10.2.3 应用程序的 DLLs 列表 .....	251
10.2.4 采集 WAP 数据 .....	251
10.2.5 使用 WAP 时,拒绝对某些 API 的分析 .....	252
10.2.6 对 Solitaire 运行 WAP .....	253
10.2.7 结束 WAP .....	254
10.3 调用属性统计程序 .....	254
10.3.1 使用 CAP .....	256
10.4 Capview: CAP 的直观形式 .....	259
10.5 FIOSAP 统计程序 .....	262
10.5.1 使用 I/O 统计程序 .....	264
10.6 Win32 API 日志记录程序 .....	267
10.7 其他工具 .....	268
10.7.1 PView .....	268
10.7.2 PMon .....	270
10.7.3 WPerf .....	271
10.7.4 符号调试程序的 wt 命令 .....	271

<b>第十一章 调整应用程序工作集</b> .....	273
11.1 工作集调整的作用.....	274
11.2 工作集调整器的工作原理.....	275
11.3 工作集调整器的使用.....	278
11.4 查看工作集的内部.....	279
<b>第十二章 自定义 Windows NT 性能监视程序</b> .....	283
12.1 性能监视器的源代码.....	283
12.2 设计原理.....	284
12.3 性能数据的获取.....	285
12.3.1 性能数据结构.....	287
12.3.2 在性能数据结构中定位.....	289
12.4 获取计数器名称和描述.....	290
12.5 获取选定数据.....	291
12.6 性能计数器定义.....	293
12.7 内部监视应用.....	300
<b>第十三章 增加应用程序性能计数器</b> .....	301
13.1 增加性能计数器.....	301
13.2 对象和计数器设计.....	302
13.3 设置登录表.....	303
13.3.1 创建应用程序性能关键字.....	303
13.3.2 在登录表中加入计数器名称和描述.....	304
13.3.3 从登录表中删除计数器名称和描述.....	308
13.3.4 其他登录表项.....	309
13.4 采集性能数据.....	310
13.5 创建性能 DLL.....	311
13.5.1 如何与性能监视器应用进行交互.....	312
13.5.2 DLL 中的错误处理.....	316
13.6 测量外部计算机.....	316
13.7 应用程序的安装.....	317
13.8 示例代码.....	317
13.8.1 安装 VGA 驱动程序.....	318
13.8.2 数据采集 DLL.....	321

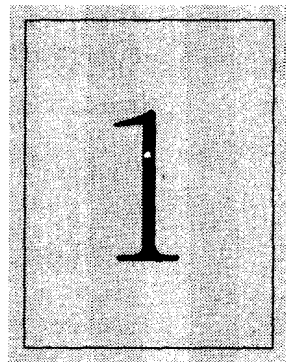
---

<b>附录 A Windows NT 性能计数器</b> .....	353
A.1 Cache 对象 .....	354
A.2 ICMP 对象 .....	358
A.3 图像对象 .....	362
A.4 IP 对象 .....	363
A.5 逻辑盘对象 .....	365
A.6 内存对象 .....	368
A.7 NBT 连接对象 .....	371
A.8 NetBEUI 对象 .....	371
A.9 NetBEUI 资源对象 .....	377
A.10 网络接口对象 .....	377
A.11 NWLink IPX 对象 .....	380
A.12 NWLink NetBIOS 对象 .....	385
A.13 NWLink SPX 对象 .....	390
A.14 目标对象 .....	395
A.15 页面调度文件对象 .....	396
A.16 物理盘对象 .....	397
A.17 进程对象 .....	399
A.18 进程地址空间对象 .....	402
A.19 处理器对象 .....	408
A.20 转发器对象 .....	408
A.21 服务器对象 .....	414
A.22 系统对象 .....	417
A.23 TCP 对象 .....	419
A.24 线程对象 .....	420
A.25 线程细节对象 .....	422
A.26 UDP 对象 .....	422
<b>附录 B 登录表数值项</b> .....	425
B.1 CurrentControlSet\Control 子关键字 .....	426
B.1.1 WOW 启动控制项 .....	426
B.1.2 CurrentControlSet\Services 子关键字中的串行口子关键字数据项 .....	427
B.1.3 鼠标和键盘驱动器项 .....	427

B. 1. 4	SCSI 微端口驱动程序项 .....	430
B. 1. 5	视频设备驱动程序项 .....	432
B. 2	网络服务登录表项 .....	433
B. 2. 1	SFM 的 AppleTalk 和 MacFile 服务项 .....	434
B. 2. 2	DiskPerf 服务项 .....	435
B. 2. 3	EventLog 服务项 .....	436
B. 2. 4	NBF(NetBEUI)传输项 .....	436
B. 2. 5	NetLogon 服务项 .....	440
B. 2. 6	NWLink 传输项(IPX/SPX) .....	440
B. 2. 7	转发器(Rdr) 服务项 .....	443
B. 2. 8	远程访问服务(RAS)项 .....	444
B. 2. 9	服务器服务项 .....	445
B. 2. 10	TCP/IP 服务项 .....	449
B. 2. 11	工作站服务项 .....	452
B. 3	Microsoft Mail 登录表项 .....	456
B. 3. 1	Microsoft Mail 项 .....	456
B. 3. 2	Mail MMF 项 .....	457
<b>附录 C</b>	<b>使用响应探测器 .....</b>	<b>459</b>
C. 1	为什么要使用响应探测器 .....	459
C. 2	响应探测器设计 .....	460
C. 2. 1	正态分布 .....	460
C. 2. 2	THINK 状态 .....	461
C. 2. 3	FILE ACCESS 状态 .....	462
C. 2. 4	COMPUTE 状态 .....	462
C. 3	响应探测器输入文件 .....	462
C. 4	进行响应探测器实验 .....	463
C. 4. 1	描述文件 .....	463
C. 5	输出格式 .....	467

## 第一章

# 如何优化 Windows NT



有许多种方法可以使计算机运行得更快一些——例如,可以把它从十层楼的窗户中扔下去,许多人都曾有这种想法。但最好还是找出慢的原因,以便采取更为合理的措施。

本书将罗列出导致计算机运行减慢的各种因素,特别是那些导致 **Microsoft Windows® NT™** 操作系统运行减慢的因素。并将找出解决这些问题的方法。因为无论是用户的时间,计算机的时间还是网络的时间都是非常宝贵的。

同时还要探讨一下性能规划方面的问题,以便计算机或网络能够在其必需的工作负荷之外还留有一定的余地。并且还会涉及到一些策略和工具,以确保应用程序能更好地在 Windows NT 上。

本章概述了 Windows NT 的性能问题,一开始介绍的是一个真正可靠的系统调整工具,然后介绍 Windows NT 是如何改变传统方式的。我们将给出性能瓶颈的定义以及如何找出它们在系统中出现的位置,如何预测潜在瓶颈以便技术管理者能积累有效的工具。我们将简要地介绍一下程序员对 Windows NT 产生的性能信息可能做出的处理,最后浏览一下本书中包含的主要的性能测试工具。

### 1.1 可调整的 Windows NT

过去,当设计操作系统时,总要定义许多可调参数,通过调整这些参数,来影响系统性能。这些参数频繁地在系统内产生着深刻而灵敏的影响。懂得了这些影响就意味着抓住了那些难以理解的设计细节。事实上,操作系统设计人员非常清楚是他们把最困难的系统性能问题留给了用户。

不幸的是,用户手头上很少有系统的源代码来帮助自己作出决策。在手册保持简单而友

好的同时,文档很少包括合理设置参数所必需的信息。调整一个操作系统变成了一种不可思议的艺术,即要设法弄懂文档提供的大量的参数以及它们对系统性能的影响。这个任务由于参数间的相互作用比参数本身的作用更为复杂而变得更加困难。

Windows NT 的一个主要设计目标就是消除早期操作系统使用的大量的参数。Windows NT 中的种种算法,使得系统在运行时便能确定正确的参数值。32 位地址空间消除了许多内存上的限制并且消除了人为调整内存参数的要求。

Windows NT 从根本上改变了未来计算机的管理方式。优化 Windows NT 的任务并不是人工调整冲突参数的艺术,而是这样一个过程:决定硬件资源遇到的最大需求是什么,然后调整操作以缓和这些需求。系统配备有最优的工具来实现这个任务。描述如何使用这些工具来加快计算机运行的速度是本书的主要目标。

Windows NT 并不是在每种情况下都能自动调整。许多参数还保留着,其主要原因是不能精确地知道每一台计算机如何被使用。所有参数缺省值的设置是面向常规系统一个很宽的使用范围的,这些缺省值很少需要改变。但还是有一些特殊的情况,这时的改变必须合理。本书将提到几个保留在 Windows NT 中的调整参数,以及何时改变它们的缺省值最为合适。

## 1.2 检测瓶颈

也许读者从来没有喝过瓶装啤酒,但是如果喝过就一定会注意到瓶子的颈部比底部狭窄。当把瓶子倒转过来时,狭窄的瓶颈限制了流量,所以只能勉强地喝下一一定量的啤酒来解渴。

对于计算机瓶颈来说,瓶颈就是限制计算机工作流的那一部分。但不像啤酒瓶的瓶颈。计算机的瓶颈可以在系统中的一部分和另一部分之间来回移动。

瓶颈检测是隔离硬件部件的过程,此时硬件部件限制某一活动的流动。但是,通常移动软件优先于移动硬件,所以查找正在活动的软件部件同样是有用的。

例如,假设一台计算机偶尔响应很缓慢。按照本书的指导,很快发现问题是主磁盘一直很忙。这时可先查找这个盘上活动的资源。根据这些活动资源,可以把它们移到另一个磁盘驱动器上,并因此减少了对正常工作的干扰。

软件有时也有故障。这些故障可能导致程序过于浪费硬件资源。如果发现软件有这种



问题,可以替换掉或修改它,把软件从十层楼上扔下去比扔硬件更不甚费力,这点一般来说是正确的。

不论对于桌面计算机上 Windows NT 的单个独立的用户还是管理大量的文件和打印服务器的用户,查找计算机瓶颈的问题和方法都是相同的。在第三章至第七章中将全面讨论有关瓶颈检测的问题。计算机的使用方式不同,对其考虑的方式就不同,这在本书中将会看到。但一般来说,对于瓶颈检测问题都必须按照相同的原则来处理。

### 1.3 性能规划

即使计算机今天充满了活力,但总会有一天它将变得力不从心。因为新软件通常总是使用硬件来完成它的工作。那时就会发现硬件资源无法满足系统的要求。

老板喜欢有关硬件需求的建议,许多资料显示了这种新设备的需求。在第八章中将讨论如何按照一定的原则选择数据,以便预测未来的设备需求。Windows NT 有一些工具,可以方便地为计算机或网络收集性能规划数据。

同时,按照一定的原则监视系统将会产生一些瓶颈检测方面必不可少的信息。一个将详细讨论的主题是性能计数器的设备依赖特性。例如,磁盘驱动器的最大传输率依赖于系统的诸多方面。没有人能提供一个非常“巧妙”的数据。但是由于有一个计算机正常操作的记录,所以可以建立一个易于理解的合理的计数器值。然后,如果要做改变或者发现速度慢了下来的时候,就有了一个参考基准来同新情况作比较。没有这个基准,瓶颈检测将会很棘手。

### 1.4 优化应用程序

在 Windows NT 上开发软件时,其先进的特性可使应用程序变得更快更好。有一点很重要,即为 16 位计算机开发基于 Windows 的应用程序与全新的 Windows NT 32 位体系结构不再有什么关系。在第九章中将详细讨论这些问题。

对于一个冷酷的程序员的患者来说,第九章将提供许多保护自己的武器。利用这些武器可以测定出这些程序员是否能使用正确的技术从 Windows NT 获得最大好处。想象着自己正对着他们的脸,然后向他们指出应该变换 WIZBANG.DAT 文件,这时请把精彩的录相带寄给我们!

如果程序员正在 Windows NT 上开发软件,便可以立即使用(通过 Windows NT 软件开