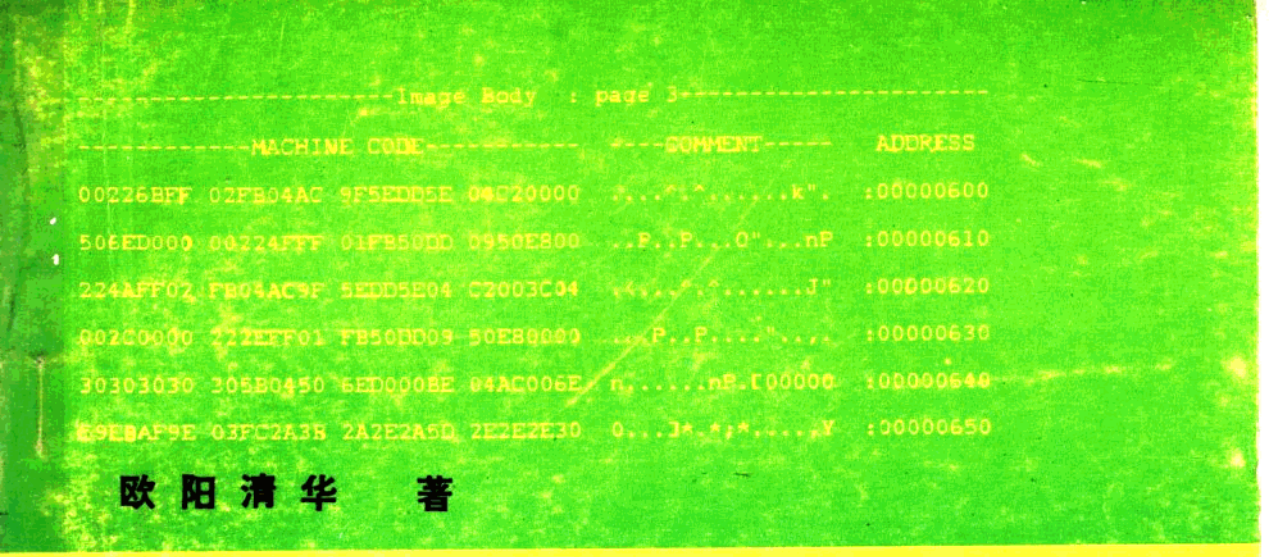

欧阳清华 著

反汇编原理及其实现技术

```
WUHN602:        SUBL2   S^#^X4,SP
WUHN605:        PUSHL   SP
WUHN607:        PUSHAB  B^4(AP)
WUHN60A:        CALLS   S^#^X2,G^LIB$GET_VM
WUHN611:        BLBS    R0,B^WUHN61D
WUHN614:        PUSHL   R0
WUHN616:        CALLS   S^#^X1,G^LIB$SIGNAL
WUHN61D:        MOVL    (SP),R0
WUHN620:        RET
WUHN621:        MOVZWL  S^#^X0,W^24068(R2)
WUHN626:        PUSHL   SP
WUHN628:        PUSHAB  B^4(AP)
WUHN62B:        CALLS   S^#^X2,G^LIB$GET_VM
```

武汉大学出版社

反汇编原理及其实现技术

欧阳清华　著

武 汉 大 学 出 版 社

内 容 提 要

反汇编程序要将机器语言程序翻译为等效的汇编语言程序。机器语言程序是二进制代码；每个字节的二进制代码既可以是指令，也可以是数据。就是说，机器语言一般是二义性语言。翻译二义性语言目前尚无全自动方法，具有这种二义性机器语言的计算机也没有反汇编程序。本书提出二义性机器语言的半自动反汇编，并且在 VAX 机上实现，这在国内属于首创。

翻译二义性机器语言程序，首先要将指令和数据分开，然后分别将指令和数据翻译出来。这就要有指令识别程序，指令翻译程序和数据翻译程序；这三个程序是反汇编的核心。本书提出的这三个程序不仅是精巧的作品，而且具有普遍性的意义。

反汇编原理及其实现技术

欧阳清华 著

*

武汉大学出版社出版发行

(430072 武昌 珞珈山)

武汉水利电力学院印刷厂印刷

*

787×1092 1/16 11.375 印张 264 千字

1992 年 10 月第 1 版 1992 年 10 月第 1 次印刷

印数 1—2500

ISBN 7-307-01353-3/TP・40

定价：11.00 元

序　言

汇编程序将汇编语言程序翻译为可执行的机器语言程序，反汇编程序为其逆，将可执行的机器语言程序翻译为等效的汇编语言程序。从分析机器代码出发，或者仅从逆向翻译出发，普遍性地解决反汇编的问题都是有意义的。

汇编语言为一义性语言，汇编程序在将汇编语言程序翻译成机器代码程序时，自顶向下将源程序语句一句一句地翻译成二进制的机器代码。对于指令和数据可以按任意次序排列的汇编语言来说，其对应的机器码程序就不是一义性的了。反汇编程序在将这种机器代码翻译为汇编语言程序时，它所读到的每一个字节的二进制代码，既可以是指令的一部分，也可以是数据的一部分。试图全自动地区别指令和数据是不可能的；这等于是说，一般地讲，全自动反汇编是不可能的。VAX 计算机就是这种情况，它的厂家也未能提供反汇编程序。我们从原理上和技术上解决了反汇编问题，并实际地在 VAX 机上实现了。我们发展了一种半自动可控块扫描技术(详细情形见第 4 章)，利用它可以将 VAX 的机器码程序近于自动地翻译出来。这个技术具有普遍性的意义，可以用于任何计算机的反汇编。

反汇编是要将机器码翻译成汇编语言程序，首先必需要有能够区别指令和数据的程序；区别了指令和数据之后，必须有翻译指令的程序和翻译数据的程序，以便将指令和数据分别翻译出来。所以，反汇编的基本功能模块必须包括：

1. 指令识别程序；
2. 指令翻译程序；
3. 数据翻译程序；

反汇编所操作的源程序是机器码，必需要有阅读机器码的程序，使机器码程序像其它语言的源程序一样成为可读的。虽然多数计算机随机带有打印机器码的程序，那些都不能满足反汇编的特殊需要，反汇编必须有更强的阅读机器码的手段。

半自动可控块扫描采用交互式的工作方式，而交互式就必须有一个命令语言解释程序(Command Language Interpreter)。

最后，为了使各功能模块根据人工命令有效地运行起来，还必须有完善的控制程序。

所以，辅助功能模块应当有

4. 阅读机器码的程序；
5. 命令语言解释程序 CLI(Command Language Interpreter)；
6. 控制程序；

反汇编的工作过程是这样的：

第一步：找出所有的指令，将它们的地址填入指令地址表中；

第二步：用互补原理(第 6 章)找出所有的数据，将它们的地址填入数据地址表中；

第三步：按照指令地址表和数据地址表中的地址，将指令和数据翻译出来。

由于机器语言或汇编语言代表具体计算机的功能，随计算机的种类而不同，当反汇编在一种特定的机型上实现时，必然还要有与机型有关的特定的功能模块。我们选择在 VAX 机上实现反汇编，对于 VAX 计算机而言，这就需要有

7. 系统符号的识别和翻译程序(第 7 章)。

反汇编程序的结构如下图所示，总控程序通过命令语言解释程序接受人工命令，根据人工命令选择性地调用其余的模块。

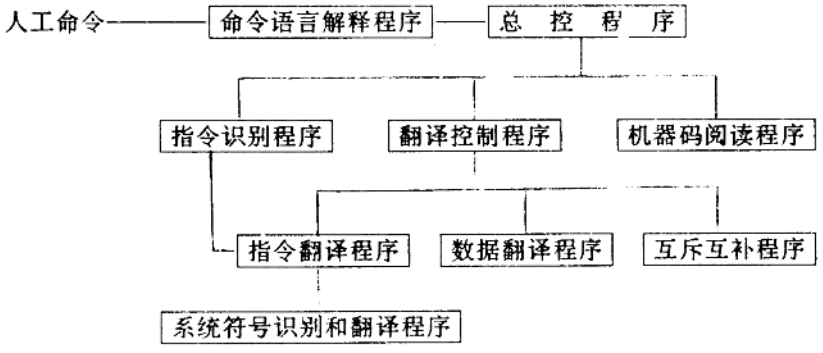

图 0.0 反汇编程序框图

与编译程序的行扫描方式不同，指令识别程序是一个块扫描程序；总控程序根据人工命令可以调用这个块扫描程序对机器码自顶向下反复地扫描以找出所有的指令地址和数据地址。找出的指令地址填入指令地址表中，找出的数据地址填入数据地址表中。对指令地址表和数据地址表进行互斥互补处理之后，根据指令地址表和数据地址表分别用指令翻译程序和数据翻译程序自顶向下地将指令和数据翻译出来。本书从第 1 章开始将详细展示上述各步骤的细节。第 1 章至第 6 章是反汇编的基本原理及其实现技术。虽然这几章的讨论似乎是与 VAX 计算机联系在一起，那只不过是为了使讨论的问题具体罢了。讨论中以 VAX 汇编语言(包括汇编语言语句格式、寻址方式、指令格式和数据类型等)为讨论素材，然而这些素材也是其它计算机汇编语言所共有的。把这几章中有关 VAX 计算机的数据换成其它计算机的，结论仍然成立。只有第 7 章，即最后一章，是专为 VAX 计算机设立的。

本书采用 C 语言来描述模块和算法，书中对于某些重要模块的描述就是从实际模块中摘录出来的。为了便于不太熟悉 C 语言的读者阅读，力求使摘录出来的 C 语言程序尽可能简单。在描述算法的地方还加上汉字注释，使之类似于伪代码的形式。书后附录 A 有关于 C 语言的简单介绍。这样，即使从未接触过 C 语言的读者，也能够理解本书关于反汇编程序的描述。对于熟悉 C 语言的读者，读完本书之后，就可以动手写一个简单的反汇编程序。

在 C 语言中，赋值号用'='，等号用'=='。例如 A=B，意思是 B 的值赋给 A；A==B，意思是 A 等于 B。在 C 语言以外，=仍然表示等于号。由于本书采用 C 语言描述模块和算法，符号'=='和'='用得很多，请读者注意它们的区别，以免混淆。

反汇编程序既然要将机器语言程序翻译为等效的汇编语言程序，自然要彻底地明白机器

语言和汇编语言。这两种语言的详尽描述应从提供计算机的厂家获得。对于 VAX 计算机，有关的信息主要列在厂家提供的 "VAX MACRO and Instruction Set" 一书中。本书后面的附录 B 有关于 VAX 汇编语言的提要，这对于不熟悉汇编语言的读者是有用的。

尽管有的计算机带有反汇编，但是不仅尚无一本阐述反汇编原理的书，就是这方面的文献也似属空白。本书是解决这个问题的第一个尝试。作者学识有限，遗漏和错误之处在所难免，诚望读者和专家们赐教。

目　录

第 1 章　机器码的阅读

1.1　阅读机器码的程序 dump

高级语言程序最先是写在纸上，可阅读是自然的。机器码程序最先是存在于机器之中，首先要解决可阅读性问题。多数计算机提供有一个诸如 DUMP 的实用程序，可以将机器码程序用一位十六进制数字对应四位二进制代码的形式表示出来。DUMP 将机器代码分页读出，每页 512 字节。

机器提供的 DUMP 的最大缺点是每页的地址都是从 0 到 1FF(512)，没有机器码的真实地址，不能看出上下文的关系。

在动手作反汇编程序之前，先要做一个阅读机器码程序的工具。按习惯，我们做的阅读机器码的工具也称之为 dump。

要对机器码进行处理，先要将它从磁盘上读到内存的一个缓冲区来。为确定起见，我们称此缓冲区为 m __buf[size]。size 是 m __buf 的字节数，m __buf 的下标从 0 到(size − 1)。size 的大小以能容纳最大的机器码程序为度。就现有 VAX/VMS 各机器码程序而言，size 等于 10 兆就足够了。VAX 提供给用户编程的空间是虚拟空间，它远比实际的物理内存大，约 1073 兆字节。在此虚拟空间中设置一个 10 兆字节的缓冲区是不成问题的。对于编程空间少于物理内存的的机器(主要是微机)，不能设立大的缓冲区以容纳最大的机器码程序时，我们就得自己解决虚拟空间技术问题，这个问题不是反汇编本身的问题，限于篇幅，就不在此讨论了。

计算机处理数据的时候，总是先将数据从磁盘读到内存再进行处理。内存的大小直接影响到计算机处理数据的速度。不过，用户一般并不直接与内存打交道。用户编程的时候，他所关心的是他的程序最大允许有多大，或者说他编程的空间有多大。用户编程的空间称为程序空间(Program Space)。程序空间不是内存，内存一般是指计算机的物理内存。在早期，程序空间一般比内存小；现代的计算机，程序空间一般比内存大。VAX 就是如此，详细情形见第 7 章。本书以后一般只说程序空间。倘若提到内存，但未指出是物理内存，则都应理解为程序空间。

VAX 的机器码程序分为两种：一种是由源语言程序经过编译但未经链接程序连接的机器码，称为目标程序或目标码(object)；另一种是由目标程序经链接程序连接之后得到的机器码，称为映像(image)。因为目标码可以变为映像再处理，下面我们只就映像讨论就可以了。

被反汇编的映像读入到 m __buf，映像在 m __buf 中的地址安排要和此映像在运行时装入程序空间的地址安排完全一致：映像在程序空间运行时地址为 n 的字节被装入 m __buf[n]。这样，m __buf 就摸拟着映像运行时占据的程序空间：m __buf[n]的下标 n 代表映像运行时一个字节的地址；m __buf[n]的内容就是映像中地址 n 的字节中的内容。以后，对映像的处理就变成对 m __buf 的内容的处理。

将字节 m __buf[n]的内容用两位十六进制数字表示出来，n 代表此字节的地址。从 n=0

开始，每 512(十六进制 200)个字节为一页，将 m__buf 分页打印出来，就得到上下文连续的可阅读的机器码程序了。图 1.1.1 是机器码阅读程序 dump 读出的 VAX 系统程序

SYS$SYSTEM:DELETE.EXE

的一部分。

```
----------Image Header: page 0----------
------MACHINE CODE--------COMMENT---ADDRESS
35303230  00000000  00580044  003000A8 ..0.D.X.....0205 ;00000000
00000000  FFFFFFFF  FFFFFFFF  00000101 ................ ;00000010
00002800  00000000  45180927  01000028 (...'..E....(.. ;00000020
00000000  00000000  00000000  00001414 ................ ;00000030
00000000  00000000  00000000  00000000 ................ ;00000040
00455445  4C454406  00000000  00000000 .........DELETE. ;00000050
00000000  00000000  00000000  00000000 ................ ;00000060
00000000  00000000  00000000  00000000 ................ ;00000070
00000000  00000000  00000000  322D5803 .X-2............ ;00000080
00003030  2D343005  008EC518  0927BF80 ..'.....04-00.. ;00000090
00000001  00020010  00000000  00000000 ................ ;000000A0
00000003  00110010  00000002  0000008A ................ ;000000B0
00000014  00010010  00000004  00000080 ................ ;000000C0
003FFFEC  0014000C  00000015  0000040A ..............?. ;000000D0
03000021  00000000  0081001F  FD00008C .............!... ;000000E0
5F4C5452  42494C0A  0100000D  00000000 .........LIBRTL__ ;000000F0
FFFFFFFF  FFFFFFFF  FFFFFF00  00313030 001............. ;00000100
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000110
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000120
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000130
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000140
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000150
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000160
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000170
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000180
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;00000190
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;000001A0
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;000001B0
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;000001C0
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;000001D0
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;000001E0
FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF................ ;000001F0
----------Image Body: page 1----------
------MACHINE CODE--------COMMENT---ADDRESS
00000000  00000000  00000000  00000000 ................   ;00000200
00000000  00000000  00000000  00000001 ................   ;00000210
```

```
00000000 00000000 00000000 00000001 ..... ;00000220
00000000 00000000 00000000 00000000 ..... ;00000230
00000000 00000000 00000000 00000000 ..... ;00000240
00000000 00000000 00000000 00000000 ..... ;00000250
00000000 00000000 00000000 00000000 ..... ;00000260
00000000 00000000 00000000 00000000 ..... ;00000270
00000000 00000000 00000000 00000000 ..... ;00000280
00000000 00000000 00000000 00000000 ..... ;00000290
00000000 00000000 00000000 00000000 ..... ;000002A0
00000000 00000000 00000000 00000000 ..... ;000002B0
00000000 00000000 00000000 00000000 ..... ;000002C0
00000000 00000000 00000000 00000000 ..... ;000002D0
00000000 00000000 00000000 00000000 ..... ;000002E0
00000000 00000000 00000000 00000000 ..... ;000002F0
00000000 00000000 00000000 00000000 ..... ;00000300
00000000 00000000 00000000 00000000 ..... ;00000310
00000000 00000000 00000000 00000000 ..... ;00000320
00000000 00000000 00000000 00000000 ..... ;00000330
00000000 00000000 00000000 00000000 ..... ;00000340
00000000 00000000 00000000 00000000 ..... ;00000350
00000000 00000000 00000000 00000000 ..... ;00000360
00000000 00000000 00000000 00000000 ..... ;00000370
00000000 00000000 00000000 00000000 ..... ;00000380
00000000 00000000 00000000 00000000 ..... ;00000390
00000000 00000000 00000000 00000000 ..... ;000003A0
00000000 00000000 00000000 00000000 ..... ;000003B0
00000000 00000000 00000000 00000000 ..... ;000003C0
00000000 00000000 00000000 00000000 ..... ;000003D0
00000000 00000000 00000000 00000000 ..... ;000003E0
00000000 00000000 00000000 00000000 ..... ;000003F0
----------Image Body: page 2----------
-----MACHINE CODE-------COMMENT--ADDRESS
00000000 00000000 00000000 00000000 ..... ;00000400
00000000 00000000 00000000 00000000 ..... ;00000410
00000000 00000000 00000000 00000000 ..... ;00000420
00006002 00000000 00000000 00000000 .....'.. ;00000430
00000000 00000000 00000000 00000000 ..... ;00000440
00000000 00000000 00000000 00000000 ..... ;00000450
00000000 00000000 00000000 00000000 ..... ;00000460
00000000 00000000 00000000 00000000 ..... ;00000470
00000000 00000000 00000000 00000000 ..... ;00000480
00005813 00000000 00000000 00000000 .....X.. ;00000490
```

```
00000000  00000000  0000FFFF  00000000  ................  ;000004A0
00000000  00000000  00000000  00000000  ................  ;000004B0
00000000  00000000  00000000  00000000  ................  ;000004C0
00000000  00000000  00000000  00000000  ................  ;000004D0
00000000  00000000  00000000  00000000  ................  ;000004E0
00000000  0000049C  00002C12  00000000  .....,.........  ;000004F0
00000000  00000000  00000000  00000000  ................  ;00000500
00000000  00000000  00000000  00000000  ................  ;00000510
00000000  00000000  00000000  00005003  .P..............  ;00000520
02000000  00000000  00020000  00000000  ................  ;00000530
00000000  0000043C  000004F4  00000000  ........<.......  ;00000540
00000000  00000000  00000000  00000000  ................  ;00000550
00000000  00000000  00000000  00000000  ................  ;00000560
00000000  00000000  00000000  00000000  ................  ;00000570
00000000  00000000  00000000  00000000  ................  ;00000580
00000000  00000000  00000000  00000000  ................  ;00000590
00000000  00000000  00000000  00000000  ................  ;000005A0
00000000  00000000  00000000  00000000  ................  ;000005B0
00000000  00000000  00000000  00000000  ................  ;000005C0
00000000  00000000  00000000  00000000  ................  ;000005D0
00000000  00000000  00000000  00000000  ................  ;000005E0
00000000  00000000  00000000  00000000  ................  ;000005F0
----------Image Body: page 3----------
------MACHINE CODE----------COMMENT----ADDRESS
00226BFF  02FB04AC  9F5EDD5E  04C20000  ....^.^.....k".  ;00000600
506ED000  00224FFF  01FB50DD  0950E800  ..P..P...O"...nP  ;00000610
224AFF02  FB04AC9F  5EDD5E04  C2003C04  .<...^.^.....J"  ;00000620
002C0000  222EFF01  FB50DD09  50E80000  ...P..P...".,..  ;00000630
30303030  305B0450  6ED000BE  04AC006E  n......nP.[00000  ;00000640
59EBAF9E  03FC2A3B  2A2E2A5D  2E2E2E30  0...]*.*;*.....Y  ;00000650
AED47ED4  5EFDECCE  9E580000  2206FF9E  ...".X....^.~..  ;00000660
B0AD5003  8FB0B0AD  00508F00  6E002C04  .,.n..P......P..  ;00000670
ACD0D8AD  FF50CD9E  CFAD0290  C6AD0290  ..........P.....  ;00000680
90E4AD67  90E0AD69  9EDCAD04  A7D05704  .W......i...g...  ;00000690
028FB0FF  50CD0060  8F006E00  2CE5AD10  ...,.n..'..P....  ;000006A0
D0FF5CCD  08AE9EFF  5ACD018E  FF50CD60  '.P....Z.....\..  ;000006B0
006E002C  FF60CD01  08CE9E23  13560CAC  ..V.#.....'.,.n.  ;000006C0
CE669001  08CE6002  8FB00108  CE00608F  .'.......'....f.  ;000006D0
E2309F01  FBB0AD9F  FEF4CD04  A6D0010A  ..............0.  ;000006E0
FF54CDD0  081351FF  53CD9A3C  50E87FFE  ...P<..S.Q....T.  ;000006F0
AEFF5CCD  D00B1351  FF5BCD9A  0D1104AE  ......[.Q....\..  ;00000700
50DD6E67  B004AE04  A7D00811  6E51B004  ..Qn........gn.P  ;00000710
```

```
05FB0003  132C8FDD  02DD08AE  9F10ACDD  ..........,.....  ;00000720
00060080  8F84ADD3  0A1353FF  5BCD9A68  h..[.S.........  ;00000730
FB000313  2C8FDD02  DD57DD10  ACDD1013  ......W....,....  ;00000740
DD02A201  0E8FB062  53B05208  ACD06804  .h...R.Sb.......  ;00000750
04B208AE  532804A2  50D0FE9A  CF01FB53  S......P..(S....  ;00000760
B118A051  D0511CAC  D05020AC  D0000004  ..... P...Q.Q...  ;00000770
FF8F18A0  B11CA002  8804123F  FF8F1AA0  ....?...........  ;00000780
04ACD000  04045001  D01CA001  880412FF  .........P......  ;00000790
DD091302  A2B5520C  ACD0171C  A004E050  P........R......  ;000007A0
08ACD024  A210ACD0  7FFEE1B8  9F01FB52  R........... $...  ;000007B0
8F52D112  13001590  088F52D1  1B52E852  R.R..R........R.  ;000007C0
D0000020  8DFF01FB  52DD0913  00159018  ........R.... ...  ;000007D0
45534C41  46315345  59455552  54045052  RP. TRUEYES1FALSE  ;000007E0
00000014  00005449  55514C4C  41304F4E  NO0ALLQUIT......  ;000007F0
```

————————————Image Body: page 4————————————

```
——————MACHINE CODE——————————COMMENT————ADDRESS
000007E7  00030000  00010000  07E30004  ................  ;00000800
00050000  00010000  07EA0001  00000001  ................  ;00000810
90100000  07F00002  00159010  000007EB  ................  ;00000820
07F30003  00159010  000007F2  00010015  ................  ;00000830
00159018  000007F6  00040015  90200000  .. ............  ;00000840
8FD00812  67B55704  ACD059AB  AF9E03FC  .....Y...W.g....  ;00000850
001FDFFF  02FB57DD  57DD0450  00159010  ....P..W.W......  ;00000860
5401CE58  500AC750  FCA90278  55673C00  .<gUx...P..PX..T  ;00000870
1E506946  1000ED50  55D05654  0AC52D11  .—..TV.UP...FiP.  ;00000880
552D519E  D002A946  9F509E3C  69469F06  ..Fi<.P.F....Q—U  ;00000890
F204509E  D006A946  9F081261  500004B7  ...Pa...F....P..  ;000008A0
08ACD056  FEDFCF9E  007C0450  D4CF5458  XT..P.|.....V...  ;000008B0
50D05014  ACD0311C  A404E054  04ACD053  S...T....1...P.P  ;000008C0
10BC18AC  D004A018  ACD00CBC  50D024A3  .$.P............  ;000008D0
1CACDD62  50E87FFE  E2089F01  FB08ACDD  ..........Pb...  ;000008E0
52640102  EF046604  FB54DD50  DD08ACDD  ....P.T..f....dR  ;000008F0
01A40100  EF5252D2  5250C850  640103EF  ...dP.PR.RR.....  ;00000900
11123155  E80352E9  5324A3D0  5555D255  U.UU..$S.R..U1..  ;00000910
04FB54DD  001583A4  8FDD08AC  DD1CACDD  ..............T.  ;00000920
13639152  01D00CBC  53D00712  12639166  f.c....S....R.c.  ;00000930
04C91153  04A3D055  01D010BC  53D00712  ...S....U...S...  ;00000940
00000950  00000007  004D5249  464E4F43  CONFIRM.....P...  ;00000950
00000960  00000007  00454455  4C435845  EXCLUDE.....'...  ;00000960
00000970  00000006  00004552  4F464542  BEFORE......p...  ;00000970
00000980  00000005  00000045  434E4953  SINCE...........  ;00000980
00000990  00000007  00444554  41455243  CREATED.........  ;00000990
```

```
000009A0  00000008  44454946  49444F4D  MODIFIED.......  ;000009A0
000009B0  00000007  00444552  49505845  EXPIRED........  ;000009B0
000009C0  00000006  00005055  4B434142  BACKUP.........  ;000009C0
000009D0  00000008  52454E57  4F5F5942  BY__OWNER......  ;000009D0
00000000  00000000  00000000  00000000  ................  ;000009E0
00000000  00000000  00000000  00000000  ................  ;000009F0
- - - - - - - - - - - - Image Body; page 21 - - - - - - - - - - - -
- - - - - MACHINE CODE - - - - - - - - - - - - COMMENT - - - - - ADDRESS
00000040  00000040  00000000  00000000  ........@...@...  ;00002A00
00000002  00000090  00000084  00000000  ................  ;00002A10
00000000  00000000  00000000  00000000  ................  ;00002A20
00000000  00000000  00000000  00000000  ................  ;00002A30
00000698  00000660  00000001  0000000E  ........'.......  ;00002A40
000003E8  000003F0  00000378  00000778  x...x...........  ;00002A50
000004F0  000004F8  000004A0  00000408  ................  ;00002A60
00000550  00000548  00000530  00000500  ....0...H...P...  ;00002A70
000D0001  00002600  00000001  00000000  ........&.......  ;00002A80
00000000  00000000  00000000  00000000  ................  ;00002A90
00000000  00000000  00000000  00000040  @...............  ;00002AA0
00000000  00000000  00000000  00000000  ................  ;00002AB0
00000000  00000000  00000000  00000000  ................  ;00002AC0
00000000  00000000  00000000  00000000  ................  ;00002AD0
004C5452  42494C06  00000000  00000000  .........LIBRTL.  ;00002AE0
00000000  00000000  00000000  00000000  ................  ;00002AF0
00000000  00000000  00000000  00000000  ................  ;00002B00
00000000  00000000  00000000  00000000  ................  ;00002B10
00000000  00000000  00000000  00000000  ................  ;00002B20
00000000  00000000  00000000  00000000  ................  ;00002B30
00000000  00000000  00000000  00000000  ................  ;00002B40
00000000  00000000  00000000  00000000  ................  ;00002B50
00000000  00000000  00000000  00000000  ................  ;00002B60
00000000  00000000  00000000  00000000  ................  ;00002B70
00000000  00000000  00000000  00000000  ................  ;00002B80
00000000  00000000  00000000  00000000  ................  ;00002B90
00000000  00000000  00000000  00000000  ................  ;00002BA0
00000000  00000000  00000000  00000000  ................  ;00002BB0
00000000  00000000  00000000  00000000  ................  ;00002BC0
00000000  00000000  00000000  00000000  ................  ;00002BD0
00000000  00000000  00000000  00000000  ................  ;00002BE0
00000000  00000000  00000000  00000000  ................  ;00002BF0
```

图 1.1.1　dump 读出的机器码程序

每页机器码程序分为三个纵区，它们分别以"MACHINE CODE"、"COMMENT"和"ADDRESS"为首。在ADDRESS下面的每一个数是十六进制数字的地址，相邻两个地址相差16。这就是说，MACHINE CODE下面的每一行是16个字节的代码，每个字节用两位十六进制数字表示。16个字节对应32个十六进制数字。32个十六进制数字分又分成4组；每组8个十六进制数字代表着4个字节的内容。我们以地址00000000（在page 0上）所在的那一行为例，

35303230 00000000 00580044 003000A8 . . 0. D. X. . . . . 0205 ;00000000

MACHINE CODE下面的字节的地址从右向左递增。地址00000000那个字节的内容是A8，地址00000001那个字节的内容是00，地址00000002那个字节的内容是30，地址00000003那个字节的内容是00，地址00000004那个字节的内容是41，……，地址0000000F那个字节的内容是35。

COMMENT下面的字符的地址从左向右递增。COMMENT的下面每行有16个可印刷字符，它们是机器码的字符表示。当一个字节的机器码恰好是某个可印刷字符的值时，便用那个字符表示出来；当一个字节的机器码不是任何一个可印刷字符的值时，一律用一个点表示。地址00000000那个字节的内容A8不是某个可印刷字符的值，在COMMENT下面对应一个点；地址00000001那个字节的内容00也不是某个可印刷字符的值，在COMMENT下面也对应一个点；地址00000002那个字节的内容30是可印刷字符"0"的值，在COMMENT下面对应字符"0"；……；地址0000000F那个字节的内容35是可印刷字符"5"的值，在COMMENT下面对应字符"5"。

1.2 dump()的构造

我们用一个C函数来实现前节提到的DUMP的功能，这C函数不妨也叫做dump()。

作为阅读机器码的工具的dump()的功能应当包括：

(1) 将字节m__buf[n]用两位十六进制数字表示出来，n代表此字节的地址；

(2) 从n=0开始，每512字节为一页；

(3) 从指定的一页page1读起，读到指定的另一页page2止；

(4) 读出的内容既可从屏幕输出，也可以写进一个文件。

实现上述功能的dump()构造如下：

```
dump(page1,page2,pt,q2,par1)
int page1,page2;              char *pt,q2[],par1[];
{
int f,i,k,page__count,addr,pageend,displ,zero;
char addrchr[10],chr[17],dispchr[4][10],buf[80];
FILE *fpd;

page__count=page1; f=1;
if(q2[0]!='\0') fpd=fopen(q2,"w");
if(q2[0]=='\0') printf("\ndump: page1=%d page2=%d\n",page1,page2);
while(page__count<=page2)
```

```
{
k=addr=0X200*(page__count);pageend=0X200*(page__count+1);
while(k<pageend && pt[k]=='\0') k++;
if((k-addr)==0X200) zero=1; else zero=0;
if(q2[0]=='\0')
 {
 if(par1[0]=='H')
   {
   printf("\n\n----------Image Header: page %d",page__count);
   printf("----------\n");
   }
 else
  {
  printf("\n\n----------Image Body : page %d",page__count);
  printf("----------\n");
  if(zero)
    {
    printf("----------Zero page ! ----------\n");
    goto quit;
    }
  }
  printf("------MACHINE CODE--------COMMENT--ADDRESS\n");
  }
 else
  {
  if(par1[0]=='H')
    {
    fprintf(fpd,"\n\n----------Image Header: page %d",page__count);
    fprintf(fpd,"----------\n");
    }
  else
    {
    fprintf(fpd,"\n\n----------Image Body : page %d",page__count);
    fprintf(fpd,"----------\n");
    if(zero)
      {
      fprintf(fpd,"----------Zero page ! ----------\n");
      goto quit;
      }
    }
   fprintf(fpd,
       "------MACHINE CODE--------COMMENT--ADDRESS \n");
```

```
    }
  while(addr<pageend)
    {
    for(i=0;i<16;i++)
      {
      if(isprint(pt[addr+i])&&(pt[addr+i]>0X1F))
       chr[i]=pt[addr+i];
      else chr[i]='.';
      } chr[16]=chr[17]='\0';
    sprintf(addrchr,"%X",addr); k=strlen(addrchr);
    if(k<8)
      {
      for(i=0;i<=8-k-1;i++) buf[i]='0';buf[i]='\0';
      strcat(buf,addrchr);strcpy(addrchr,buf);addrchr[8]='\0';
      }
    for(i=0;i<=3;i++) { longw(pt,addr,&displ,dispchr[i]);addr=addr+4; }
    strcpy(buf,dispchr[3]);
    for(i=2;i>=0;i--) { strcat(buf," "); strcat(buf,dispchr[i]); }
    strcat(buf," ");strcat(buf,chr);
    strcat(buf," :"); strcat(buf,addrchr); strcat(buf,"\n");
    if(q2[0]=='\0') printf("%s",buf);
    else fprintf(fpd,"%s",buf);
    }

  quit: page__count++;
  }
exit: if(q2[0]!='\0') fclose(fpd); return;
}
```

图 1.2.1　机器码阅读程序

函数 dump(page1,page2,pt,q2,par1) 带五个形式参数：page1,page2,pt,q2,par1。在形式参数说明

```
int     page1,page2;     char *pt,q2[],par1[];
```

中，不带星号的变量(page1,page2)为传值参数，带星号的变量(*pt,*q2,*par1)为传址参数。在 C 语言中，字符串参数皆为传址参数，不管是否带星号。

参数 page1 是被读出的存储区的起始页号，page2 是结束页号。

参数 pt 是指向被读存储区首地址的指针。当 pt 被某实际存储区替换时，dump()就读出该存储区。例如 dump(page1,page2,m__buf,g2,par1)读出 m__buf，而 dump(page1,page2,buf1,q2,par1) 和 dump(page1,page2,buf2,q2,par1) 就分别读出 buf1 和 buf2 两个存储区。可见上面构造的 dump()不仅限于读出 m__buf 的内容，而是可以读出任何一个存储区的内容。

参数 q2 是读出的内容向何处输出的指针。当 q2='\0' 时从屏幕输出；否则写进一个文