软件安全

C2 计算机引导与磁盘管理

彭国军　《软件安全》课程组
武汉大学国家网络安全学院
guojpeng@whu.edu.cn

提纲

2.1 系统引导与控制权
2.2 80X86处理器的工作模式
2.3 Windows内存结构与管理
2.4 磁盘的物理与逻辑结构
2.5 FAT32文件系统及数据恢复
2.6 NTFS文件系统

2.6 NTFS文件系统

NTFS文件系统总体结构图

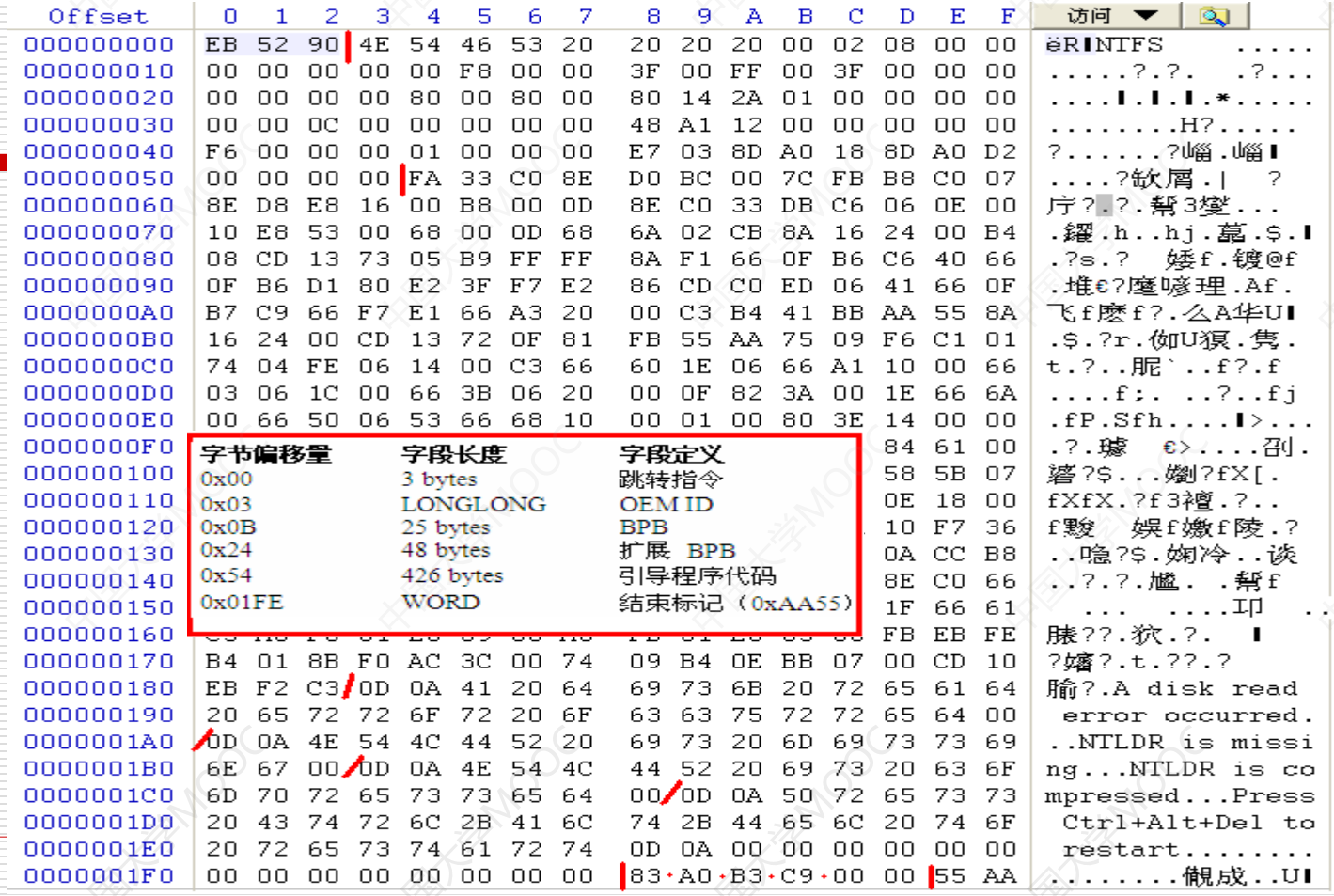
1：1个引导扇区和15个扇区的NTLDR区域

2：MFT元数据文件

3：MFT前几个数据文件的备份

- MFT（主控文件表）是NTFS卷结构的核心。
  - MFT是一个与文件相对应的文件属性数据库，它记录了除文件数据外的所有属性，甚至小文件的数据本身也包含在MFT中。
  - MFT以文件数组来实现，每个文件记录的大小固定为1KB。

NTFS 引导分区



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000000000	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00
000000010	00	00	00	00	00	F8	00	00	3F	00	FF	00	3F	00	00	00
000000020	00	00	00	00	80	00	80	00	80	14	2A	01	00	00	00	00
000000030	00	00	0C	00	00	00	00	00	48	A1	12	00	00	00	00	00
000000040	F6	00	00	00	01	00	00	00	E7	03	8D	A0	18	8D	A0	D2
000000050	00	00	00	00	FA	33	C0	8E	D0	BC	00	7C	FB	B8	C0	07
000000060	8E	D8	E8	16	00	B8	00	0D	8E	C0	33	DB	C6	06	0E	00
000000070	10	E8	53	00	68	00	0D	68	6A	02	CB	8A	16	24	00	B4
000000080	08	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66
000000090	0F	B6	D1	80	E2	3F	F7	E2	86	CD	C0	ED	06	41	66	0F
0000000A0	B7	C9	66	F7	E1	66	A3	20	00	C3	B4	41	BB	AA	55	8A
0000000B0	16	24	00	CD	13	72	0F	81	FB	55	AA	75	09	F6	C1	01
0000000C0	74	04	FE	06	14	00	C3	66	60	1E	06	66	A1	10	00	66
0000000D0	03	06	1C	00	66	3B	06	20	00	0F	82	3A	00	1E	66	6A
0000000E0	00	66	50	06	53	66	68	10	00	01	00	80	3E	14	00	00
0000000F0														84	61	00
000000100														58	5B	07
000000110														0E	18	00
000000120														10	F7	36
000000130														0A	CC	B8
000000140														8E	C0	66
000000150														1F	66	61
000000160														FB	EB	FE
000000170	B4	01	8B	F0	AC	3C	00	74	09	B4	0E	BB	07	00	CD	10
000000180	EB	F2	C3	0D	0A	41	20	64	69	73	6B	20	72	65	61	64
000000190	20	65	72	72	6F	72	20	6F	63	63	75	72	72	65	64	00
0000001A0	0D	0A	4E	54	4C	44	52	20	69	73	20	6D	69	73	73	69
0000001B0	6E	67	00	0D	0A	4E	54	4C	44	52	20	69	73	20	63	6F
0000001C0	6D	70	72	65	73	73	65	64	00	0D	0A	50	72	65	73	73
0000001D0	20	43	74	72	6C	2B	41	6C	74	2B	44	65	6C	20	74	6F
0000001E0	20	72	65	73	74	61	72	74	0D	0A	00	00	00	00	00	00
0000001F0	00	00	00	00	00	00	00	00	83	A0	B3	C9	00	00	55	AA

字节偏移量	字段长度	字段定义
0x00	3 bytes	跳转指令
0x03	LONGLONG	OEM ID
0x0B	25 bytes	BPB
0x24	48 bytes	扩展 BPB
0x54	426 bytes	引导程序代码
0x01FE	WORD	结束标记（0xAA55）

MFT(Master File Table):组织结构示意表

最开始是保存系统关键信息的16个元数据文件。

从第24个记录开始，MFT记录的都是文件或者目录（其实被NTFS同样视为文件）的描述信息

0	$MFT
1	$MFTMirr
2	$LogFile
3	$Volume
4	$AttrDef
5	$Root
6	$Bitmap
7	$Boot
	…… ……
15	$Extend\$ObjID
16-23	为扩展保留
23以上（用户文件和目录）	File Record 1 （小文件，直接存放在MFT中）
	File Record 2 （大文件，另外开辟空间）
	……
	File Record n

MFT(Master File Table):主控文件表

- 主控文件表中的每个文件记录由两部分组成：
  - 表头（文件记录头）
    - 长度和偏移处的数据含义不变
  - 属性列表
    - 属性是File具体信息的载体，一个File的所有信息（包括文件的内容）都通过属性体现。
    - 不同的属性列表的对应偏移对应着不同的含义
- MFT中每个文件记录的结束标记为FFFFFFFFH

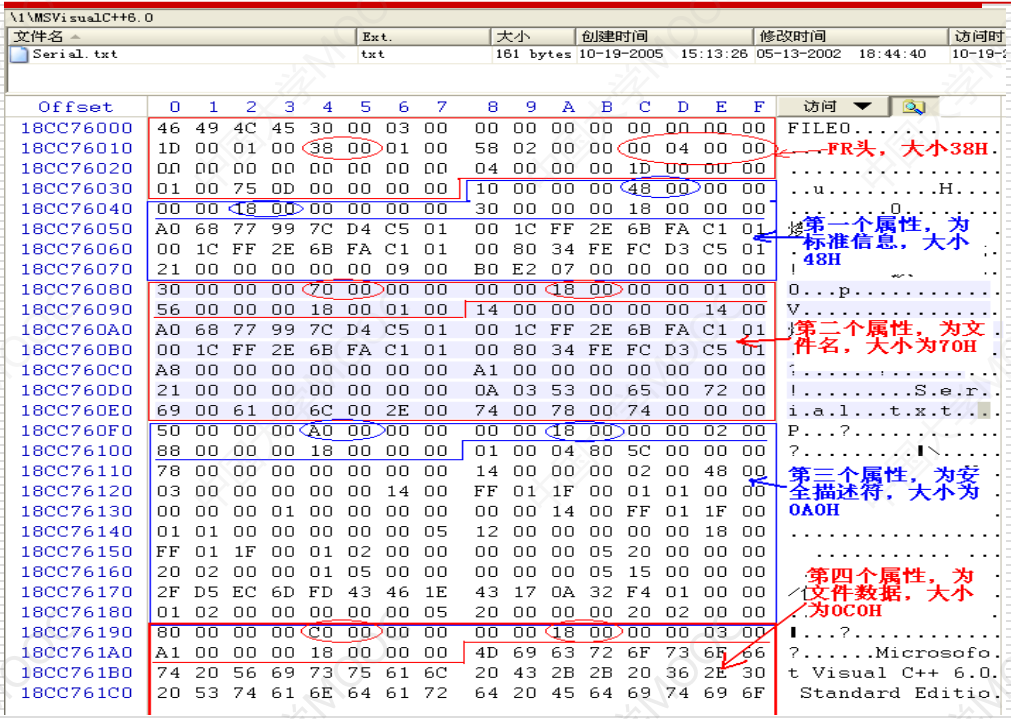
File Record(FR)

- File Record（文件记录，以下简称FR），大小保持为1KB，即2个扇区
  - 如果一个File足够小（大概700多字节以下）：
    - NTFS将其数据直接存放在该File的FR中；
  - 否则：
    - NTFS将开辟新空间存放数据，存放位置记录在FR中，通过Data Run指明每段起始簇号和每段（即碎片）占用的簇的个数。

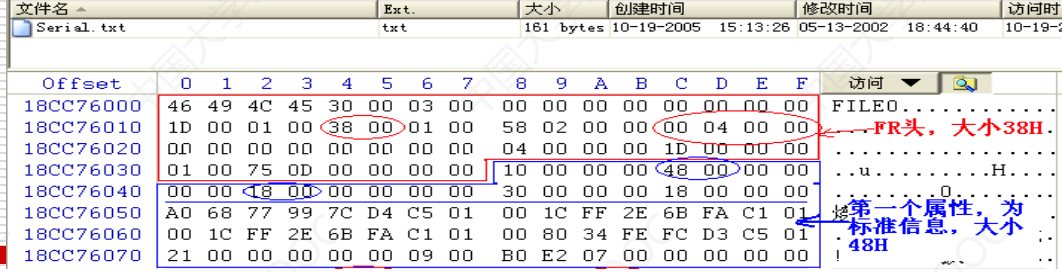
File Record组织结构示意表

FR头
属性1，通常是$STANDARD_INFORMATION
属性2，通常是$FILE_NAME
属性3，通常是$DATA（普通的数据文件），或者 $INDEX_ROOT
其它属性，比如：$INDEX_ALLOCATION
结束标志0xFF FF FF FF

实例：Serial.txt文件



FR头



偏移量0x00－0x03，标志“FILE”，每个FR头都以它开始

偏移量0x14处，2个字节，第一个属性的偏移位置，实际意义相当于FR头的长度，用来推算其后属性（Attribute）参数的位置

偏移量0x16处，2个字节，标志位，该FR是文件01/目录03/未使用00

偏移量0x18处，4个字节，FR实际占用的字节数

偏移量0x1C处，4个字节，总共分配给记录的长度

偏移量0x2C处，4个字节，MFT记录号，每个卷上的每个File都有一个唯一的记录号（在Windows XP下有效）

属性类型

常驻：属性内容全部在存储在MFT中，非常驻：属性内容在MTF之外存储

- 类型标志：偏移00H-03H。
  - 10 00 00 00H-00 10 00 00H
    - 如，30 00 00 00H表示该属性为文件名。
    - 80 00 00 00H表示该属性为文件数据。
- 按照有无属性名，是否常驻还可以分为四类：
  - 常驻、没有属性名
  - 常驻、有属性名
  - 非常驻、没有属性名
  - 非常驻、有属性名
- 每类属性的头部具体偏移含义有所不同。

属性类型说明

10 $STANDARD_INFORMATION （标准信息）
20 $ATTRIBUTE_LIST （属性列表）
30 $FILE_NAME （文件名）
40 $VOLUME_VERSION （卷版本）
50 $SECURITY_DEscriptOR （安全描述符）
60 $VOLUME_NAME （卷名）
70 $VOLUME_INFORMATION （卷信息）
80 $DATA （数据）
90 $INDEX_ROOT （索引根）
A0 $INDEX_ALLOCATION （索引分配）
B0 $BITMAP （位图）
C0 $SYMBOLIC_LINK（符号链接）
D0 $EA_INFORMATION （？信息）
E0 $EA

常驻属性与非常驻属性结构

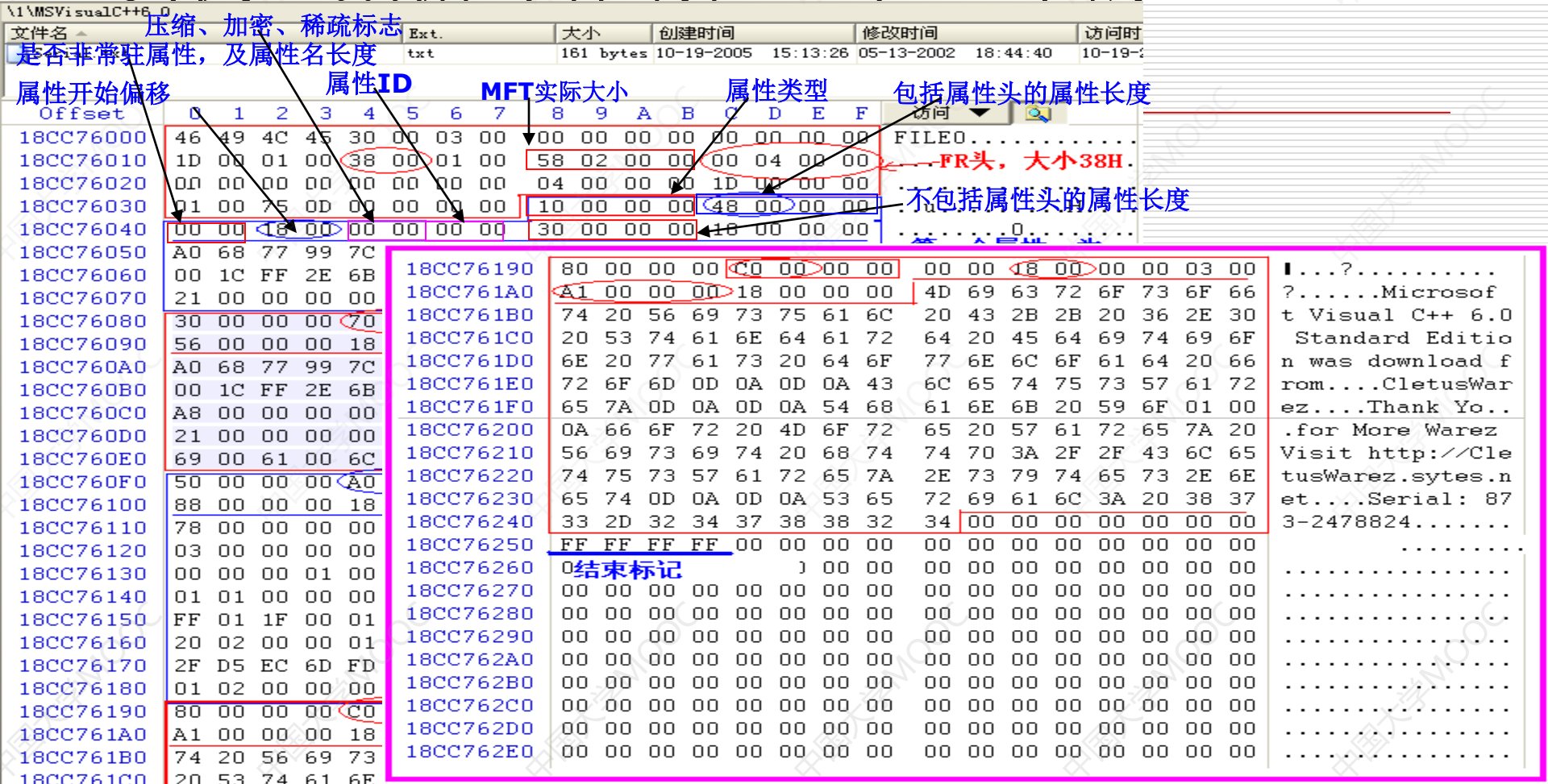
偏移字节（16进制）	常驻属性描述
00-03	属性类型
04-07	属性长度
08	常驻属性标志00:常驻；01表示非常驻
09	属性名长度(为0表示没有属性名)
0A-0B	属性名偏移(相对于属性头)
0C-0D	标志
0E-0F	属性ID标志
10-13	属性体大小
14-15	属性头的大小
16	索引
17	保留

常驻属性结构

偏移字节（16进制）	非常驻属性描述
00-03	属性类型
04-07	属性长度
08	常驻属性标志00:常驻；01表示非常驻
09	属性名长度(为0表示没有属性名)
0A-0B	属性名偏移(相对于属性头)
0C-0D	标志
0E-0F	属性ID标志
10-17	簇流的起始虚拟簇号(总是从0开始)
18-1F	簇流的结束虚拟簇号
20-21	簇流列表相对本属性头起始处偏移
22-23	压缩单位大小
24-27	保留
28-2F	为属性内容分配的空间大小字节数
30-37	属性内容实际占用的大小字节数
38-3F	属性内容初始大小字节数

非常驻属性结构

实例：数据可容纳在一个FR中的



数据无法容纳在一个FR中，怎么

偏移字节（16进制）	非常驻属性描述
00-03	属性类型
04-07	属性长度
08	常驻属性标志00:常驻；01表示非常驻
09	属性名长度(为0表示没有属性名)
0A-0B	属性名偏移(相对于属性头)
0C-0D	标志
0E-0F	属性ID标志
10-17	簇流的起始虚拟簇号(总是从0开始)
18-1F	簇流的结束虚拟簇号
20-21	簇流列表相对本属性头起始处偏移
22-23	压缩单位大小
24-27	保留
28-2F	为属性内容分配的空间大小字节数
30-37	属性内容实际占用的大小字节数
38-3F	属性内容初始大小字节数

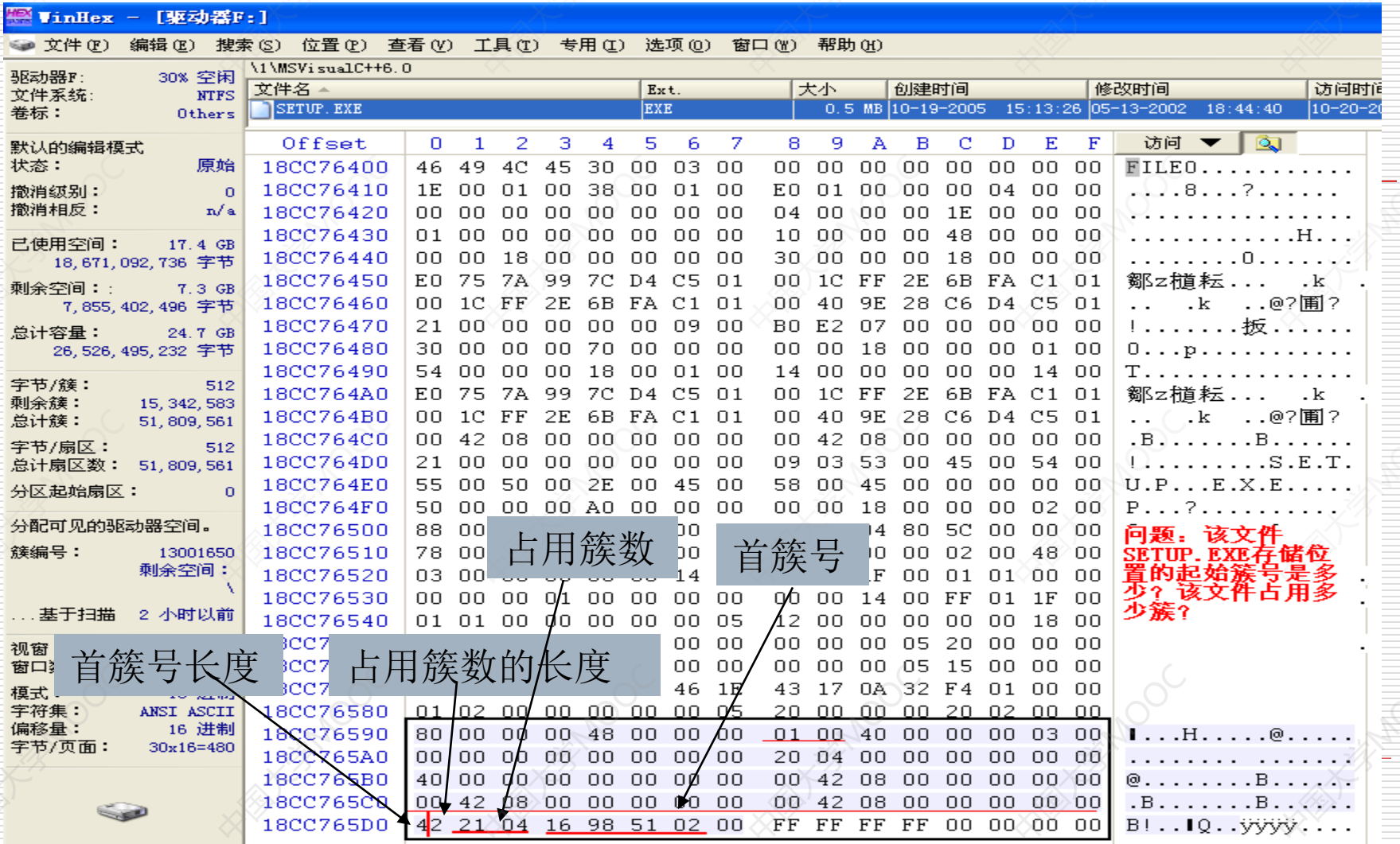
- **Data Run：指向数据存储位置。**
  - Data Run所在位置：属性的0x20 处
  - Data Run含义解读：
    - 可由多个子运行组成
      - 每个子运行第一个字节：分为前后两个部分，分别是“起始存储位置字段的字节数”和“长度字段的字节数”
      - 后续字节分别存储：长度和起始存储位置

```
18CC76590  80 00 00 00 48 00 00 00  01 00 40 00 00 00 03 00
18CC765A0  00 00 00 00 00 00 00 00  20 04 00 00 00 00 00 00
18CC765B0  40 00 00 00 00 00 00 00  00 42 08 00 00 00 00 00
18CC765C0  00 42 08 00 00 00 00 00  00 42 08 00 00 00 00 00
18CC765D0  42 21 04 16 98 51 02 00  FF FF FF FF 00 00 00 00
```

后续子运行说明

- 后续子运行的起始簇号：是相对于前一子运行的开始位置的偏移。
- 整个Data Run以00结束。

数据无法容纳在一个FR中的例子：Setup.exe


WinHex – [驱动器F:]
文件(F) 编辑(E) 搜索(S) 位置(P) 查看(V) 工具(T) 专用(I) 选项(O) 窗口(W) 帮助(H)
驱动器F: 30% 空闲
文件系统: NTFS
卷标: Others
默认的编辑模式
状态: 原始
撤消级别: 0
撤消相反: n/a
已使用空间: 17.4 GB
18,671,092,736 字节
剩余空间: 7.3 GB
7,855,402,496 字节
总计容量: 24.7 GB
26,526,495,232 字节
字节/簇: 512
剩余簇: 15,342,583
总计簇: 51,809,561
字节/扇区: 512
总计扇区数: 51,809,561
分区起始扇区: 0
分配可见的驱动器空间。
簇编号: 13001650
剩余空间:
基于扫描 2 小时以前
模式:
字符集: ANSI ASCII
偏移量: 16 进制
字节/页面: 30x16=480
\1\MSVisualC++6.0
文件名 Ext. 大小 创建时间 修改时间 访问时间
SETUP.EXE EXE 0.5 MB 10-19-2005 15:13:26 05-13-2002 18:44:40 10-20-2
Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F 访问
18CC76400 46 49 4C 45 30 00 03 00 00 00 00 00 00 00 00 00 FILE0.....
18CC76410 1E 00 01 00 38 00 01 00 E0 01 00 00 00 04 00 00 ....8...?.....
18CC76420 00 00 00 00 00 00 00 00 04 00 00 00 1E 00 00 00 ....
18CC76430 01 00 00 00 00 00 00 00 10 00 00 00 48 00 00 00 .....H...
18CC76440 00 00 18 00 00 00 00 00 30 00 00 00 18 00 00 00 .....0.....
18CC76450 E0 75 7A 99 7C D4 C5 01 00 1C FF 2E 6B FA C1 01 鄰z楦耘... .k .
18CC76460 00 1C FF 2E 6B FA C1 01 00 40 9E 28 C6 D4 C5 01 .. .k ..@?圃?
18CC76470 21 00 00 00 00 00 09 00 B0 E2 07 00 00 00 00 00 !.....扳.....
18CC76480 30 00 00 00 70 00 00 00 00 00 18 00 00 00 01 00 0...p.....
18CC76490 54 00 00 00 18 00 01 00 14 00 00 00 00 00 14 00 T.....
18CC764A0 E0 75 7A 99 7C D4 C5 01 00 1C FF 2E 6B FA C1 01 鄰z楦耘... .k .
18CC764B0 00 1C FF 2E 6B FA C1 01 00 40 9E 28 C6 D4 C5 01 .. .k ..@?圃?
18CC764C0 00 42 08 00 00 00 00 00 00 42 08 00 00 00 00 00 .B.....B.....
18CC764D0 21 00 00 00 00 00 00 00 09 03 53 00 45 00 54 00 !.....S.E.T.
18CC764E0 55 00 50 00 2E 00 45 00 58 00 45 00 00 00 00 00 U.P...E.X.E.....
18CC764F0 50 00 00 00 A0 00 00 00 00 00 18 00 00 00 02 00 P...?.....
18CC76500 88 00 ... 4 80 5C 00 00 00
18CC76510 78 00 ... 0 00 02 00 48 00
18CC76520 03 00 ... F 00 01 01 00 00
18CC76530 00 00 00 01 00 00 00 00 00 00 14 00 FF 01 1F 00
18CC76540 01 01 00 00 00 00 00 05 12 00 00 00 00 00 18 00
18CC76580 01 02 00 00 00 00 00 05 20 00 00 00 20 02 00 00
18CC76590 80 00 00 00 48 00 00 00 01 00 40 00 00 00 03 00 ▮...H.....@.....
18CC765A0 00 00 00 00 00 00 00 00 20 04 00 00 00 00 00 00 ....
18CC765B0 40 00 00 00 00 00 00 00 00 42 08 00 00 00 00 00 @.....B.....
18CC765C0 00 42 08 00 00 00 00 00 00 42 08 00 00 00 00 00 .B.....B.....
18CC765D0 42 21 04 16 98 51 02 00 FF FF FF FF 00 00 00 00 B!..▮Q..ÿÿÿÿ....
问题：该文件SETUP.EXE存储位置的起始簇号是多少？该文件占用多少簇？
占用簇数
首簇号
首簇号长度
占用簇数的长度

多个子运行的例子

- 数据流的Data Run描述：21 20 ED 05 22 48 07 48 22 21 28 C8 DB
  - 第一个子运行：开始于簇5EDH的20H个簇（5EDH-60CH）
  - 第二个子运行：开始于簇2835H的748H个簇（2835H=5EDH+2248H）
  - 第三个子运行：开始于簇3FDH的28H个簇（3FDH=2835H+0DBC8H）