

Hex 文件格式及其校验算法

hex 文件记录格式如下：

:AABBBBCCDD.....DDEE

其中

: 表示行起始标志

AA: 表示本条记录中的数据字节数

BBBB: 本条记录中的数据在存储区中的起始地址

CC: 记录类型，含义如下：

'00' Data Record

'01' End of File Record

'02' Extended Segment Address Record

'03' Start Segment Address Record

'04' Extended Linear Address Record

'05' Start Linear Address Record

DD.....DD:表示数据字节

EE: 表示校验和，具体算法为 $0x100 - ((0xAA + 0xBB + 0xBB + 0xCC + 0xDD + \dots + 0xDD) \% 256)$

示例：

:020000040000FA，我把它看做 0x02 0x00 0x00 0x04 0x00 0x00 0xFA

其中：

0x02: 表示本记录有效数据长度为 0x02

0x0000: 表示本条记录在

0x04: 表示本条记录为 Extended Linear Address Record

接下来的两个字节表示有效数据

0xFA: 表示校验和 $0xFA = 0x100 - (0x02 + 0x00 + 0x00 + 0x04 + 0x00 + 0x00)$

最后一行为文件结束标志：

:00000001FF

Hex 实际占用存储区大小计算办法

Hex 实际大小=文件结束标志前面一条记录的存储起始地址+这条记录的有效数据长度

相关校验测试程序:

下面算法实现记录的校验, 校验成功返回 1, 失败返回 0

```
1. bit AnalyseHEX(char hex[],int len)

2. {
3.     unsigned char i=1;
4.     unsigned char data;
5.     int cc=0;
6.     char temp[2];
7.     do
8.     {
9.         temp[0] = hex[i++];
10.        temp[1] = hex[i++];
11.        sscanf(temp,"%x",&data);
12.        cc += data;
13.    } while (i<(len-2));
14.    cc%=256;
15.    cc=0x100-cc;
16.    temp[0] = hex[i++];
17.    temp[1] = hex[i++];
18.    sscanf(temp,"%x",&data);
19.    return (cc==data)?1:0;
20. }

21.
22. int _tmain(int argc, _TCHAR* argv[])
23. {
24.     char hex[]="::020000040000FA";
25.     AnalyseHEX(hex,sizeof(hex));
26. }
```