# 用 C 语言读写 SGY 格式的地震数据

**Yangwqcumt**



地震勘探野外采集的数据，以及经过资料处理获得的三维数据体，只要是放在计算机里，都是以二进制文件的形式存放的。这些文件的处理显示等工作，一般都可以用商业化的软件进行。但是作为一个从事地震勘探研究的技术人员，有时会有些想法，有某种灵感，但是原有的软件又不允许你去做某种试验以验证你的想法。这时候，自编个小程序显然有必要，而且弄好后你的成就感会很强烈。

## 1. SEGY 格式地震数据文件

地震数据，是以各种格式存放的。所谓格式，指的是地震数据以及各种信息在文件内部的存放方式及顺序。

常见的地震数据格式，有 segy 格式、seg2 格式、segd 格式等。同样的格式，还有微机版、工作站版及其它版本。

本文仅是入门级材料，我们仅就微机版 segy 格式进行分析。

Segy 格式的地震数据文件，属于典型的流式文件，它的信息和数据都是按字节顺序一个个地存放的，每个字节都有其特定的含义。

这种格式的文件，由文件头部的 3600 字节以及地震道组成。

文件头前部的 3200 字节共分为 40 行，每行 80 个字符，但这些字符不是 ascii 码，是一种称为 ebcdic 的编码。一般这部分都不去读，或者只能显示出来查看其中的内容。

接下来是 400 字节的二进制部分。这里面有长整型数和短整型数，其具体含义参见附录一。

每个地震道由道头 240 字节（参见附录一）和地震数据组成。地震数据的个数和类型(指它是浮点数整数还是什么)文件头中有定义。此处我们假定所有的数据都是微机的四字节浮点数。

**2.** 文件头部 3200 字节特殊编码部分的读取。

该部分共分为40行，每行80个字符，但这些字符不是ascii码，是一种称为ebcdic的编码。转换成ascii码可以采用查表的方法进行。一般处理地震数据不用读这部分的内容。

```
#include "stdio.h"
#include "stdlib.h"
void main()
{
unsigned char E2A[256]={
0, 1, 2, 3,156, 9,134,127,151,141,142, 11, 12, 13, 14, 15, 16, 17, 18, 19,157,133,
8,135, 24, 25,146,143, 28, 29, 30, 31, 128,129,130,131,132, 10, 23,
27,136,137,138,139,140, 5, 6, 7, 144,145, 22,147,148,149,150, 4,152,153,154,155,
20, 21,158, 26, 32,160,161,162,163,164,165,166,167,168, 91, 46, 60, 40, 43, 33,
38,169,170,171,172,173,174,175,176,177, 93, 36, 42, 41, 59, 94, 45,
47,178,179,180,181,182,183,184,185,124, 44, 37, 95, 62, 63,
186,187,188,189,190,191,192,193,194, 96, 58, 35, 64, 39, 61, 34, 195, 97, 98,
99,100,101,102,103,104,105,196,197,198,199,200,201,
202,106,107,108,109,110,111,112,113,114,203,204,205,206,207,
208,209,126,115,116,117,118,119,120,121,122,210,211,212,213,214,215,
216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231, 123, 65, 66, 67,
68, 69, 70,71, 72, 73,232,233,234,235,236,237, 125, 74, 75, 76,
77,78,79,80,81,82,238,239,240,241,242,243, 92,159, 83, 84, 85, 86, 87, 88, 89,
90,244,245,246,247,248,249, 48,49, 50, 51, 52, 53, 54, 55, 56,
57,250,251,252,253,254,255};
   int i,j;
   unsigned char f3200[3200];    FILE *f1; char s1[200];
   printf("Input Sgy File Name:\n"); scanf("%s",s1); //输入文件名
   f1=fopen(s1,"rb");  fread(f3200,3200,1,f1);
   fclose(f1);
   for(i=0;i<40;i++)
   {  for(j=0;j<80;j++) printf("%c",E2A[f3200[i*80+j]]);
      printf("\n");
   }
}
```

## 3. 文件头 400 字节二进制部分的读取

## 可以把 400 字节作为若干个长整型及短整型数据读入：

```
#include "stdio.h"
#include "stdlib.h"
void main()
{
    FILE *f1;
    int i,l;  int traces;
    unsigned char f3200[3200];
    char FileName[200];
    long int s1[100];
    short int s2[200];
    printf("输入地震文件名[*.sgy]:"); scanf("%s",FileName);
    f1=fopen(FileName,"rb");
    if(f1==NULL) //文件打开不成就显示错误信息，然后退出。
    {   printf("File Open error!\n");  exit(0);  }

    fread(f3200,3200,1,f1); //读入前面的字节
    fread(s1,400,1,f1);     //读入接着的字节，作为长整型数
    fseek(f1,3200,0);       //退回去，
    fread(s2,400,1,f1);     //再以短整型数读入那些个字节

////////////////////////////////
    for(i=0;i<3;i++) printf("[%d]:%d \n",i+1,s1[i]);
    //显示文件头格式说明中的二进制部分的前三项
    for(i=3;i<15;i++)  //显示4-1,4-2，。。。，15-1,15-2
    {
        printf("[%d-1]:%d \n",i+1,s2[2*i]);
        printf("[%d-2]:%d \n",i+1,s2[2*i+1]);
    }
////////////////////////////////
    /*
    在文件头中，最重要的是下面几项：
    5-1:这个文件的地震道的时间采样间隔，单位是微秒；
    6-1：每个地震道的样点数；
    7-1:数据的格式。现在一般都是四字节的浮点数，格式取.
    */
    printf("时间采样间隔[微秒]：%d\n",s2[8]);
    printf("地震道的样点数：%d\n",s2[10]);
    printf("数据格式代码：%d\n",s2[12]);
    /*
    有了这些信息，这个文件含有的地震道数是多少呢？
```

```
    若文件长度：l
     则道数：traces=(l-3600)/(240+s2[10]*4)
     */
     fseek(f1,0,2);  //
     l=ftell(f1);    //l此时就是文件的字节数；
     printf("文件的长度=%d\n",l);
     traces=(l-3600)/(240+4*s2[10]);
     printf("地震道数是：%d\n",traces);
    fclose(f1);
}
```

**也可以把那个 400 字节作为一个结构体，该结构体定义见附录二。**

```
//这个例子使用了结构体来读取文件头信息
#include "stdio.h"
#include "stdlib.h"
#include "segyHeader.h"
void main()
{
     FILE *f1;
     int i,l;
     int traces,trace_length;
     char FileName[]="100.sgy";
     struct SegyReelHdrStruct  FileHeader;
     f1=fopen(FileName,"rb");
     if(f1==NULL) //文件打开不成就显示错误信息，然后退出。
     {
          printf("File Open error!\n");
          exit(0);
     }
     fread(&FileHeader,sizeof(FileHeader),1,f1);
     trace_length=FileHeader.hns;
     printf("%s:%d\n","时间采样间隔[微秒]",FileHeader.hdt);
     printf("%s:%d\n","每道的样点数",FileHeader.hns);
     printf("%s:%d\n","数据的格式代号",FileHeader.format);

     fseek(f1,0,2);  //
     l=ftell(f1);    //l此时就是文件的字节数；
     printf("文件的长度[字节数]=%d\n",l);
     traces=(l-3600)/(240+4*trace_length);
     fclose(f1);
     printf("%s:%d\n","地震道数",traces);
}
```

## 4. 某道数据的读取

```
include "stdio.h"          //包含头文件
#include "string.h"         //
#include "stdlib.h"         //
void main()
{
    FILE *f1,*f2;           //定义两个文件指针变量
    int i,l;
    char FileName[200]; //字符数组，存放文件名
    int Traces,Trace_length,Trace2read;
  //顾名思义，这几个变量用来存放:地震道数、地震道的长度、要读的那一个地震道的序号
    float *TraceData;      //定义一个浮点型的指针，一会儿开辟内存后放一个地震道的数据
   ////////////////////////获得地震那个文件名///////////
    printf("输入地震文件名[*.sgy]:");
    scanf("%s",FileName);
    //当然上面两行，可用一个这样的语句代替：strcpy(FileName,"100.sgy");
    f1=fopen(FileName,"rb");  //打开文件，打开形式为：二进制读 rb
    if(f1==NULL)   //判断打开了没有，不成功就返回吧。
    {
        printf("Cannot open input file!\n"); //显示信息
        exit(0);  //退出
    }
    Trace2read=430; // 要读取哪一道，可以键盘输入，为方便在这就给定了，设为第430道
    Trace_length=800; //一个地震道里面 数据的个数,可以从文件头获得，这里先拿来用了
    Traces=631;        //该文件地震道的个数，也可以通过文件头里面的信息设法获得，也是先拿来用
    l=3600L+(240+Trace_length*4L)*(Trace2read)+240;
    //要读取的那一个地震道的，数据部分，在这个文件中的开始的位置，这个很重要，下面做点说明
    /*
     一般来说，地震勘探的数据文件是有格式的。所谓格式，就是地震数据、及相关的信息在地震文件中的存放顺序。
      现在研究的这个叫做sgy格式。这种格式比较普遍。它由文件头和地震道组成，文件头字节，每个地震道由字节的道头和数据组成。
      所以一个sgy格式的文件的组成为：3600字节+240字节+第一道数据+240字节+第二道数据+240字节+第三道数据+。。。。。。。
      文件头部的字节以及每道的字节道头有信息，这个先不研究。

    对于我们给的这个数据，道数据的长度800，是指个浮点数，所以其字节数要乘以,得到，加上字节的道头，共计字节，
     所以第道的开始位置，就是：+430*3440，然后由于我们不读这个地震道的道头，数据部分的位置还要加，
     l=3600+3440*430+240;
    请时刻注意，c的数组以及其他什么东西的编号多是从0开始的。
    */
    fseek(f1,l,0);  //定位到那个地震道的 数据的 开始位置
    TraceData=new float[Trace_length];
```

```
   //先开一块内存，要不数据没地方放，你看事先定好可以吗，比如：float TraceData[800];
    fread(TraceData,4L*Trace_length,1,f1);
    /*
      读数据了，嘿嘿。这个fread函数，括弧内有四个东西：
      第一个，读了数据放在哪，可以是数组名或者指针
      第二个：一次读入多少字节
      第三个：读多少次，
      第四个，文件指针
      看起来，里面的第二个和第三个，暂时叫参数吧，是可以互换的，
      但是这个,我猜测它的读取的速度不同，没有验证过。
     与他对应的有个二进制写的函数,叫做fwrite();里面也有四个参数,四个参数的意义也和fread一样,只是那个文件指针是要写出的文件的指针
   */
    fclose(f1);
//记得用完文件要关闭，有的时候，没有关闭动作数据会被破坏，是教材上这么说的，我也都是这么做的
    f2=fopen("aline.txt","w");
 //打开一个文件文件，用来输出，一会你运行完了看看aline.txt里面的内容
    for(i=0;i<Trace_length;i++)
    {
        fprintf(f2,"%f \n",TraceData[i]);  //写到哪里了这是？
        printf("%f \n",TraceData[i]);     //这个又是写到哪里了？
    }
    fclose(f2);  //不管他，把文件关闭了
    delete TraceData;
//记得方才的开辟内存，现在用完了，该释放了。new 和delete 是一对，原则上必须成双。
//可能你不去delete也成，但是，你还是delete了吧
}
```

## 5. 一个完整的例子

这个例子，我们读入一个sgy文件的数据，每道数据给它除以该道的绝对值最大值，然后输出（这样，该地震道的数据映射到[-1,1]间了），文件头和每一个地震道的道头原样写出，只是数据换成处理后的。

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "math.h"
void main()
{
    FILE *filein,*fileout;        //输入文件及输出文件两个文件指针
    char inputfile[200],outputfile[200]; //输入文件名和输出文件名
    int i,j,l;  //定义一些整数，循环的时候用
    int traces,trace_length,samp_interval;
//道数、道长度（即每道的时间采样点数），时间采样间隔
    unsigned char f3200[3200]; //文件头部开始的字节
```

```
    short int f400[200];        //文件头的字节二进制部分

    float  xmax;  //用来存放某一个地震道的绝对值最大值
    char Trace_header[240];
    float  *tracedata;//浮点型的指针，用来开辟内存，放置地震道的数据

////////////////////////////////////////////////////////////
    printf("输入文件名[*.sgy]:\n");
    scanf("%s",inputfile);
    printf("输出文件名[*.sgy]:\n");
    scanf("%s",outputfile);
////////////////////////////////////////////////////////////
    filein=fopen(inputfile,"rb");
    if(filein==NULL)
    {
        printf("Can not open File %s\n",inputfile);
        exit(0);
    }
////////////////////////////////////////////////////////////
    fileout=fopen(outputfile,"wb");
    if(filein==NULL)
    {
        printf("Can not open File %s \n",outputfile);
        fclose(filein);
        exit(0);
    }

    fread(f3200,3200,1,filein);  //读入3200字节文件头部分
    fread(f400,400,1,filein);    //读入400字节文件头部分

    samp_interval=f400[8];
    trace_length=f400[11];
    fseek(filein,0,2);
    l=ftell(filein);
    traces=(l-3600)/(240+4L*trace_length);
    fseek(filein,3600,0);
    fwrite(f3200,3200,1,fileout);//写出3200字节
    fwrite(f400,400,1,fileout);//写出400字节

    tracedata=new float[trace_length];
    for(i=0;i<traces;i++)
    {
        fread(Trace_header,240,1,filein); //读入道头
        fread(tracedata,4L*trace_length,1,filein); //读入地震道数据
```

```
            xmax=fabs(tracedata[0]);
            for(j=1;j<trace_length;j++)  //循环取得最大的绝对值
                if(xmax<fabs(tracedata[j])) xmax=fabs(tracedata[j]);
           if(xmax>0)  //如果非0，就对每个数据除以该最大
            {
                for(j=0;j<trace_length;j++)
                    tracedata[j]/=xmax;
            }

            fwrite(Trace_header,240,1,fileout); //原样输出道头
            fwrite(tracedata,4L*trace_length,1,fileout); //输出那个变化的地震数据
        }
        fcloseall();  //关闭文件
        delete tracedata; //释放内存
        printf("File process finished.\n"); //告诉你活干完了。
}
```

## 6.后记

你看，刚才的地震处理程序完成了对单道地震数据的一种处理，程序很长。但是程序的主要工作是在进行数据的输入和输出， 大量的工作是核心内容之外的. 在地震数据的处理工作中，常常都是这样。所以，如果自己编程处理数据， 具备读写文件的能力，是非常必要的。当然，实际的地震资料数据处理非常复杂，使用的也都是商业化的大型软件，处理员不用编程。如果商业软件的某项处理功能不好，或者根本就不具备，而你自己有某种灵感想实现，那就要自己编程序了。这种灵感，是进行科学研究需要的。

最后跟大家说，地震勘探中各种文件的格式，在 seg 网站上都能找到相应的文档。

http://www.seg.org/resources/publications/misc/technical-standards

# 附录一 SEGY 格式地震数据文件头及道头说明

（这个是老标准的格式说明，新标有改变）

SEG-Y 格式 400 字节二进制文件头信息

| 字(32 位) | 字节号 | 说　　明 |
|---|---|---|
| 1 | 3201—3204 | 作业标识号。 |
| 2 | 3205-3208* | 测线号(每卷仅—条线) |
| 3 | 3209-3212* | 卷号。 |
| 4-1 | 3213-3214* | 每个记录的道数(包括 DUMMY 道和插入到记录或者共深度点的零记录道)。 |
| 4-2 | 3215-3216 | 每个记录的辅助道数(包括扫描道、时断、增益、同步和其他所有非地震数据道) |
| 5-1 | 3917-3918 | 这一卷带的采样间隔，以微秒表示。 |
| 5-2 | 3219-3220 | 野外记录的采样间隔，以微秒表示。 |
| 6-1 | 3221-3222 | 本卷数据的每个数据道的样点数。 |
| 6-2 | 3223-3224 | 野外记录的各数据道的样点数。 |
| 7-1 | 3225-3226* | 数据采样格式码：<br>1=浮点(4 字节)；　　2=定点(4 字节)；<br>3=定点(2 字节)；　　4=定点(W/增益 4 字节)<br>辅助道的每个采样使用相同的字节数。 |
| 7-2 | 3227-3228* | CMP 覆盖次数(每个 CMP 道集所希望的数据道数)。 |
| 8-1 | 3229-3230 | 道分选码：<br>1=同记录(没有分选)；　2=CMP 道集；<br>3=单次覆盖剖面：　　4=水平叠加剖面。 |
| 8-2 | 3231-3232 | 垂直叠加码<br>1=没有叠加：2=两次叠加；…<br>N=N 次相加(N=32，767) |
| 9-1 | 3233-3234 | 起始扫描频率 |
| 9-2 | 3235-3236 | 终止扫描频率。 |
| 10-1 | 3237-3238 | 扫描长度。以 ms 表示 |
| 10-2 | 3239-3240 | 扫描类型码：<br>1：线性扫描：2：抛物线扫描：<br>3：指数扫描；4：其他。 |
| 11-1 | 3241-3242 | 扫描通道的道号 |
| 11-2 | 3243-3244 | 有斜坡时，为起始斜坡长度(斜坡起始于时间零，使用时间为该长度)。以 ms 表示。 |
| 12-1 | 3245-3246 | 终了斜坡长度(终了斜坡起始于扫描长度；减终于斜坡长度)。以 ms 表示。 |
| 12-2 | 3247-3248 | 斜坡类型：　1=线性：2=$COS^2$　　3=其他 |
| 13-1 | 3249-3250 | 相关数据道　1=没有相关　　2=相关。 |
| 13-2 | 3251-3252 | 二进制增益恢复：1：恢复；2：没有恢复。 |

| | | |
|---|---|---|
| 14-1 | 3253-3254 | 振幅恢复方式：<br>1=没有； 2=球面扩散；<br>3=AGC： 4=其他 |
| 14-2 | 3255-3256* | 测量系统：1=米 2=英尺。 |
| 15-1 | 3257-3258 | 脉冲信号极性:<br>1=压力增加或者使检波器向上运动在磁带上记的是负数：<br>2=压力增加或者使检波器向上运动在磁带上记的是正数。 |
| 15-2 | 3259-3260 | 可控震源 地震信号滞<br>极性代码 后引导信号<br>1 =337.5 -22.5<br>2 =22.5 -67.5<br>3 =67.5 -112.5<br>4 =112.5 -157.5<br>5 =157.5 -202.5<br>6 =202.5 -247.5<br>7 =247.5 -292.5<br>8 =292.5 -337.5 |
| 3261-3600 | 没有确定，选择使用 | |

SEG—Y 格式道头说明

| 字（32 位） | 字节号 | 说 明 |
|---|---|---|
| 1 | 1-4* | 一条测线中的道顺序号。如果一条测线有若干卷带，顺序号连续递增。 |
| 2 | 5-8 | 在本卷磁带中的道顺序号。每卷带的道顺序号从 1 开始。 |
| 3 | 9-12 | 原始的野外记录号。 |
| 4 | 13-16 | 在原始野外记录中的道号。 |
| 5 | 17-20 | 震源点号(在同一个地面点有多于一个记录时使用)。 |
| 6 | 21-24 | CMP 号。 |
| 7 | 25-28 | 在 CMP 道集中的道号(在每个 CMP 道集中道号从 1 开始)。 |
| 8-1 | 29-30* | 道识别码：<br>l=地震数据 4：时断： 7=记录：<br>2=死道： 5：井口时间； 8=水断：<br>3=DUMMY； 6=扫描道； 9…N=选择使用(N=32767) |
| 8-2 | 31-32 | 产生这一道的垂直叠加道数(1 是一道： 2 是两道相加；…)。 |
| 9-1 | 33-34 | 产生这一道的水平替加道数(1 是一道： 2 是两道替加：…)。 |
| 9-2 | 35-36 | 数据类型：1：生产；2：试验。 |
| 10 | 37-40 | 从炮点到接收点的距离(如果是相反向激发为负值)。 |
| 11 | 41-44 | 接收点高程。高于海平而的高程为止，低于海平面为负。 |
| 12 | 45-48 | 炮点的地面高程。 |
| 13 | 49-52 | 炮点低于地面的深度(正数)。 |
| 14 | 53-56 | 接收点的基准面高程。 |
| 15 | 57-60 | 炮点的基准面高程。 |
| 16 | 61-64 | 炮点的水深。 |
| 17 | 65-68 | 接收点的水深。 |

| | | |
|---|---|---|
| 18-1 | 69-70 | 对 41-68 字节中的所有高程和深度应用了此因子给出真值。比例因子：1，±10，±100，±1000 或者±10000。如果为正，乘以因子： 如果为负，则除以因子。 |
| 18-2 | 71-72 | 对 73-88 字节中的所有坐标应用了此因子给出真值。比例因子：1，±10，±100，±1000 或者±10000。如果为正，乘以因子： 如果为负，则除以因子。 |
| 19 | 73-76 | 炮点坐标 X ｜如果坐标单位是弧度的秒，X 值代表。 |
| 20 | 77-80 | 炮点坐标 y ｜经度，Y 值代表纬度。正值代表格林 |
| 21 | 81-84 | 检波点 X ｜ 威治子午线东或者赤道北的秒数。负 |
| 22 | 85-38 | 检波点 Y ｜ 值则为西或者南的秒数 |
| 23-1 | 89-90 | 坐标单位； 1：长度(米或者英尺)； 2：弧度的秒。 |
| 23-2 | 91-92 | 风化层速度。 |
| 24-1 | 93-94 | 风化层下的速度。 |
| 24-2 | 95-96 | 震源处的井口时间。 |
| 25-1 | 97-98 | 接收点处的井口时间。 |
| 25-2 | 99-100 | 炮点的静校正。 |
| 26-1 | 101-102 | 接收点的静校正。 |
| 26-2 | 103—104 | 应用的总静校正量(如果没有应用静校正为零)。 |
| 27-1 | 105-106 | 延迟时间-A，以 ms 表示。240 字节的道标识的结束和时间信号之间的时间。如果时间信号山现在道头结束之前为正。如果时间信号出现在道头结束之后为负。时间信号就是起始脉冲，它记录在辅助道上或者由记录系统指定。 |
| 27-2 | 107-108 | 时间延迟—B，以 ms 表示。为时间信号和能量起爆之间的时间。可正可负。 |
| 28-1 | 109-110 | 时间延迟时间，以 ms 表示。能量源的起爆时间和开始记录数据样点之间的时间(深水时，数据记录不从时间零开始。) |
| 28-2 | 111-112 | 起始切除时间。 |
| 29-1 | 113-114 | 结束切除时间。 |
| 29-2 | 115-116 | 本道的采样点数。 |
| 30-1 | 117_118 | 本道的采样间隔，以 us 表示。 |
| 30-2 | 119-120 | 野外仪器的增益类别：1：固定增益；2：二进制增益： 3：浮点增益：4…N：选择使用。 |
| 31-1 | 121-122 | 仪器增益常数。 |
| 31-2 | 123—124 | 仪器起始增益(db)。 |
| 32-1 | 125-126 | 相关码：1 二没有相关：2：相关。 |
| 32-2 | 127—128 | 起始扫描频率。 |
| 33-1 | 129-130 | 结束扫描频率。 |
| 33-2 | 131—132 | 扫描长度，以 ms 表示。 |
| 34-1 | 133-134 | 扫描类型：1：线性；2；抛物线；3：指数：4：其他 |
| 34-2 | 135-136 | 扫描道起始斜坡长度，以 ms 表示。 |
| 35-1 | 137—138 | 扫描道终了斜坡长度，以 ms 表示。 |
| 35-2 | 139-140 | 斜坡为型：1：线性；2；$COS^2$：3；其他。 |
| 36-1 | 141-142 | 滤假频的频率(如果使用)。 |

| | | |
|---|---|---|
| 36-2 | 143—144 | 滤假频的陡度。 |
| 37-1 | 145-146 | 陷波陡率(如果使用)。 |
| 37-2 | 147-148 | 陷波陡度。 |
| 38-1 | 149-150 | 低截频率(如果使用;。 |
| 38-2 | 151—152 | 高截频率(如果使用)。 |
| 39-1 | 153—154 | 低截频率陡度。 |
| 39-2 | 155—156 | 高截频率陡度。 |
| 40-1 | 157-158 | 数据{己录的年。 |
| 40-2 | 159-160 | 日。 |
| 41-1 | 161-162 | 小时(24 时制) |
| 41-2 | 163—164 | 分。 |
| 42-1 | 165-166 | 秒。 |
| 42-2 | 167-168 | 时间代码： |
| | | 1：当地时间：2:格林威治时间：3：其他 |
| 43-1 | 169-170 | 道加权因子。 |
| | | (最小有效位定义为 2**(-N)，N=0，1，2，…，32767) |
| 43-2 | 171-172 | 覆盖开关位置 1 的检波器道号。 |
| 44-1 | 173-174 | 在原始野外记录中道号 1 的检波器号。 |
| 44-2 | 175-176 | 在原始野外记录中最后一道的检波器号。 |
| 45-1 | 177-178 | 缺口大小(滚动的总道数)。 |
| 45-2 | 179-180 | 在测线的开始或者结束处的斜坡位置： |
| | | 1：在后面；2：在前面。 |
| | 181-240 | 没有定义，可以选择使用? |

说明：
1. 带*的字节的信息必须记录。
2. 本说明仅供参考。

# 附录二 C 语言 400 字节文件头结构体及道头结构体

```
/* the SEG-Y reel identification header */
struct SegyReelHdrStruct
{
char comment[3200];
long  int jobid;  /* job identification number */
long  int lino; /* line number (only one line per reel) */
long  int reno; /* reel number */
short int ntrpr; /* number of data traces per record */
short int nart; /* number of auxiliary traces per record */
short int hdt; /* sample interval in micro secs for this reel */
short int dto; /* same for original field recording */
short int hns; /* number of samples per trace for this reel */
short int nso; /* number of samples per trace for original field recording */
short int format; /* data sample format code:
1 = floating point (4 bytes) 2 = fixed point (4 bytes) 3 = fixed point (2 bytes)
```

```
4 = fixed point w/gain code (4 bytes) */
short int fold; /* CDP fold expected per CDP ensemble */
short int tsort; /* trace sorting code: 1 = as recorded (no sorting) 2 = CDP ensemble
3 = single fold continuous profile 4 = horizontally stacked */
short int vscode; /* vertical sum code: 1 = no sum 2 = two sum ... N = N sum (N =
32,767) */
short int hsfs; /* sweep frequency at start */
short int hsfe; /* sweep frequency at end */
short int hslen; /* sweep length (ms) */
short int hstyp; /* sweep type code:
1 = linear 2 = parabolic 3 = exponential 4 = other */
short int schn; /* trace number of sweep channel */
short int hstas; /* sweep trace taper length (msec) at start if tapered (the taper
starts at zero time and is effective for this length) */
short int hstae; /* sweep trace taper length (msec) at end (the ending taper starts
at sweep length minus the taper length at end) */
short int htatyp; /* sweep trace taper type code: 1 = linear 2 = cos-squared 3 =
other */
short int hcorr; /* correlated data traces code:1 = no 2 = yes */
short int bgrcv; /* binary gain recovered code: 1 = yes 2 = no */
short int rcvm; /* amplitude recovery method code: 1 = none 2 = spherical divergence
3 = AGC 4 = other */
short int mfeet; /* measurement system code: 1 = meters 2 = feet */
short int polyt; /* impulse signal polarity code:
1 = increase in pressure or upward geophone case movement gives negative number on
tape
2 = increase in pressure or upward geophone case movement gives positive number on
tape */
short int vpol; /* vibratory polarity code: code seismic signal lags pilot by
337.5 to 22.5 degrees 22.5 to 67.5 degrees 67.5 to 112.5 degrees
112.5 to 157.5 degrees 157.5 to 202.5 degrees 202.5 to 247.5 degrees 247.5 to 292.5
degrees 293.5 to 337.5 degrees */
short int hunass[170];  /* unassigned */ } ;  /*  3600 bytes if tightly  packed  */


/* the SEG-Y trace identification header */
struct  SegyTraceHdrStruct
 { long int tracl; /* trace sequence number within line */
long int tracr; /* trace sequence number within reel */
long int fldr; /* field record number */
long int tracf; /* trace number within field record */
long int ep; /* energy source point number */
long int cdp; /* CDP ensemble number */
long int cdpt; /* trace number within CDP ensemble */
```

short int trid; /* trace identification code:
1 = seismic data 2 = dead 3 = dummy 4 = time break 5 = uphole 6 = sweep 7 = timing 8 = water break 9---, N = optional use (N = 32,767) Following are CWP id flags: 9 = autocorrelation 10 = Fourier transformed - no packing
xr[0],xi[0], ..., xr[N-1],xi[N-1]
11 = Fourier transformed - unpacked Nyquist xr[0],xi[0],...,xr[N/2],xi[N/2] 12 = Fourier transformed - packed Nyquist
even N:
xr[0],xr[N/2],xr[1],xi[1], ..., xr[N/2 -1],xi[N/2 -1] (note the exceptional second entry)
odd N:
xr[0],xr[(N-1)/2],xr[1],xi[1], ..., xr[(N-1)/2 -1],xi[(N-1)/2 -1], xi[(N-1)/2] (note the exceptional second & last entries) 13 = Complex signal in the time domain xr[0],xi[0], ..., xr[N-1],xi[N-1] 14 = Fourier transformed -amplitude/phase a[0],p[0], ..., a[N-1],p[N-1] 15 = Complex time signal - amplitude/phase a[0],p[0], ..., a[N-1],p[N-1] 16 = Real part of complex trace from 0 to Nyquist 17 = Imag part of complex trace from 0 to Nyquist 18 = Amplitude of complex trace from 0 to Nyquist
19 = Phase of complex trace from 0 to Nyquist 21 = Wavenumber time domain (k-t) 22 = Wavenumber frequency (k-omega) 30 = Depth-Range (z-x) traces
101 = Seismic data packed to bytes (by supack1) 102 = Seismic data packed to 2 bytes (by supack2) 200 = GPR data
*/
short int nvs; /* number of vertically summed traces (see vscode in reel header structure) */
short int nhs; /* number of horizontally summed traces (see vscode in reel header structure) */
short int duse; /* data use: 1 = production 2 = test */
long int offset; /* distance from source point to receiver group (negative if opposite to direction in which the line was shot) */
long int gelev; /* receiver group elevation from sea level (above sea level is positive) */
long int selev; /* source elevation from sea level (above sea level is positive) */
long int sdepth; /* source depth below surface (positive) */
long int gdel; /* datum elevation at receiver group */
long int sdel; /* datum elevation at source */
long int swdep; /* water depth at source */
long int gwdep; /* water depth at receiver group */
short int scalel; /* scale factor for previous 7 entries
with value plus or minus 10 to the power 0, 1, 2, 3, or 4 (if positive, multiply, if negative divide) */
short int scalco; /* scale factor for next 4 entries with value plus or minus 10 to the power 0, 1, 2, 3, or 4 (if positive, multiply, if negative divide) */

```
long int sx; /* X source coordinate */
long int sy; /* Y source coordinate */
long int gx; /* X group coordinate */
long int gy; /* Y group coordinate */
short int counit; /* coordinate units code:
for previous four entries 1 = length (meters or feet) 2 = seconds of arc (in this case, the X values are longitude and the Y values are latitude, a positive value designates the number of seconds east of Greenwich or north of the equator */
short int wevel; /* weathering velocity */
short int swevel; /* subweathering velocity */
short int sut; /* uphole time at source */
short int gut; /* uphole time at receiver group */
short int sstat; /* source static correction */
short int gstat; /* group static correction */
short int tstat; /* total static applied */
short int laga; /* lag time A, time in ms between end of 240-
  byte trace identification header and time break, positive if time break occurs after end of header, time break is defined as the initiation pulse which maybe recorded on an auxiliary trace or as otherwise specified by the recording system */
short int lagb; /* lag time B, time in ms between the time break and the initiation time of the energy source, may be positive or negative */
short int delrt; /* delay recording time, time in ms between initiation time of energy source and time when recording of data samples begins (for deep water work if recording does not start at zero time) */
short int muts; /* mute time--start */
short int mute; /* mute time--end */
unsigned short int ns; /* number of samples in this trace */
unsigned short int dt; /* sample interval; in micro-seconds */
short int gain; /* gain type of field instruments code: 1 = fixed 2 = binary 3 = floating point 4 ---- N = optional use */
short int igc; /* instrument gain constant */
short int igi; /* instrument early or initial gain */
short int corr; /* correlated: 1 = no 2 = yes */
short int sfs; /* sweep frequency at start */
short int sfe; /* sweep frequency at end */
short int slen; /* sweep length in ms */
short int styp; /* sweep type code: 1 = linear 2 = parabolic 3 = exponential 4 = other */
short int stas; /* sweep trace taper length at start in ms */
short int stae; /* sweep trace taper length at end in ms */
short int tatyp; /* taper type: 1=linear, 2=cos^2, 3=other */
short int afilf; /* alias filter frequency if used */
short int afils; /* alias filter slope */
```

```
short int nofilf; /* notch filter frequency if used */
short int nofils; /* notch filter slope */
short int lcf; /* low cut frequency if used */
short int hcf; /* high cut frequncy if used */
short int lcs; /* low cut slope */
short int hcs; /* high cut slope */
short int year; /* year data recorded */
short int day; /* day of year */
short int hour; /* hour of day (24 hour clock) */
short int minute; /* minute of hour */
short int sec; /* second of minute */
short int timbas; /* time basis code:1 = local 2 = GMT 3 = other */
short int trwf; /* trace weighting factor, defined as 1/2^N volts for the least
sigificant bit */
short int grnors; /* geophone group number of roll switch position one */
short int grnofr; /* geophone group number of trace one within original field record
*/
short int grnlof; /* geophone group number of last trace within original field record
*/
short int gaps; /* gap size (total number of groups dropped) */
short int otrav; /* overtravel taper code: 1 = down (or behind) 2 = up (or ahead)
*/
short int unass[30]; /* unassigned -- for optional info */
                                                    };
```