Dustin Paltz

4/28/2015

Home Work 4

Below is the hexadecimal contents of ‘mocities.shp’ generated by executing the octal dump command od –x on the shapefile. The first column of the dump is an octal counter which on each row displays the 0 based index of the first byte in that row, which is the first byte in the second column. The octal counts have been replaced with a decimal count for readability. There is a total of 268 bytes in this block.

The structure of a shapefile containing two records is as follows:

* File header (100 bytes)
* Record header (8 bytes)
* Record Contents
* Record Header
* Record Contents

Data types in shapefile:

* Integer: Signed 32-bit integer (4 bytes)
* Double: Signed 64-bit IEEE double-precision floating point number (8 bytes)

Byte order is Big Endian

**0 0000 270a 0000 0000 0000 0000 0000 0000**

Bytes 0-3: (Integer) File Code = 270A = 9994

Bytes 4-23: unused

**16 0000 0000 0000 0000 0000 0086 e803 0000**

Bytes 24-27: (Integer) File Length = 86 = 134

File length is measured in 16-bit words, where one 16-bit word is two bytes. Therefore 134 16-bit words equals 268 bytes.

Byte order is now Little Endian

Bytes 28-31: (Integer) Version = 3E8 = 1000

**32 0100 0000 14ae 47e1 7ab4 57c0 5c8f c2f5**

Bytes 32-35: (Integer) Shape Type = 1 = Point

All non-Null shapes in a shapefile are required to be of the same shape type.

Bytes 36-43: (Double) Bounding Box X minimum

Bytes 44-51: (Double) Bounding Box Y minimum

**48 289c 4240 52b8 1e85 eb61 56c0 3d0a d7a3**

Bytes 52-59: (Double) Bounding Box X maximum

Bytes 60-67: (Double) Bounding Box Y maximum

**64 70dd 4340 0000 0000 0000 0000 0000 0000**

Bytes 68-75: (Double) Bounding Box Z minimum

Bytes 76-83: (Double) Bounding Box Z maximum

**80 0000 0000 0000 0000 0000 0000 0000 0000**

Bytes 84-91: (Double) Bounding Box M minimum

Bytes 92-99: (Double) Bounding Box M maximum

**96 0000 0000 0000 0001 0000 000a 0100 0000**

Shapefile headers have a fixed length of 100 bytes and ends on the above line. Now begins the first record header. Byte indexes continue offset from the beginning of the record header.

Byte order is now Big Endian.

Bytes 0-3: (Integer) Record Number = 1

Bytes 4-7: (Integer) Content Length = A = 10

Record content lengths are measured in 16-bit words, where one 16-bit word is two bytes. Therefore 10 16-bit words equals 20 bytes. Record contents consist of the shape type and the geometric data for the shape. Byte indexes continue offset from the beginning of the record contents.

Byte order is now Little Endian.

Bytes 0-3: (Integer) Shape Type = 1 = Point

**112 3333 3333 3353 57c0 5c8f c2f5 289c 4240**

Bytes 4-11: (Double) X

Double precision number calculation of X

1. Convert 3333 3333 3353 57c0 to Big Endian  
   = c057 5333 3333 3333
2. Base 2:   
   1100000001010111010100110011001100110011001100110011001100110011
3. Bit 0: Sign = 1 = negative
4. Bits 1-11: Exponent = 1029 = 1029 – 1023   
   = 6 = 2^6 = 64
5. Bits 12-63: Fraction = 1 + 0.4578125000
6. X (Longitude) =   
   -1 \* 1.4578125 \* 64 = **-93.3**

Bytes 12-19: (Double) Y

Double precision number calculation of Y

1. Convert 5c8f c2f5 289c 4240 to Big Endian  
   = 4042 9c28 f5c2 8f5c
2. Base 2:  
   0100000001000010100111000010100011110101110000101000111101011100
3. Bit 0: Sign = 0 = positive
4. Bits 1-11: Exponent = 1028 = 1028 – 1023   
   = 5 = 2^5 = 32
5. Bits 12-63: Fraction = 1 + 0. 163125
6. Y (Latitude) =   
   1 \* 1. 163125 \* 32 = **37.22**

**128 0000 0002 0000 000a 0100 0000 cdcc cccc**

Start record header. Byte indexes continue offset from the beginning of the record header.

Byte order is now Big Endian.

Bytes 0-3: (Integer) Record Number = 2

Bytes 4-7: (Integer) Content Length = A = 10

Start record content. Byte indexes continue offset from the beginning of the record contents.

Byte order is now Little Endian.

Bytes 0-3: (Integer) Shape Type = 1 = Point

Bytes 4-11: (Double) X

**144 cc0c 57c0 cdcc cccc cc4c 4340 0000 0003**

Bytes 12-19: (Double) Y

Start record header. Byte indexes continue offset from the beginning of the record header.

Byte order is now Big Endian.

Bytes 0-3: (Integer) Record Number = 3

**160 0000 000a 0100 0000 cdcc cccc cc8c 56c0**

Bytes 4-7: (Integer) Content Length = A = 10

Start record content. Byte indexes continue offset from the beginning of the record contents.

Byte order is now Little Endian.

Bytes 0-3: (Integer) Shape Type = 1 = Point

Bytes 4-11: (Double) X

**176 713d 0ad7 a350 4340 0000 0004 0000 000a**

Bytes 12-19: (Double) Y

Start record header. Byte indexes continue offset from the beginning of the record header.

Byte order is now Big Endian.

Bytes 0-3: (Integer) Record Number = 4

Bytes 4-7: (Integer) Content Length = A = 10

**192 0100 0000 85eb 51b8 1ea5 57c0 0ad7 a370**

Start record content. Byte indexes continue offset from the beginning of the record contents.

Byte order is now Little Endian.

Bytes 0-3: (Integer) Shape Type = 1 = Point

Bytes 4-11: (Double) X

Bytes 12-19: (Double) Y

**208 3d8a 4340 0000 0005 0000 000a 0100 0000**

Start record header. Byte indexes continue offset from the beginning of the record header.

Byte order is now Big Endian.

Bytes 0-3: (Integer) Record Number = 5

Bytes 4-7: (Integer) Content Length = A = 10

Start record content. Byte indexes continue offset from the beginning of the record contents.

Byte order is now Little Endian.

Bytes 0-3: (Integer) Shape Type = 1 = Point

**224 14ae 47e1 7ab4 57c0 3d0a d7a3 70dd 4340**

Bytes 4-11: (Double) X

Bytes 12-19: (Double) Y

**240 0000 0006 0000 000a 0100 0000 52b8 1e85**

Start record header. Byte indexes continue offset from the beginning of the record header.

Byte order is now Big Endian.

Bytes 0-3: (Integer) Record Number = 6

Bytes 4-7: (Integer) Content Length = A = 10

Start record content. Byte indexes continue offset from the beginning of the record contents.

Byte order is now Little Endian.

Bytes 0-3: (Integer) Shape Type = 1 = Point

Bytes 4-11: (Double) X

**256 eb61 56c0 a470 3d0a d7a3 4240**

Bytes 12-19: (Double) Y

**268**

Below is the hexadecimal contents of ‘mocities.shx’ generated by executing the octal dump command od –x on the index file. The first column of the dump is an octal counter which on each row displays the 0 based index of the first byte in that row, which is the first byte in the second column. The octal counts have been replaced with a decimal count for readability. Index files contain a 100-byte header followed by 8-byte, fixed-length records. The index file header is identical in structure to the main file header.

Byte order is Big Endian

**0 0000 270a 0000 0000 0000 0000 0000 0000**

Bytes 0-3: (Integer) File Code = 270A = 9994

Bytes 4-23: unused

**16 0000 0000 0000 0000 0000 004a e803 0000**

Bytes 24-27: (Integer) File Length = 4A = 74

File length is measured in 16-bit words, where one 16-bit word is two bytes. Therefore 74 16-bit words equals 148 bytes.

Byte order is now Little Endian

Bytes 28-31: (Integer) Version = 3E8 = 1000

**32 0100 0000 14ae 47e1 7ab4 57c0 5c8f c2f5**

Bytes 32-35: (Integer) Shape Type = 1 = Point

Bytes 36-43: (Double) Bounding Box X minimum

Bytes 44-51: (Double) Bounding Box Y minimum

**48 289c 4240 52b8 1e85 eb61 56c0 3d0a d7a3**

Bytes 52-59: (Double) Bounding Box X maximum

Bytes 60-67: (Double) Bounding Box Y maximum

**64 70dd 4340 0000 0000 0000 0000 0000 0000**

Bytes 68-75: (Double) Bounding Box Z minimum

Bytes 76-83: (Double) Bounding Box Z maximum

**80 0000 0000 0000 0000 0000 0000 0000 0000**

Bytes 84-91: (Double) Bounding Box M minimum

Bytes 92-99: (Double) Bounding Box M maximum

**96 0000 0000 0000 0032 0000 000a 0000 0040**

The header ends on the above line. Now begins the first Index Record. The Nth record in the index file stores the offset and content length for the Nth record in the main file. Byte indexes continue offset from the beginning of each record.

Byte order is now Big Endian.

Record 1

Bytes 0-3: (Integer) Offset = 32

= 50 16-bit words

= 100 bytes

The offset is the number of 16-bit words from the start of the main file to the first byte of the record header for the record

Bytes 4-7: (Integer) Content Length = A = 10

The content length stored in the index record is the same as the value stored in the main file record header.

Record 2

Bytes 0-3: (Integer) Offset = 40

= 64 16-bit words

= 128 bytes

**112 0000 000a 0000 004e 0000 000a 0000 005c**

Bytes 4-7: (Integer) Content Length = A = 10

Record 3

Bytes 0-3: (Integer) Offset = 4E

= 78 16-bit words

= 156 bytes

Bytes 4-7: (Integer) Content Length = A = 10

Record 4

Bytes 0-3: (Integer) Offset = 5C

= 92 16-bit words

= 184 bytes

**128 0000 000a 0000 006a 0000 000a 0000 0078**

Bytes 4-7: (Integer) Content Length = A = 10

Record 5

Bytes 0-3: (Integer) Offset = 6A

= 106 16-bit words

= 212 bytes

Bytes 4-7: (Integer) Content Length = A = 10

Record 6

Bytes 0-3: (Integer) Offset = 78

= 120 16-bit words

= 240 bytes

**144 0000 000a**

Bytes 4-7: (Integer) Content Length = A = 10

**148**