Seismic Unix: endianness and suswapbytes

1. Overview

Sometimes I get a Seismic Unix (SU) data file that I did not create. In other words, I download or copy an SU data file from an external location. I might do this to process or to view the data.

The first program I use on any SU dataset is *surange*:

```
SURANGE - get max and min values for non-zero header entries
```

This is a simple program that prints useful information about the dataset to the screen (or to a file if I use Unix re-direct) such as number of traces, number of samples in each trace, and sample interval. I can also see which SU keys have values and whether those key values make sense. I like *surange* a lot because it usually runs quickly and it gives useful output.

I like *surange* for one more reason. If *surange* works, I know the dataset is in the correct format for me to continue using it. But, if I get something like the following error message:

```
surange: fgettr.c: on trace #2 number of samples in header (4325)
differs from number for first trace (13320)
```

I suspect the SU file has the wrong endianness. To solve an endianness problem, I use *suswapbytes*. Of course, after I use *suswapbytes*, I again use *surange* to test the dataset. If the output of *surange* is, again, an error message, I stop trying to use the SU dataset.

"Endianness" is categorized as either big endian or little endian (or even bi-endian). The discussion that follows does not explain endianness in any detail; I leave that to you and Wikipedia. The discussion below is intended to help you recognize an endianness problem and shows you how to solve it.

> I want to emphasize, this discussion is about examining a new (to me) dataset that is in Seismic Unix format, not in SEG-Y or SEG-D or any other non-SU format.

2. Get the example data – Oz Yilmaz shot gathers

In the mid-1980s, Dr. Oz Yilmaz did research and taught seismic data processing at Western Geophysical, at the time a division of Litton Industries, but originally founded in 1933 by Henry Salvatori. During his teaching, Dr. Yilmaz often used 40 shot gathers that had been collected by Western Geophysical in seismic surveys around the world. About the time Dr. Yilmaz wrote his 1987 book, "Seismic Data Processing" for the Society of Exploration Geophysicists (SEG), the company kindly made the 40 shot gathers available to the public.

Use the link below to download a compressed tar file of the 40 Oz shot gathers (15 Mb) from the Seismic Unix Wiki. These files are in "big endian" format:

https://wiki.seismic-unix.org/doku.php?id=tutorials:data/

To get these same shot gathers in "little endian" format, use the link below to download a compressed tar file of the 40 Oz shot gathers (15 Mb) from my web site:

http://www.seismicrocks.com/seismicunix/oz-little-endian.tgz

To uncompress a .tgz file, use the following command

```
$ tar -xvzf yourfile.tgz
```

The dollar sign ($) is the command line prompt, not part of the command. Note there is some importance to the order of the x, the v, the z, and the f. So, to avoid errors, use them in the order they are written.

3. The SU Makefile.config file and big endian versus little endian

Big endian and little endian refer to byte order. For one technical description I refer you to a Wikipedia page: Endianness: https://en.wikipedia.org/wiki/Endianness

Some years back, most computers were big endian; for example, computers made by Solaris and the IBM 360. However, personal computers that are x86 and AMD64 architecture are, natively, little endian. I installed Linux (Ubuntu) and SU on a Windows Surface Pro 3, an x86 hardware. Therefore, when I configured my SU installation, I edited my Makefile.config file so my SU datasets would be created in the native little endian format; that is, I edited the XDR line in my Makefile.config file before I installed the SU software. Let me explain …

When I install Seismic Unix; that is, when I run

```
$ make install
```

```
$ make xtinstall
```

and other "make" commands like these, these commands use the Makefile.config file.

For big endian configuration, I would use the XDR line as it appears in Makefile.config:

```
XDRFLAG    =   -DSUXDR
```

However, I want a little endian configuration to match my x86 hardware. So, before I ran the "make" commands, I commented out the XDR line by putting the hash (#) sign at the beginning of that line. Then I added the next line that has "XDRFLAG" on the left side of the equal sign and a space on the right side of the equal sign:

```
#XDRFLAG    =   -DSUXDR
```

```
XDRFLAG    =
```

How important is this? What is the data significance of big endian compared to little endian? At the beginning of the Makefile.config file, there is a comment that describes what it means to use XDR:

```
forces all SU data to be big endian
independent of processor architecture
```

4. Linux and suswapbytes

As I described above, I can choose whether the seismic data I create on my OS will be big endian or little endian. Because my machine is x86, I set my environment to be "little endian." And, because my SU environment is "little endian," when I use *surange* on a "big endian" SU file, I get the error message below:

```
$ surange < ozdata.25.su

surange: fgettr.c: on trace #2 number of samples in header (4325)
differs from number for first trace (13320)
```

That response does not make any sense to me, but I recognize it as the *surange* response to a mis-matched byte order data file. Fortunately, the solution is simple:

```
SUSWAPBYTES - SWAP the BYTES in SU data to convert data from big
endian to little endian byte order, and vice versa
```

Below, I use *suswapbytes* to create a new data file, then I use *surange*. Notice that I do not have to supply any parameters to *suswapbytes*.

```
$ suswapbytes < ozdata.25.su > ozle.25.su

$ surange < ozle.25.su
```

96 traces

tracl	1 96 (1 - 96)
tracr	1 96 (1 - 96)
fldr	10025
tracf	1 96 (1 - 96)
cdp	25 120 (25 - 120)
cdpt	1
trid	1
nvs	1
nhs	1
duse	1
scalel	1
scalco	1
counit	1
delrt	2
muts	2
ns	2100
dt	2000

Now I have a successful output from *surange*. The endianness of my data file was corrected for my machine by *suswapbytes*. I can process the file output from *suswapbytes*.