

Small discovery I made how to understand written examples on libm3 Internet page ([Library for working with m3 files](#)). Skip to second page to super fast get to the point of this.

Reading my previous writings comes handy because this time I do not go into such details.

Our new knowledge which we learn can be applied to any m3 binary file. In my study I observe still Bunker.m3 binary file. On libm3 page:

<http://code.google.com/p/libm3/wiki/MODL>

is written this code:

```
struct MODL23
{
    /*0x00*/ Reference name;
    /*0x08*/ uint32 version;
    /*0x0C*/ Reference sequenceHeader;
    /*0x14*/ Reference sequenceData;
    /*0x1C*/ Reference sequenceLookup;
    /*0x24*/ uint32 d2;
    /*0x28*/ uint32 d3;
    ....
}
```

In Bunker.m3 it looks *something* like this:

```
Line3: 3F 00 00 00 02 00 00 00 53 0D 00 00 0B 00 00 00
Line4: 03 00 00 00 0B 00 00 00 0F 00 00 00 0B 00 00 00
Line5: 02 02 00 00 00 00 00 00
```

Syntax looks like this:

Example1: /*0x00*/ Reference name;

where:

/*0x00*/ is index in HEX format (write it to any decent HEX2DEC converter and you get index in decimal).

Reference is data type, there are many different data types. All having their own name (like 'reference', 'uint32', etc).

name is a variable name. To make it clear you need to look struct MODL23 like this:

[index] [data type] [variable name]

Example1 starts at /*0x00*/ (0 in decimal) and ends before /*0x08*/ (8 in decimal). Yes, /*0x08*/ in HEX is 8, but Reference data type ends at 7 decimal. In HEX this part is:

```
Line3: 3F 00 00 00 02 00 00 00
```

Example2: /*0x08*/ uint32 version;

Whole data is between /*0x08*/ and /*0x0C*/ (in decimal between 8 and 12). In HEX this:

```
Line3: 53 0D 00 00
```

It starts at /*0x08*/ (53) and ends before /*0x0C*/ (0B, before 12 in decimal).

I hope you have got hang of it how to read indexes, for now I only write down matching pairs and you should see connection between them:

```
/*0x0C*/ Reference sequenceHeader;  
Line3: 3F 00 00 00 02 00 00 00 53 0D 00 00 0B 00 00 00  
Line4: 03 00 00 00 0B 00 00 00 0F 00 00 00 0B 00 00 00  
Line5: 02 02 00 00 00 00 00 00
```

```
/*0x14*/ Reference sequenceData;  
Line3: 3F 00 00 00 02 00 00 00 53 0D 00 00 0B 00 00 00  
Line4: 03 00 00 00 0B 00 00 00 0F 00 00 00 0B 00 00 00  
Line5: 02 02 00 00 00 00 00 00
```

```
/*0x1C*/ Reference sequenceLookup;  
Line3: 3F 00 00 00 02 00 00 00 53 0D 00 00 0B 00 00 00  
Line4: 03 00 00 00 0B 00 00 00 0F 00 00 00 0B 00 00 00  
Line5: 02 02 00 00 00 00 00 00
```

```
/*0x24*/ uint32 d2;  
Line3: 3F 00 00 00 02 00 00 00 53 0D 00 00 0B 00 00 00  
Line4: 03 00 00 00 0B 00 00 00 0F 00 00 00 0B 00 00 00  
Line5: 02 02 00 00 00 00 00 00
```

```
/*0x28*/ uint32 d3;  
Line3: 3F 00 00 00 02 00 00 00 53 0D 00 00 0B 00 00 00  
Line4: 03 00 00 00 0B 00 00 00 0F 00 00 00 0B 00 00 00  
Line5: 02 02 00 00 00 00 00 00 XX XX XX XX
```

Actually point is pretty simple as you can see, if you got new ideas how to understand binary m3 file educate me. I am happy if this text could be shorten, Maybe I went over complicated with such a simple task this time.