

Fake Data Studies

- fake_seb has successfully read a mBooNE style uncompressed data file and shipped events over sockets to fake_assembler. Glenn and Eric each have done this. Glenn did it on 2 machines: traffic across network for real, demonstrating faster-than-required data movement to an output file.
- Now, I can report I've done this with a compressed uBooNE data file format. So, I altered Georgia's Root Macro to write fake-Huff coding of events of varying size (in principle). Created file. Fired up fake_seb, fake_assembler and read up evts, shipped, wrote data to output file.
- => Will cross this small task off the list.

Fake data work so far, in context

- However, the important guts of getting the data from the DMA and handling the data in threads, tracking data through circular buffers, parsing data in assembler, integrating evts, has all been skipped in this exercise.
- I thus am now embarking on coding up the real thing. Soup to nuts, as they say.

uBooNE DAQ SEB code design

- There is nice legacy mBooNE C code that is suitable, elegant(?), not out-of-date. I propose to re-use it. Main differences being at the lowest level where the events come into the process memory: DMA into our PCIe card as opposed to mBooNE's VME DMA into Motorola monoboard.
- We will run 2 parent seb processes: one for SN data, one for triggered data. They look the same in almost every respect, except SN version will write to file, rather than over socket to assembler. Will worry about SN later.
- 3 threads per seb process: poller, getter, sender.
- PMT/Trigger crate's data structures slightly different.

uBooNE DAQ SEB code (2)

- getter in a while(FOREVER) walks through the memory as new memory is filled, increments a newEvent semaphore and packages up digitized signals into a buffer of events.
- sender in its own while(FOREVER) decrements newEvent semaphore when it's available, grabs pointer to latest event, sends it over sockets to assembler (file) for triggered (supernova) data. Decrements sentEvent semaphore.
- Third thread: poller in its own while (FOREVER) Nevis's code bundled up in its own thread.

uBooNE DAQ SEB code (3)

sem_post,wait
=> give,take a semaphore.

seb.c

```
{  
extern sem_t newData,eventToAssembler,newDMA;  
extern Event p_rec;  
  
...  
pthread_create(&sender,...)  
pthread_create(&getter,...)  
pthread_create(&poller,...)  
}
```

sender

```
{  
while()  
{  
sem_wait(&newData);  
  
fd=disk OR sockets  
  
write(fd, Event, ...)  
sem_wait(&eventToAssembler);  
}  
}
```

getter

```
{  
while()  
{  
  
sem_wait(&newDMA);  
while(ind<DMA.size())  
{  
sem_post(&newData);  
ind++;  
while( t<9600)  
{  
Event+t->ADC = p_rec+ind+t->ADC;  
}  
  
sem_post(&eventToAssembler);  
}  
}  
}
```

poller

```
{  
while()  
{  
  
// Reserve space, DMAlock p_rec  
DMA(p_rec, ...);  
sem_post(&newDMA);  
  
}  
}
```

Then ...

- Make/resurrect assembler to walk through each received buffer from each seb, and then send 10 crates' worth of integrated evts w crate_headers tacked on by me, to the process that writes it out.
- Right now the seb/run and host/assembler directories in git are in total upheaval and will not build. I need some weeks/assembler of code-writing time to get it all to work. First in (real) fake-data mode.