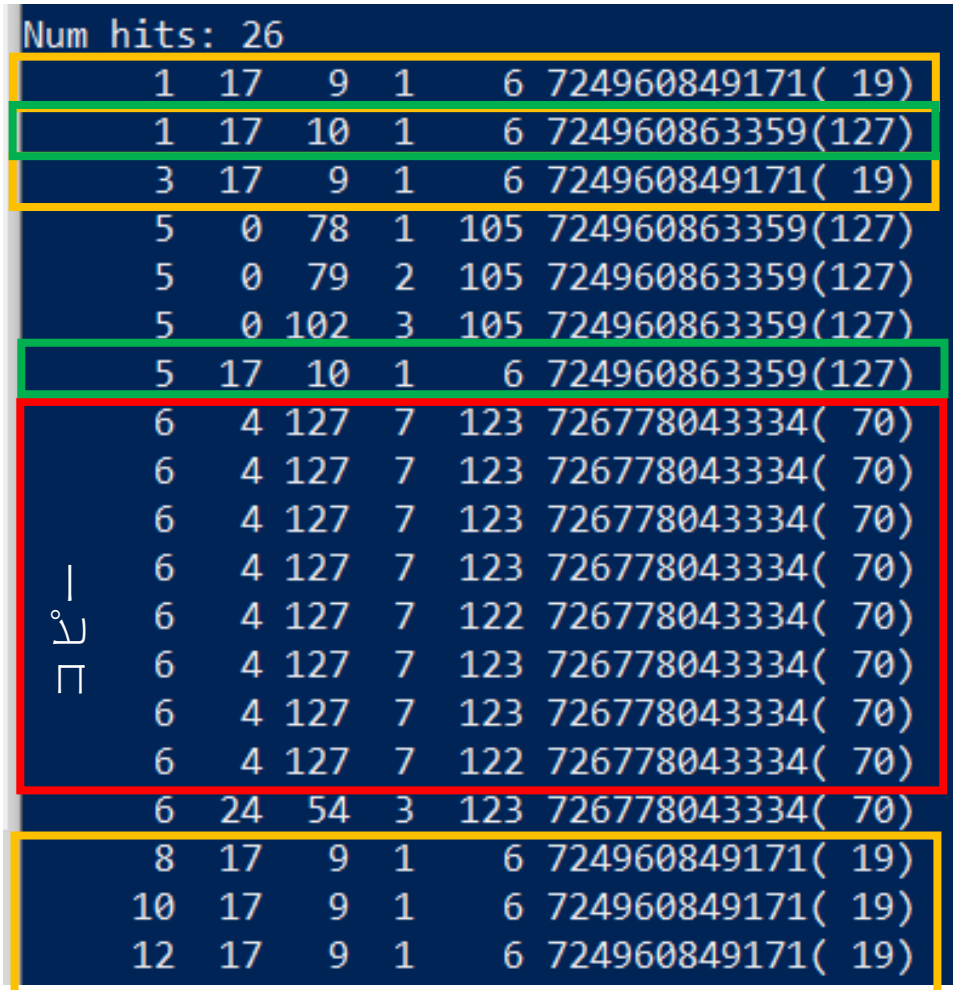
BCOとイベントの問題点

蜂谷

問題点

- EVTファイル中のEventに複数のBCOが含まれていることが分かった。　どういう問題か、なぜ起こるのかを調べる。

- Run9558を使ってデータを見る。
  - DAC設定を変えてMIPピークを見るためのデータ
  - 以前のデータ(8059)では、起こっていない



1イベント目

- 同一ラダー中のコピーヒット
- 別のラダーだが、Chip, Chan, ADC, BCOが同じコピーヒット,

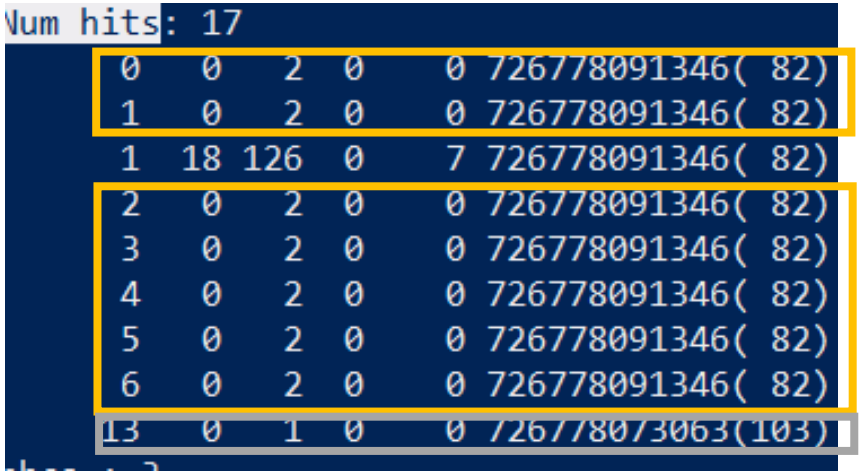
```
Num hits: 64
     0   0   1  0    0 726778073063(103)
     1   0   1  0    0 726778073063(103)
     1   2 108  0   24 726778073063(103)
     1   2 126  0   24 726778073063(103)
     1   3 109  0   24 726778073063(103)
     1   3 123  0   24 726778073063(103)
     1   3 125  0   24 726778073063(103)
     1   4 126  0   24 726778073063(103)
     1   5   0  0   25 726778073063(103)
     1   5 108  0   24 726778073063(103)
     1   5 113  0   24 726778073063(103)
     1   6   0  0   24 726778073063(103)
     1   6 126  0   24 726778073063(103)
     1   7 127  0   24 726778073063(103)
     1  10 116  0   24 726778073063(103)
     1  10 127  0   24 726778073063(103)
     1  11 124  0   24 726778073063(103)
     1  13 112  0   24 726778073063(103)
     1  14 102  0   24 726778073063(103)
     1  14 114  0   25 726778073063(103)
     1  14 127  0   25 726778073063(103)
     1  15 108  0   25 726778073063(103)
     1  15 122  0   24 726778073063(103)
     1  16  92  0   25 726778073063(103)
     1  16 116  0   25 726778073063(103)
     1  16 126  0   25 726778073063(103)
     1  17  16  0   25 726778073063(103)
     1  18 120  0   24 726778073063(103)
     1  21  80  0   25 726778073063(103)
     1  23 116  0   25 726778073063(103)
     2   0   1  0    0 726778073063(103)
     3   0   1  0    0 726778073063(103)
     4   0   1  0    0 726778073063(103)
     5   0   1  0    0 726778073063(103)
     7   0   1  0    0 726778073063(103)
     8   0   1  0    0 726778073063(103)
     9   0   1  0    0 726778073063(103)
    10   0   1  0    0 726778073063(103)
    11   0   1  0    0 726778073063(103)
    12   0   1  0    0 726778073063(103)
nbco : 2
```

2イベント目

- 別のラダーだが、Chip, Chan, ADC, BCOが同じコピーヒット,

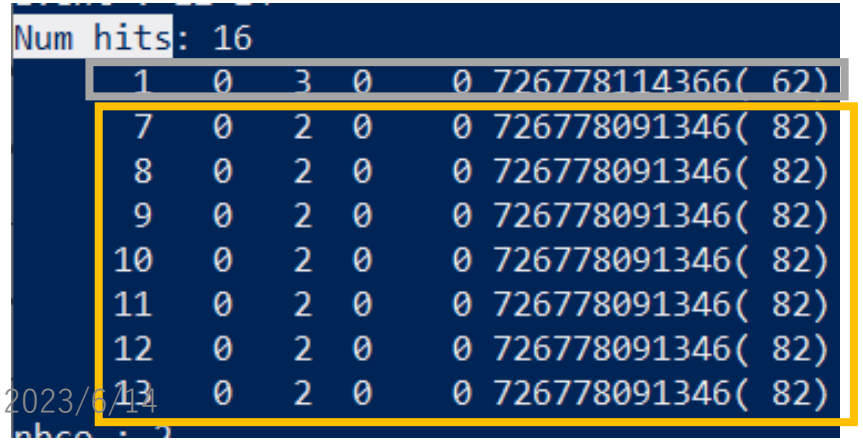
- Module＝1にたくさんヒットがあるが、すべてADC=0。Channelも110の周りが多い。

3イベント目



- 別のラダーだが、Chip, Chan, ADC, BCOが同じコピーヒット,
- 前のイベントのBCO_FULLと同じ。　別ラダーのコピーヒット

4イベント目



- 前のイベントのBCO_FULLと同じ。　別ラダーのコピーヒット

5イベント目

```
Num hits: 18
  0   0   4   0     0 726778147010( 66)
  0  14  91   2   115 726778147010( 66)
  2   0   4   0     0 726778147010( 66)
  3   0   4   0     0 726778147010( 66)
  4   0   4   0     0 726778147010( 66)
  5   0   4   0     0 726778147010( 66)
  6   0   3   0     0 726778114366( 62)
  6   0   4   0     0 726778147010( 66)
  7   0   4   0     0 726778147010( 66)
  8   0   4   0     0 726778147010( 66)
```

前のイベントのBCO_FULLと同じ。
別ラダーのコピーヒット

前のイベントのBCO_FULLと同じ。
別ラダーのコピーヒット

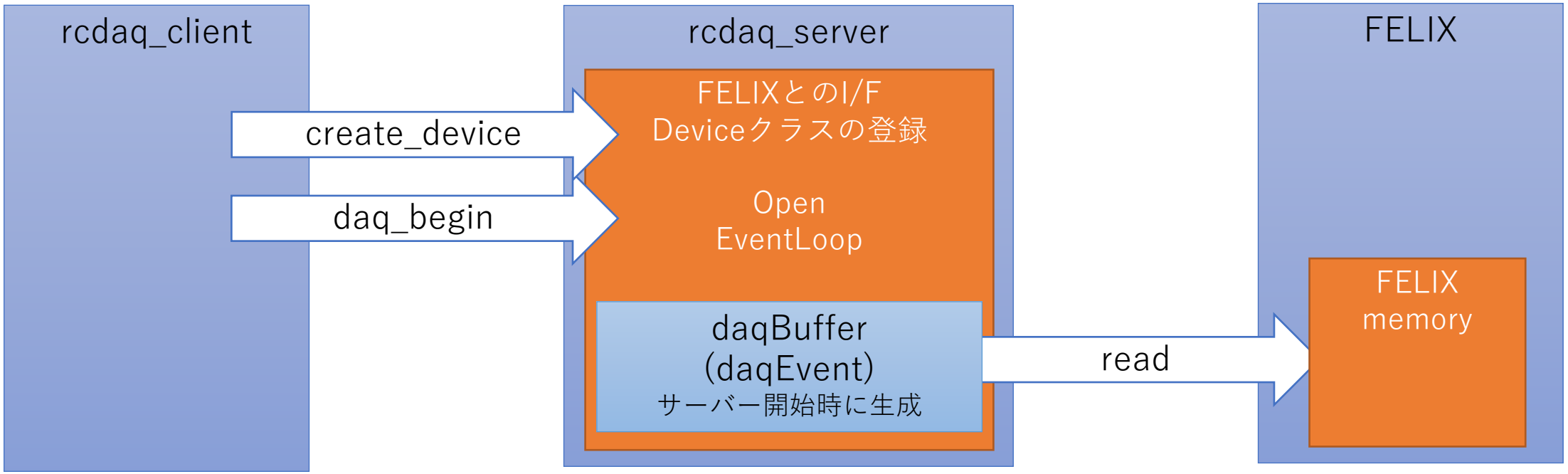
6イベント目

```
Num hits: 38
  0   0   5   0     0 726778172375( 87)
  0   0   6   0     0 726778199514( 90)
  1   0   4   0     0 726778147010( 66)
  1   0   5   0     0 726778172375( 87)
  2   0   6   0     0 726778199514( 90)
  3   0   6   0     0 726778199514( 90)
  4   0   6   0     0 726778199514( 90)
  5   0   5   0     0 726778172375( 87)
  5   0   6   0     0 726778199514( 90)
  6   0   5   0     0 726778172375( 87)
  7   0   6   0     0 726778199514( 90)
  8   0   6   0     0 726778199514( 90)
  9   0   4   0     0 726778147010( 66)
  9   0   5   0     0 726778172375( 87)
  9   0   6   0     0 726778199514( 90)
 10   0   4   0     0 726778147010( 66)
 10   0   5   0     0 726778172375( 87)
 10   0   6   0     0 726778199514( 90)
 11   0   4   0     0 726778147010( 66)
 12   0   4   0     0 726778147010( 66)
 13   0   4   0     0 726778147010( 66)
```

前のイベントのBCO_FULLと同じ。
別ラダーのコピーヒット

別ラダーのコピーヒット

RCDAQ server and client



rcdaq_server中のread関数で、FELIXのメモリ上にの全データをdaqBufferにコピーしてくる。

リード1回が1イベントになっている。その際、データの並べ替えなどは行っていない。

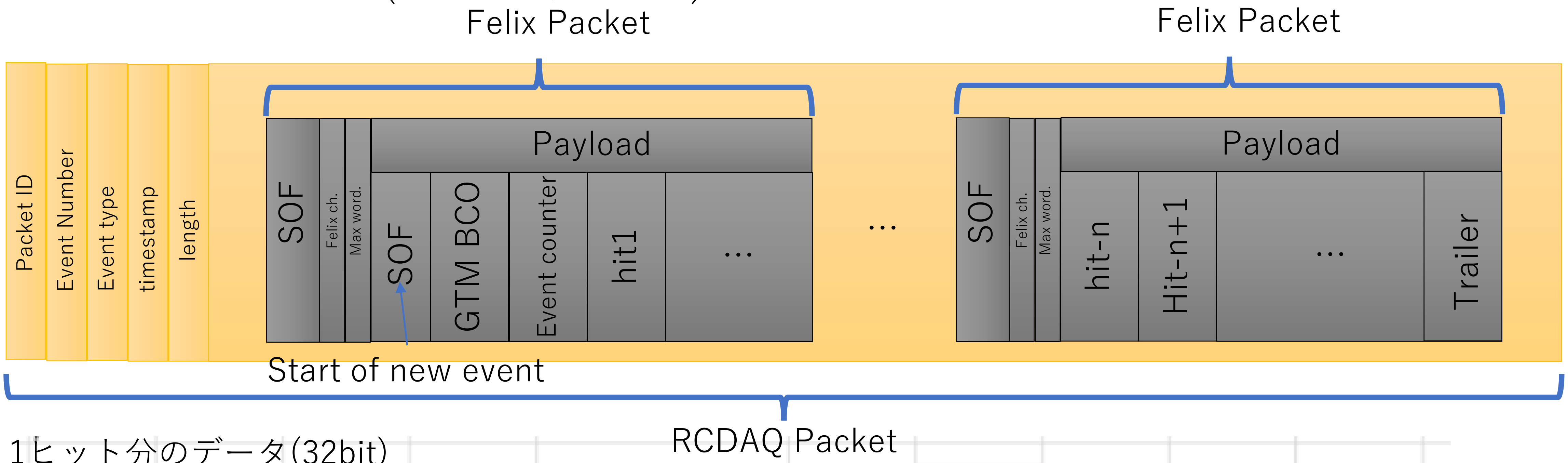
⇒いずれにしても並べ替えが必要

対策案

1. rcdaq_server上で並べ替えを行う。
2. FELIX上で並べ替えを行う

- rcdaq_serverで並べ替えるほうが簡単だがかなり遅い。
- streamingを考えるとFELIXで並べ替えるのがよいと思う。

INTTのRawData (EVTファイル)



1ヒット分のデータ(32bit)

General		Felix Full	Amplitude	Full ROC	Full FPHX	FPHX BCO	ADC	Chip ID	Channel ID
		1 bit	6 bits	1 bit	1 bit	7 bits	3 bits	6 bits	7 bits

- EVTファイルの構造を確認中

The transmission frame payload sends, 224 bits at a time, the event frame. The event frame has a variable length, depending on the number of hits of the collision.

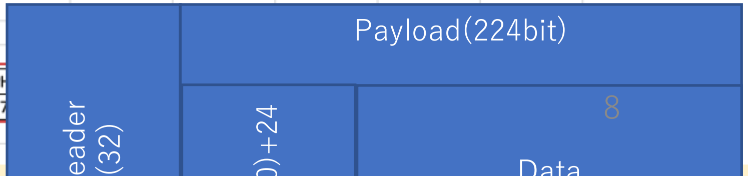
Transmission Frame			
SOF	Felix Channel	Max Words	Payload
24 bits	4 bits	4 bits	224 bits
CAF0F0	U	F	UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU

Transmission Frame Payload - Event Frame				
SOF	GTM BCO			Hits
24 bits	40 bits			(N*32) bits
CADEAD	UUUUUUUUUU			?

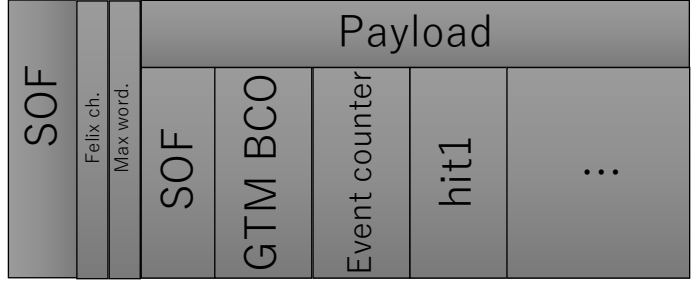
]

General		Felix Full	Amplitude	Full ROC	Full FPHX	FPH
		1 bit	6 bits	1 bit	1 bit	7

If (Physics mode){



EVTファイルの構造



		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB16, MSB16が入れ替わっている
1		Header								fee				len				Header																header=0xF000CAF0
2	1	Hit-Header								BCO								Hit-Header																Hit-header
3	2	BCO																																
4	3	adc				chip_id						chan_id										AMPL			full ROC	full FPX			FPHX_BCO					hit
5	4	adc				chip_id						chan_id										AMPL			full ROC	full FPX			FPHX_BCO					hit
6	5	adc				chip_id						chan_id										AMPL			full ROC	full FPX			FPHX_BCO					hit
7	6	adc				chip_id						chan_id										AMPL			full ROC	full FPX			FPHX_BCO					hit
8	7	adc				chip_id						chan_id										AMPL			full ROC	full FPX			FPHX_BCO					hit

- 全FELIXパケットがこのような構造になっている。
  - 256bits = 8words
  - Header + BCO = 3words, hit=最大5hit =5words
  - Decoderでは、この5つのヒットに同じBCOFULLが付く。

- このことをもう少し見ていく予定

まとめ

- EventをまたがってBCOFULLが存在することがわかった。

- 以前のデータではその割合は少ない。

- もう少し調査が必要
  - 生データを

rcdaq_client

rcdaq_serverに対して命令する。

- 実行時のマクロ
  - /home/phnxrc/operations/INTT/intt0_setup.sh
- マクロ内でDAQを実行している行
  - rcdaq_client load /home/purschke/softwarerepo/install_debian/lib/librcdaqplugin_intt.so
  - rcdaq_client create_device device_intt 1 $PACKETID 1 1
- その後、RCDAQ実行コマンドがある。
  - rcdaq_client daq_begin,
- コードの場所
  - INTT0@~purschke/softwarerepo/rcdaq/rcdaq_client.cc
  - INTT0@~purschke/softwarerepo/intt_plugin/
- rcdaq_client.cc :
  - create_device: handle_device :
    - rcdaq_server::r_create_device_1_svcないで、inttplugin::create_deviceを呼び出す
      - add_readoutdevice (new daq_device_intt(eventtype, subid, trigger));　でFELIXを登録する
  - daq_begin
    - rcdaq_server::r_action_1_svcのDAQ_BEGIN
      - result.status = daq_begin (ab->ipar[0], outputstream)
        - EventLoopのスレッドを実行 (enable_trigger関数の中)

rcdaq_server

- コードの場所
  - INTT0@~purschke/softwarerepo/rcdaq/
  - INTT0@~purschke/softwarerepo/intt_plugin/

- rcdaq_server : RPCを受信し、DAQ関数を実行
  - rcdaq_init: daqBufferを初期化し、writeBufferスレッドを実行
    - – Eventをファイルに書き出す
  - EventLoop

- daqBuffer
  - FELIXで取得したデータをPCメモリに保持するためのBufferクラス
    - daqPRDFEvent がイベント毎のデータ
    - writeout : ファイルに書き出す

int status = fillBuffer->nextEvent(etype,Event_number, Eventsize[etype]); // new daqEvent (PRDFのこと)

EventLoop内のreadout関数。　Bufferとデバイスから読み出したdataを関連付けている

```
for ( d_it = DeviceList.begin(); d_it != DeviceList.end(); ++d_it)
        len += fillBuffer->addSubevent ( (*d_it) );
```

addSubeventの中身：

```
    len = dev->put_data ( etype, &(evthdr->data[current]), left );
```

⇒　put_dataの中

```
    ret = read(_intt_fd, dest, _length);   // FELIXのBUFFER(メモリ)から データをコピーする。
```

とすると、FELIXのメモリから全データをコピーしてくるので、データの並べ替えなどは行っていない。

Unpackerコード

- ~purschke/softwarerepo/online_distribution/newbasic/oncsSub_idinttv0.cc

Packet->iValue/lValueで可能な引数

“NR_HITS”	ヒットのデータ構造
“FEE_LENGTH”	struct intt_hit {
“BCO” (l)	uint64_t bco;
“FEE”	uint16_t fee;
“CHANNEL_ID”	uint16_t channel_id;
“CHIP_ID”	uint16_t chip_id;
“ADC”	uint16_t adc;
“FPHX_BCO”	uint16_t FPHX_BCO;
“FULL_FPHX”	uint16_t full_FPHX;
“FULL_ROC”	uint16_t full_ROC;
“AMPLITUDE”	uint16_t amplitude;
“FULL_FPHX” (なぜか2つある)	uint16_t full_fphx;
“DATAWORD”	uint32_t word; };