## Subject: Re: Reading columns of binary data Posted by Foldy Lajos on Tue, 08 Aug 2006 18:35:00 GMT

View Forum Message <> Reply to Message

On Tue, 8 Aug 2006, JD Smith wrote:

```
On Tue, 08 Aug 2006 16:57:41 +0200, FÖLDY Lajos wrote:
>
>> On Mon, 7 Aug 2006, JD Smith wrote:
>>
>>> I'd regard this as yet another example of why IDL needs a fast
>>> compiled "side-loop" primitive for generic looping operations which
>>> don't need the full conveniences of the interpreter loop (ability to
>>> hit Control-C to interrupt, etc.). I should be able to say:
>>>
>>> a=fltarr(500000,/NO_ZERO)
>>> f=0.0
>>> for noblock i=0L,500000L-1L do begin
      point lun,un,i*100L
>>>
      readu,un,f
      a[i]=f
>>>
>>> endfor
>>>
>>> and not have it be 100x slower than the equivalent in C.
>>>
>>> JD
>>>
>>
>> Hi,
>> I mentioned assoc yesterday, so let's try again:
>>
>> arr=assoc(un, [0.])
>> for i=0L,500000L-1L do a[i]=arr[25l*i]
>> it will do the point_lun/readu in one step, so it should be a little bit
>> faster. But I think a faster loop here would not help much, as disk I/O
>> is the limiting factor.
> If that were the case then the IDL loop and a similar C loop (offsetting
> the file pointer and reading 4 bytes) should perform similarly. I haven't
  done the test (and would be happy to be proven wrong), but my guess is
> they wouldn't match by 1-2 orders of magnitude.
>
  JD
>
>
```

```
OK, I have written my homework :-)
IDL:
n=1000000001
openw, unit, 'test.dat', /get_lun
writeu, unit, findgen(n)
close, unit
k=n/100l
a=fltarr(k)
b=a
openr, unit, 'test.dat', /get_lun
f=0.
t=systime(1)
for j=0l,k-1l do begin
  point_lun, unit, j*400l
  readu, unit, f
  a[j]=f
  endfor
print, 'point/read: ', systime(1)-t
close, unit
openr, unit, 'test.dat', /get_lun
arr=assoc(unit, [0.])
t=systime(1)
for j=0l,k-1l do b[j]=arr[100l*j]
print, 'assoc: ', systime(1)-t
close, unit
end
C (without any error checking, gcc-4.1.1 -O3, linux 32 bit):
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
int main()
{
  struct timeval tval:
  struct timezone tzone;
  double t1, t2;
  int j;
```

```
float* fp;
  fp=(float*)malloc(1000000l*sizeof(float));
  FILE* file=fopen("test.dat", "r");
  gettimeofday(&tval, &tzone);
  t1=tval.tv_sec+1.0e-6*tval.tv_usec;
  for (j=0; j<10000001; j++)
    { fseek(file, j*400, SEEK_SET);
     fread(fp+j, sizeof(float), 1, file);
    }
  gettimeofday(&tval, &tzone);
  t2=tval.tv_sec+1.0e-6*tval.tv_usec;
  printf("time: %f\n", t2-t1);
}
Best times:
  IDL: point/read: 3.68
  IDL: assoc:
                   3.05
  C:
                1.39
YMMV.
regards,
lajos
```