Subject: Re: Use IDL6.0 to read gcc3.4(Mingw32) written data
Posted by Nianming Zuo on Wed, 08 Aug 2007 06:10:28 GMT
View Forum Message <> Reply to Message

I am really frustrated by the interface between IDL6.0 and
gcc3.4.2(Mingw32).
The following is the test program.

**********

// c program, compiled by Mingw32 gcc3.4.2 on MS Windows XP

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define col 8
#define row 4
#define hit 1

int main()

{
   float mat[row][col][hit];
   int i, j, k;
   FILE *fn;

   for (i=0; i<row; i++)
  {
     for (j=0; j<col; j++){
    for (k=0; k<hit; k++){
    mat[i][j][k] = i + j/2.0 +k/5.0 + (float)i/(j+1);
  }
  }
  }

   fn = fopen("cmat.dat", "w");
   printf("fn = %d", fn);
   if(fn == NULL)
  {
    printf("Can't open cmat.dat to write\n");
    exit(1);
  }
   fwrite(&i, sizeof(int), 1, fn);
   fwrite(mat, sizeof(float), col*row*hit, fn);
   fwrite(&j, sizeof(int), 1, fn);
   fclose(fn);
```

```
   exit(0);
}

*****************************************
; IDL6.0 program, on the same OS.

device, retain = 2

; To test whether swap is necessary.

openr,lun,'cmat.dat', /GET_LUN
; -- Check the record size
RecordSize = 10000L * 4L
RecordSize_Test = 0L
READU, lun, RecordSize_Test
IF ( RecordSize_Test NE RecordSize ) THEN $
   Swap = 1 $
ELSE  $
   Swap = 0

; -- Close the file
FREE_LUN, lun

print, "Swap= ", Swap


cmat = fltarr(8,4)
openr, lun, "cmat.dat",/GET_LUN,/swap_endian

readu, lun, ii
readu, lun, cmat
readu, lun, jj


print, "ii", ii
print, "cmat", cmat
print, "jj", jj


end

***********************************
```

Previously, I have easily implemented the interface above on Linux/
Mandriva 10.2 .
But now it can not be repeated on MS Windows.
The cpu of computer are all Intel P4 .

Thanks,

Tony

On 8 8 ,  11 52 , Nianming Zuo <nianm...@gmail.com> wrote:
> Thank you, Paul, David, Mike, chl and other guys.
> I have read the links (and other related links), and it is really
> helpful for my puzzels.
>
> And now, I have another problem. (The following are on MS Windows
> XP(sp2))
>
> IDL6.0 can not read data saved by gcc3.4. (Mingw32)
>
> in "gccfile.dat", I saved a seriers of data, including int and float
> type, using
> gf = fopen("gccfile.dat", "w");
> fwrite(NLAM, sizeof(int),1, gf);
> //repeat this sentence to store several vars,
> NLAM,R,D,H,ALAM0,ALAM1,DLAM , with different type.
>
> Now, I want to read datas in "gccfile.dat", and I have tried many
> methods.
>
> Way 1:
> openr, lun, "gccfile.dat", /GET_LUN
> readu,lun,NLAM,R,D,H,ALAM0,ALAM1,DLAM
> print, NLAM,R,D,H,ALAM0,ALAM1,DLAM
>
> It prints strange data like 3.36641e+038, and prompts:
> % Program caused arithmetic error: Floating underflow
> % Program caused arithmetic error: Floating illegal operand
>
> Way 2: (learn from this forum. THANKS :)  )
>
> openr, lun, "gccfile.dat", /GET_LUN, /SWAP_ENDIAN
> readu,lun,NLAM,R,D,H,ALAM0,ALAM1,DLAM
> print, NLAM,R,D,H,ALAM0,ALAM1,DLAM
>
> It still prints the garbage!
>
> I have tested the endian-ness things with (from Paul. Thanks):
>
> openr,lun,'shepp.sgm', /GET_LUN  ; "shepp.sgm" is my file.
>  ; -- Check the record size
> RecordSize = 10000L * 4L
> RecordSize_Test = 0L
> READU, lun, RecordSize_Test
> IF ( RecordSize_Test NE RecordSize ) THEN $

>  Swap = 1 $
> ELSE  $
>  Swap = 0
>
> ; -- Close the file
> FREE_LUN, lun
>
> print, "Swap", Swap
>
> ; The above Swap turns out 1. So swap is necessary.
>
> Way 3:
>
> openr, lun, "gccfile.dat", /GET_LUN ; Without /SWAP_ENDIAN
> readu,lun,NLAM,R,D,H,ALAM0,ALAM1,DLAM
> NLAM = SWAP_ENDIAN(NLAM)
> print, NLAM,R,D,H,ALAM0,ALAM1,DLAM
>
> Amazingly,  NLAM (integer) is wrong, and other vars (float) are right!
>
> I am totally confused by its behavious!
>
> Additionally, I have tried another ways, and did't take effect.
> byteorder, NLAM,R,D,H,ALAM0,ALAM1,DLAM, /lswap
>
> One suggested "binread" function, but it doesn't exist in IDL6.0.
>
> Thanks,
>
> Tony
>
> On 8 7 ,  8 39 , Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
>
>
>
>> Nianming Zuo wrote:
>>> Dear all,
>
>>> I have sufferred file read/write problems between Fortran 90/95 and
>>> IDL 6.0.
>
>>> My Fortran compiler:
>>> Silverfrost ftn95, Compatable for Fortran 77/90/95
>>> http://www.silverfrost.com/12/ftn95/ftn95_feature_details.as p
>
>>> IDL 6.0 (Interactive Data Language, RSI)
>
>>> Both are in MS Windows XP(sp2) OS system.

>
>>> Write data to a file by use of Fortran:
>>> dimension dat(m, n)
>>> !..... Manipulations..., matrix dat(m, n) is float
>>> open(unit=11, file="file.dat", form="unformatted")
>>> write(11) dat
>>> !.....
>>> ! The above are really f77 code, so I guess it is related to Compiler.
>
>>> Read the data above by IDL6.0: (Way 1)
>>> dat = fltarr(m,n)
>>> openr, 1, 'file.dat'
>>> readu, 1, b, dat, b
>
>>> In "readu, 1, b, dat, b", the "b"s are used to skip the record area in
>>> Fortran data format.
>>> Unfortunately, it can not get the right result, and prompts "End of
>>> the file"
>
>>> I have also tried another way in IDL: (Way 2)
>>> dat = fltarr(m,n)
>>> openr, 1, 'file.dat' /f77_unformatted
>>> readu, 1, dat
>
>>> But, it prompts,
>>> "% READU: Corrupted f77 unformatted file detected. "
>
>>> For the above Fortran code, when it is compied by g77, IDL can read it
>>> by Way 2.
>
>>> So, I doubt that different compilers give different response to the
>>> standard Fortran sentences ?
>>> Since there is no f90_unformatted or f95_unformatted, f77/f90/f95 will
>>> produce the same record for the "open-write" sentence.
>
>>> Now, how can I read ftn95 compiled output data by IDL6.0 ? I have
>>> searched this forum, but without any desirable results.
>
>> Have a lookee at:
>
>> http://groups.google.com/group/comp.lang.idl-pvwave/browse_t hread/thr...
>
>> (Crikey that's a long link)
>
>> cheers,
>
>> paulv- -
>

>> -  --  -
>
> -  -

Subject: Re: Use IDL6.0 to read gcc3.4(Mingw32) written data
Posted by Nianming Zuo on Wed, 08 Aug 2007 10:07:53 GMT
View Forum Message <> Reply to Message

Dear all,

I am keeping penetrating the said problem.

Although the endian-ness test is swap=1, it doesn't seems that the garbage is aroused by the byteorder.

In c program, I try to write 1(int) + 80*4(float) + 1(int) bytes to "gccfile.dat" ((1+80*4+1)*4 bytes). Then I tried to read this file by IDL 6.0.

openr, lun, 'gccfile.dat',/GET_LUN  ; Normal

For the INT parameters, by use of,
num = long(1) ! so it is 4 bytes.
readu, lun, num

For float parameters, by use of
mat = fltarr(80,40)
readu, lun, mat

Unfortunately, in matrix "mat", the first 138 data are right, but the else are garbage!

BTW, the file size of "gccfile.dat" should be ((1+80*4+1)*4 bytes, but the actual size is 1290 bytes. What are the extra 2 bytes ?

I am keeping fighting it.....

cheers,

Tony

On 8 8 ,  2 10 , Nianming Zuo <nianm...@gmail.com> wrote:
> I am really frustrated by the interface between IDL6.0 and
> gcc3.4.2(Mingw32).
> The following is the test program.
>
> ********************************************
>
> // c program, compiled by Mingw32 gcc3.4.2 on MS Windows XP
>
> #include <stdio.h>
> #include <stdlib.h>
> #include <math.h>
>
> #define col 8
> #define row 4
> #define hit 1
>
> int main()
>
> {
>  float mat[row][col][hit];
>  int i, j, k;
>  FILE *fn;
>
>  for (i=0; i<row; i++)
>  {
>  for (j=0; j<col; j++){
>  for (k=0; k<hit; k++){
>  mat[i][j][k] = i + j/2.0 +k/5.0 + (float)i/(j+1);
>  }
>  }
>  }
>
>  fn = fopen("cmat.dat", "w");
>  printf("fn = %d", fn);
>  if(fn == NULL)
>  {
>  printf("Can't open cmat.dat to write\n");
>  exit(1);
>  }
>  fwrite(&i, sizeof(int), 1, fn);

```
>    fwrite(mat, sizeof(float), col*row*hit, fn);
>    fwrite(&j, sizeof(int), 1, fn);
>    fclose(fn);
>
>    exit(0);
>
> }
>
> ****************************************
> ; IDL6.0 program, on the same OS.
>
> device, retain = 2
>
> ; To test whether swap is necessary.
>
> openr,lun,'cmat.dat', /GET_LUN
> ; -- Check the record size
> RecordSize = 10000L * 4L
> RecordSize_Test = 0L
> READU, lun, RecordSize_Test
> IF ( RecordSize_Test NE RecordSize ) THEN $
>    Swap = 1 $
> ELSE  $
>    Swap = 0
>
> ; -- Close the file
> FREE_LUN, lun
>
> print, "Swap= ", Swap
>
> cmat = fltarr(8,4)
> openr, lun, "cmat.dat",/GET_LUN,/swap_endian
>
> readu, lun, ii
> readu, lun, cmat
> readu, lun, jj
>
> print, "ii", ii
> print, "cmat", cmat
> print, "jj", jj
>
> end
>
> ************************************
> Previously, I have easily implemented the interface above on Linux/
> Mandriva 10.2 .
> But now it can not be repeated on MS Windows.
> The cpu of computer are all Intel P4 .
```

>
> Thanks,
>
> Tony
>
> On 8 8 , 11 52 , Nianming Zuo <nianm...@gmail.com> wrote:
>
>
>
>> Thank you, Paul, David, Mike, chl and other guys.
>> I have read the links (and other related links), and it is really
>> helpful for my puzzels.
>
>> And now, I have another problem. (The following are on MS Windows
>> XP(sp2))
>
>> IDL6.0 can not read data saved by gcc3.4. (Mingw32)
>
>> in "gccfile.dat", I saved a seriers of data, including int and float
>> type, using
>> gf = fopen("gccfile.dat", "w");
>> fwrite(NLAM, sizeof(int),1, gf);
>> //repeat this sentence to store several vars,
>> NLAM,R,D,H,ALAM0,ALAM1,DLAM , with different type.
>
>> Now, I want to read datas in "gccfile.dat", and I have tried many
>> methods.
>
>> Way 1:
>> openr, lun, "gccfile.dat", /GET_LUN
>> readu,lun,NLAM,R,D,H,ALAM0,ALAM1,DLAM
>> print, NLAM,R,D,H,ALAM0,ALAM1,DLAM
>
>> It prints strange data like 3.36641e+038, and prompts:
>> % Program caused arithmetic error: Floating underflow
>> % Program caused arithmetic error: Floating illegal operand
>
>> Way 2: (learn from this forum. THANKS :) )
>
>> openr, lun, "gccfile.dat", /GET_LUN, /SWAP_ENDIAN
>> readu,lun,NLAM,R,D,H,ALAM0,ALAM1,DLAM
>> print, NLAM,R,D,H,ALAM0,ALAM1,DLAM
>
>> It still prints the garbage!
>
>> I have tested the endian-ness things with (from Paul. Thanks):
>
>> openr,lun,'shepp.sgm', /GET_LUN ; "shepp.sgm" is my file.

>>  ; -- Check the record size
>>  RecordSize = 10000L * 4L
>>  RecordSize_Test = 0L
>>  READU, lun, RecordSize_Test
>>  IF ( RecordSize_Test NE RecordSize ) THEN $
>>     Swap = 1 $
>>  ELSE  $
>>     Swap = 0
>
>>  ; -- Close the file
>>  FREE_LUN, lun
>
>>  print, "Swap", Swap
>
>>  ; The above Swap turns out 1. So swap is necessary.
>
>>  Way 3:
>
>>  openr, lun, "gccfile.dat", /GET_LUN ; Without /SWAP_ENDIAN
>>  readu,lun,NLAM,R,D,H,ALAM0,ALAM1,DLAM
>>  NLAM = SWAP_ENDIAN(NLAM)
>>  print, NLAM,R,D,H,ALAM0,ALAM1,DLAM
>
>>  Amazingly,  NLAM (integer) is wrong, and other vars (float) are right!
>
>>  I am totally confused by its behavious!
>
>>  Additionally, I have tried another ways, and did't take effect.
>>  byteorder, NLAM,R,D,H,ALAM0,ALAM1,DLAM, /lswap
>
>>  One suggested "binread" function, but it doesn't exist in IDL6.0.
>
>>  Thanks,
>
>>  Tony
>
>>  On 8 7 ,   8 39 , Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
>
>>>  Nianming Zuo wrote:
>>>>  Dear all,
>
>>>>  I have sufferred file read/write problems between Fortran 90/95 and
>>>>  IDL 6.0.
>
>>>>  My Fortran compiler:
>>>>  Silverfrost ftn95, Compatable for Fortran 77/90/95
>>>>  http://www.silverfrost.com/12/ftn95/ftn95_feature_details.as p
>

>>>> IDL 6.0 (Interactive Data Language, RSI)
>
>>>> Both are in MS Windows XP(sp2) OS system.
>
>>>> Write data to a file by use of Fortran:
>>>> dimension dat(m, n)
>>>> !..... Manipulations..., matrix dat(m, n) is float
>>>> open(unit=11, file="file.dat", form="unformatted")
>>>> write(11) dat
>>>> !.....
>>>> ! The above are really f77 code, so I guess it is related to Compiler.
>
>>>> Read the data above by IDL6.0: (Way 1)
>>>> dat = fltarr(m,n)
>>>> openr, 1, 'file.dat'
>>>> readu, 1, b, dat, b
>
>>>> In "readu, 1, b, dat, b", the "b"s are used to skip the record area in
>>>> Fortran data format.
>>>> Unfortunately, it can not get the right result, and prompts "End of
>>>> the file"
>
>>>> I have also tried another way in IDL: (Way 2)
>>>> dat = fltarr(m,n)
>>>> openr, 1, 'file.dat' /f77_unformatted
>>>> readu, 1, dat
>
>>>> But, it prompts,
>>>> "% READU: Corrupted f77 unformatted file detected. "
>
>>>> For the above Fortran code, when it is compied by g77, IDL can read it
>>>> by Way 2.
>
>>>> So, I doubt that different compilers give different response to the
>>>> standard Fortran sentences ?
>>>> Since there is no f90_unformatted or f95_unformatted, f77/f90/f95 will
>>>> produce the same record for the "open-write" sentence.
>
>>>> Now, how can I read ftn95 compiled output data by IDL6.0 ? I have
>>>> searched this forum, but without any desirable results.
>
>>> Have a lookee at:
>
>>> http://groups.google.com/group/comp.lang.idl-pvwave/browse_t hread/thr...
>
>>> (Crikey that's a long link)
>
>>> cheers,

>
>>> paulv- -
>
>>> - -- -
>
>> - -- -
>
> - -

---

Subject: Re: Use IDL6.0 to read gcc3.4(Mingw32) written data
Posted by badjelly.witch on Wed, 08 Aug 2007 21:29:50 GMT
View Forum Message <> Reply to Message

I don't know if the following is the cause of your problem, but in the
following code snippet...

cmat = fltarr(8,4)
openr, lun, "cmat.dat",/GET_LUN,/swap_endian

readu, lun, ii
readu, lun, cmat
readu, lun, jj

print, "ii", ii
print, "cmat", cmat
print, "jj", jj

...I see no sign that you have created the variables ii and jj before
reading them. In this case the variables will be created as scalars of
type float, which will not do what you intend.

On another tack, I suggest you download and install Hedit (it's free)

http://www.yurisw.com/HEdit.htm

With this tool you can look at the contents of your binary file. It
has a handy display at the top: you move the cursor to any position
and it shows the value that the 1, 2, 4 or 8 bytes starting at that
position will have if interpreted as a binary, byte, short, long,
float or double scalar. It should help you work out why one of your
files is 2 bytes longer than you expected.

On yet another tack, I don't understand why you would have to open
your files with SWAP_ENDIAN if they are being written on the same
platform as they are being read.

---

Subject: Re: Use IDL6.0 to read gcc3.4(Mingw32) written data
Posted by Nianming Zuo on Thu, 09 Aug 2007 07:03:48 GMT
View Forum Message <> Reply to Message

Mark,

Thank you for your replay and your recommendation.

Finally I found the crux of the matter. It was caused by the different
way
to treat " fwrite(fp, "w") " sentence on Linux/Unix and MS Windows.
(Initially, my program was developped on Linux and currently to
transfer it)

On MS Windows, when I use IDL6.0 to read (openr, readu) data generated
by
gcc3.4.2 (Mingw32), it produces large values and prompts
" Program caused arithmetic error: Floating underflow " , so I
suspected
it is caused by gcc (transferred from Linux), then I used
SWAP_ENDIAN .
Perhaps it is a unreasonale trial.

Any way, it has been defeated!!

Thank you!

Tony

On 8 9 , 5 29 , Mark Hadfield <badjelly.wi...@gmail.com> wrote:
> I don't know if the following is the cause of your problem, but in the
> following code snippet...
>
> cmat = fltarr(8,4)
> openr, lun, "cmat.dat",/GET_LUN,/swap_endian
>
> readu, lun, ii
> readu, lun, cmat
> readu, lun, jj
>
> print, "ii", ii
> print, "cmat", cmat
> print, "jj", jj
>
> ...I see no sign that you have created the variables ii and jj before
> reading them. In this case the variables will be created as scalars of
> type float, which will not do what you intend.
>
> On another tack, I suggest you download and install Hedit (it's free)

> 
> http://www.yurisw.com/HEdit.htm
> 
> With this tool you can look at the contents of your binary file. It
> has a handy display at the top: you move the cursor to any position
> and it shows the value that the 1, 2, 4 or 8 bytes starting at that
> position will have if interpreted as a binary, byte, short, long,
> float or double scalar. It should help you work out why one of your
> files is 2 bytes longer than you expected.
> 
> On yet another tack, I don't understand why you would have to open
> your files with SWAP_ENDIAN if they are being written on the same
> platform as they are being read.

---

Subject: Re: Use IDL6.0 to read gcc3.4(Mingw32) written data
Posted by badjelly.witch on Thu, 09 Aug 2007 22:33:22 GMT
View Forum Message <> Reply to Message

On Aug 9, 7:03 pm, Nianming Zuo <nianm...@gmail.com> wrote:
> Any way, it has been defeated!!

Good news!

---