Subject: Redhat Linux 6.0 troubles continue
Posted by Bruce L. Gotwols on Thu, 03 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Hello all IDL Linux users,
On May 10th Jeremy Sanders posted a workaround which allows IDL to run under the latest version of Linux available from Redhat (6.0). I found this workaround to be very useful so I have included it below for any latecomers. However, I have just discovered another failure mode. The spawn command no longer works. This is very important if you want to print anything from within IDL, as well as a myriad of other ways we use spawn. To demonstrate the failure, all you have to do is spawn any unix command, e.g..

```
spawn, 'ls'
```

The error messages that come back are:

```
tcsh: /usr/i386-glibc20-linux/lib/libcrypt.so.1: no version
information available (required by tcsh)

tcsh: /usr/i386-glibc20-linux/lib/libc.so.6: no version
information        available (required by tcsh)

tcsh: error in loading shared libraries:
/usr/i386-glibc20-linux/lib/libc.so.6:
undefined symbol: _dl_global_scope_end
```

My interpretation of this failure mode is that the interactive shell I am running (tcsh, but with identical failures under bash) is unhappy with the old shared libraries that the workaround uses. So I'm damned if I do and damned if I don't. Surprisingly, I find that call_external does work.

Why so much interest in upgrading to redhat 6.0? The answer is that it is the first major release that uses the new Linux 2.2 kernel. This kernel, for the first time, routinely supports Symmetric Multi Processing (SMP). SMP was available in earlier kernels as a patch, but this is the first release where it comes into it's full glory from the ground up. We at JHU/APL are deep into using multi processors as a way of simultaneously doing data acquisition as well as Quick-Look analysis in real time.

I purchased IDL for Linux about one month ago. I was not told until a week after the purchase that it had not been recompiled since Redhat version 5.1 which is on the order of one year ago! Nor would RSI give me a schedule for the next recompile, saying only that they would contact their Linux developer about the problem. So apparently they don't even have this capability in-house. I am left wondering what is the use of paying for an IDL maintenance contract, since RSI is obviously doing virtually no maintenance, at least with the Linux

version.

I don't know if there are enough Linux users out there to bring enough pressure to bear on RSI to clean up this problem. If there are, then I propose we band together and collect a list of bugs which would be submitted to RSI. That way, when they finally do get around to doing something, perhaps they will fix some problems that might go unnoticed until the next maintenance fix (which might be a year from now!). I will volunteer to collect the bug reports, collate them into a readable whole, and submit them to RSI as well as to this news group. I suggest the deadline for submission be June 15th since after that you'll have forgotten all about this message.

To start off the list here are several bugs and deficiencies that I am aware of:

1. Recompile IDL under Redhat 6.0 (and any other vendors versions if they are supported). This should fix the fatal bugs discussed below as well as the bug introduced by the workaround discussed above.

2. Provide huge file support (> 2GB). This has been advertised as being available under IDL 5.2 for about a year now, but it wasn't until I tried to use it that I discovered it is not available on any Intel based box (This means both Windows and Linux).

3. Fix systime() so it once again returns time with a precision greater than one second. This affects the suite of time test programs such as time_test2 such that they can only record integer number of seconds, and therefore their results under Linux are inaccurate.

Cheers, Bruce Gotwols

On 10 May Jeremy Sanders gave a partial workaround:

> We managed to get IDL to work by installing the compatability > libraries. I then modified the idl script (the one which runs the > binary) at the end to read:
>
> app=$IDL_DIR/bin/bin.$OS$OSVER$ARCH/$APPLICATION
> if [-e /lib/ld-2.0.7.so] ; then
> exec $app $* $APP_ARGS
> else
> libdir=/usr/i386-glibc20-linux/lib
> export LD_LIBRARY_PATH="${libdir}:${LD_LIBRARY_PATH}" exec
> ${libdir}/ld-2.0.7.so $app $* $APP_ARGS
> fi

> This seems to work, although we haven't tried running programs with
> external shared libraries.

But on 19 May, G. Hugh Song described a continuing problem:
> I thought that all the compatibility problem has been solved by users. > In my case, one problem was solved following Jeremy's patch. But > when I lauched my application, I got:
> /usr/local/rsi/idl_5.2/bin/bin.linux/idl: error in loading shared > libraries
> /usr/X11R6/lib/libX11.so.6: undefined symbol: __bzero

> So, have you guys solved all the compatibility problem as much as you > want?
(The answer obviously is no, as described at the beginnig of this message.)

Bruce L. Gotwols
Johns Hopkins University, Applied Physics Lab., Laurel MD 20723
Internet: gotwols@tesla.jhuapl.edu
Phone: 240-228-4543 FAX: 240-228-5548
Space Oceanography Group Home Page -- http://fermi.jhuapl.edu