Build and Source Control - Bug #5166

libp2j.so links to unpatched ncurses 6.2 instead of the patched ncurses 5.9

02/25/2021 04:39 PM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Greg Shah	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

Related issues:

Related to Build and Source Control - Bug #3298: libp2j build links with ncur...	**Closed**
Related to Build and Source Control - Support #5167: using static linking to ...	**Closed**

History

#1 - 02/25/2021 05:56 PM - Greg Shah

I'm running Ubuntu 20.10. When I installed it in December 2020, I used the Patching NCURSES approach along with the patch_ncurses.sh script and the ncurses 5.9 patches (ncurses_curses_h_in_20060828.patch and ncurses_lib_getch_c_v5.7_20090512.patch). The result allowed FWD to be built and was tested successfully with our Hotel ChUI application using the native terminal client.

I did setup that patching script as an apt-get hook so that it would automatically re-patch when new packages are installed. This seemingly was working as well.

Sometime recently things changed. Although FWD builds without errors, it somehow links to a libncurses.so.6 which is not patched. When the terminal client is started now, it fails with java: symbol lookup error: /home/ges/projects/samples/hotel_chui/deploy/lib/libp2j.so: undefined symbol: auto_getch_refresh.

Both 5.9 and 6.2 are installed on this system and when I build libp2j.so, it links with 6.2 which is unpatched. My patch_ncurses.sh uses the 5.7 files and it patches the 5.9 version of libncurses. But we link with the unpatched 6.2 so we fail at runtime.

#2 - 02/25/2021 06:14 PM - Roger Borrello

There are 6.1 patches from last July on that same page that I have been using for a while. I updated the instructions there to utilize those patches, and included the patch_ncurses6.sh in the Wiki. This does work with 6.2, as well. From /usr/lib:

```
-rw-r--r--   1 root root  727444 Feb 25 16:45 libncurses.a
-rw-r--r--   1 root root  128040 Feb 25 16:45 libncurses++.a
-rw-r--r--   1 root root 6525524 Feb 25 16:45 libncurses_g.a
-rw-r--r--   1 root root  858744 Feb 25 16:45 libncurses++_g.a
lrwxrwxrwx   1 root root      15 Feb 25 16:45 libncurses.so -> libncurses.so.6
lrwxrwxrwx   1 root root      17 Feb 25 16:45 libncurses.so.6 -> libncurses.so.6.2
-rwxr-xr-x   1 root root  415704 Feb 25 16:45 libncurses.so.6.1
-rwxr-xr-x   1 root root  425184 Dec 15 10:15 libncurses.so.6.2
```

#3 - 02/25/2021 06:15 PM - Greg Shah

I've reviewed #3298 and have looked more carefully at Patching NCURSES. I've managed to resolve my issue by doing the following:

1. I removed libncurses5-dev using sudo apt-get remove libncurses5-dev. Per Development Headers I confirmed that this was for version 6 and thus it should not be needed. To be fair, I did not see a failure in building FWD before I removed it, nor after its removal. And it did not solve the problem on its own. So I don't know if it is really needed or not.

2. I downloaded Roger's updated patch_ncurses6.sh and the v6.1 patches (ncurses_curses_h_in_v6.1_20200708.patch and ncurses_lib_getch_c_v6.1_20200708.patch). I made the new script executable (chmod +x) and ran it.

3. I rebuilt FWD including the native step (.gradlew core).

4. I ran ant deploy.prepare for Hotel ChUI and restarted the server.

At this point it worked.

If I look at the ldd build/lib/libp2j.so output:

```
        linux-vdso.so.1 (0x00007fffa85ba000)
        libffi.so.8 => /lib/x86_64-linux-gnu/libffi.so.8 (0x00007fc723eee000)
        libncurses.so.5 => /lib/libncurses.so.5 (0x00007fc723e94000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fc723e8e000)
        libutil.so.1 => /lib/x86_64-linux-gnu/libutil.so.1 (0x00007fc723e89000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc723c9f000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fc723f23000)
```

Previously it was linking with /lib/x86_64-linux-gnu/libncurses.so.6 which was not patched. Now it is linking with /lib/libncurses.so.5 which is patched (strings /lib/libncurses.so.5 | grep auto_getch_refresh shows output).

In fact both /lib/libncurses.so.5 and /lib/libncurses.so.6 are patched. The versions of ncurses in /lib/x86_64-linux-gnu/libncurses* are no longer used and in fact they have the wrong permissions so they really should not be used.

I suspect that things are not quite 100% perfect at this point. The new patching process did not patch the /lib/libncurses.so.5 that we are now using. The new script patched /lib/libncurses.so.6 and /lib/x86_64-linux-gnu/libncurses.so.6. It was the old patching script that patched the v5 libraries. Whatever changed the linking to use the v5 instead of v6 is what really "fixed" things. Perhaps something forced a refresh of the linker cache leading to this change. It bothers me that I don't have the answer yet.

#4 - 02/25/2021 06:17 PM - Greg Shah

Chris: Please try Roger's exact script and the 6.1 patches. His changes do cause the 6.1/6.2 version of the library to be patched. If you then rebuild FWD and it keeps using that 6.x version then it should work. If it instead switches to the v5 version, then you'll need to patch that using the older patch script and the 5.7 patches.

It is not optimal to have to patch both, but until we get the full answer here it should work.

#5 - 02/25/2021 06:17 PM - Greg Shah

- Related to Bug #3298: libp2j build links with ncurses 5.9 when the proper version is 6.0 added

#6 - 02/26/2021 09:11 AM - Chris Weaver

Followed the steps but have different results. If I look at the ldd p2j/build/lib/libp2j.so output:

linux-vdso.so.1 (0x00007fffd8aa0000)
libffi.so.7 => /lib/x86_64-linux-gnu/libffi.so.7 (0x00007fc360480000)
libncurses.so.6 => /lib/x86_64-linux-gnu/libncurses.so.6 (0x00007fc36041e000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fc360410000)
libutil.so.1 => /lib/x86_64-linux-gnu/libutil.so.1 (0x00007fc360400000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc360200000)
/lib64/ld-linux-x86-64.so.2 (0x00007fc3604af000)

What's interesting is /lib/x86_64-linux-gnu/libncurses.so.6 is patched as well as in /lib and libncurses.so.5 is also patched in both places.

strings /lib/x86_64-linux-gnu/libncurses.so.6 | grep auto_getch_refresh
auto_getch_refresh
auto_getch_refresh

strings /lib/x86_64-linux-gnu/libncurses.so.5 | grep auto_getch_refresh
auto_getch_refresh
auto_getch_refresh

strings /lib/libncurses.so.5 | grep auto_getch_refresh
auto_getch_refresh
auto_getch_refresh

strings /lib/libncurses.so.6 | grep auto_getch_refresh
auto_getch_refresh
auto_getch_refresh

#7 - 02/26/2021 09:14 AM - Greg Shah

With everything patched properly, I guess the client is starting now?

#8 - 02/26/2021 09:17 AM - Chris Weaver

Yes, running deploy/server/server.sh starts without undefined symbol: auto_getch_refresh in the server.log file

#9 - 02/26/2021 09:43 AM - Greg Shah

- Status changed from New to Closed

- Assignee set to Greg Shah

- % Done changed from 0 to 100

Alright, I'm closing this.

As an aside, we should probably work #5167 sooner rather than later.

#10 - 02/26/2021 09:44 AM - Greg Shah

- Related to Support #5167: using static linking to eliminate the need to patch the system-wide ncurses added