

Re: why no gcc in 9.1 personal?

Source: <http://linux.derkeiler.com/Newsgroups/alt.os.linux.suse/2004-05/5470.html>

From: Moritz Franosch (mail_at_Franosch.org)

Date: 05/25/04

Date: 25 May 2004 19:34:24 +0200

houghi <houghi@houghi.org.invalid> writes:

> *Moritz Franosch wrote:*

> > *First, a lot of vendors do not open up their hardware.*

>

> *Why do they not do that?*

- They licensed the driver or part of it from someone else.
- They fear that broken drivers could damage the hardware, which may increase the cost for support.
- They want to sell the same hardware with different capabilities, controlled by the driver.
- They have certified hardware/driver combinations, e.g. passive ISDN cards may have to be certified by the telephone company and wireless equipment may have to be certified to fulfill the law. Hardware vendors don't want to be liable when a broken driver damages third party equipment or is used to break the law.
- They fear that competitors use the information they get from an open source driver for competitive products.
- They save the paperwork that would be necessary to explain the interface of the hardware to driver developers.
- They don't want to be liable for wrong descriptions of the hardware.
- When the user installs a new OS, they want to sell new hardware by not porting the driver to the new OS.

> > *Second, Suse does not offer new drivers for download even if they*

> > *are available.*

>

> *SUSE does not have to do that. The hardware vendor can do that.*

The point is: A driver or at least the kernel interface of the driver has to be compiled on the target machine with the very same compiler the kernel was compiled with. Without gcc, a hardware vendor can not install his driver.

> > > *Why is it that you do not need to compile if you have Windows and need*

> > > *specific drivers? Because the manufacturer already compiled them for*

alt.os.linux.suse: Re: why no gcc in 9.1 personal?

> >> you.
> >
> > *No. It's because Windows has a stable binary driver interface and/or*
> > *there are fewer versions of Windows for which the driver has to be*
> > *compiled.*
>
> *Why does Windows has a binary driver? Because the hardware manufacturer*
> *wrote one.*
>
> *Linux has effectively 3 different versions, if you look at kernels. 2.2,*
> *2.4 or 2.6. A driverdeveloper will know how to only make one verion for*
> *all of those.*

No. There is no stable binary driver interface in Linux even during a stable kernel series. This is a design decision.

For example the 4k kernel stack patch may go into 2.6 which would break binary driver compatibility.

http://www.kerneltraffic.org/kernel-traffic/kt20040522_259.html#6
<http://lwn.net/Articles/80290/>

Binary modules and derived works

<http://lwn.net/Articles/62446/>

<http://www.ibrado.com/lkml/#s2-11>

Do I need to test my driver against all distributions?

(REG, MEA) There are minor detail changes in between each kernel version (even in stable series), and depending on what configuration options are used (basically SMP or not), certain things like spinlocks may or may not reserve space in structures, and may or may not need to be called (are even optimized away in non-SMP systems), meaning that a binary driver compiled for SMP might not work with a non-SMP kernel. And vice versa.

Also different vendors tend to inject different things into their kernel patch-sets, which again may subtly change data layouts, etc. In stable kernel series great pains are suffered at maintenance so that data layouts of in-kernel APIs (and API calls themselves) are not changed. Nevertheless something may change making binary drivers to fail in mysterious ways.

Subtle memory changes may appear with i386-PAE mode (large memory machines which can't map all of RAM into the kernel at the same time).

Because of these differences, a driver compiled for one version of the kernel, or one vendor's kernel, is not likely to work with another kernel. Thus, if you are distributing a binary-only driver, you will have a significant support load compiling drivers for different kernels. If you are distributing a driver in source form,

alt.os.linux.suse: Re: why no gcc in 9.1 personal?

then, provided the driver is well-written (i.e. does not make assumptions about byte ordering or word sizes and uses standard kernel interfaces), the driver should be portable across kernel versions and architecture types. It will of course have to be compiled by end-users for their particular kernel. Distribution maintainers are likely to provide pre-compiled drivers, thus most end-users won't need to compile the driver themselves.

Moritz

--

Dipl.-Phys. Moritz Franosch
<http://www.askos.de>