OSDL Open Drivers Summit

A Case Study of Intel's Ethernet Kernel Drivers:

e100, e1000, ixgb and I/OAT DMA

By John A. Ronciak

Agenda

Introduction

Legal Considerations

Program Management Issues with Open Source

Working with the Linux Community

Case studies of the actual drivers

Legal Considerations

First and foremost, I am not a lawyer and I do not speak for Intel Legal!

What we did and do today

- For basic drivers like e100, e1000 and ixgb
- For I/OAT
- Kernel inclusion or not

Customers may not want GPL

- Be prepared for this since you probably will hear it

Legal Considerations cont.

Licensing for your code

- Is GPL the right license?
- Dual license?
  - Know what you get back from the community!
- Not Open Sourcing? Kernel considerations

IP considerations

- Where is your IP?
  - You and only you need to make the call

Program Management Issues with Open Source

Alignment with other drivers

- Windows, FreeBSD, UNIX?
- Common code? Be careful with community patches!!!

Testing?

- Use kernel development methods? i.e. Release early, often and (possible) ugly
- Internal group testing
  - Leads to scheduling issues for the project
  - This is what we do, called a Frequent Release Model
- How to integrate community changes into your source
  - Kernel driver – it's part of the source tree
  - Kernel version and standalone – what we do
    - Need to support older version of kernels – use a kcompat module

Program Management Issues cont.

Integrating with hardware releases

- How to manage release of PCI Device Ids – pci.ids file
  - Driver updates to OSVs – timing is always poor (work is underway however)

Timing of new feature introduction

- Hardware or software?
  - Hardware – see above
  - Software – may need to protect IP for some period of time
- Kernel release model – Hitting the 2 week new feature window
  - Doesn't matter for standalone versions

Case studies of the actual drivers

Base kernel drivers – e100, e1000 and ixgb

What came first from Intel – e1000

- No driver for the 1 gigabit Ethernet controllers
- Better written code for Open Source than anything else that we had
  - Style is important – Follow the kernel coding standard
- License issues to start with – no GPL allowed by Intel back then
  - Used a FreeBSD style license
- Required kernel inclusion from our OEMs
  - Makes it easy for them to include your hardware in their systems/devices
- One of the most requested drivers today
- Highly used since there are so many PRO/1000 controller/NICs out there

Case studies of the actual drivers
cont.

E100 came second

- Highly criticized for poor quality of the code
  - Wasn't kernel compatible
  - Driver was very large in size – both binary and source
- Competing with eepro100 driver
  - eepro100 was poorly designed
    - Some problems couldn't be fixed
  - Couldn't add everything we needed to – mostly having to do with offloads
  - Major push back from community to fix "their" driver
- Help from OEMs for distribution inclusion of e100 before kernel inclusion
- Finally agreement for the kernel to supply both drivers

Case studies of the actual drivers
cont.

Next came ixgb (10 Gigabit Ethernet)

- First 10 gig driver included in the kernel
- Very close to the e1000 driver as far as coding
- Early systems had resource problems
  - This HW could bring a system to it's knees with ease
  - Driver used to find stack problems that were never seen before
- Showed lots of performance issues with both the driver and stack
  - Also showed some system limitation including PCI configuration space problems

Case studies of the actual drivers
cont.

Then the e100 rewrite

- Reasons include:
  - Widely used driver (PRO/100 parts are very common)
  - Old driver was large and very complicated
  - Lack of continued maintenance due to resource constraints at the time
  - Reduced the feature set in the process – some offloads removed like checksumming
- Wanted to get to one driver for the PRO/100 parts in the kernel
- After a number of kernel releases with the new e100 the eepro100 driver was removed
- It's now much faster and more maintainable than previous driver
- Community feels they can support it if they need to at this point
  - Documentation has been released for it

Case studies of the actual drivers
cont.

License change at Intel

- Intel did not support GPL early on
- Needed for kernel inclusion
- Corporate policy changed which allowed us to release under the GPL
- Now recommend dual license
  - Accepted by the Linux kernel
  - Still need to understand how you get patches from the community
    - Need explicit license statement for patches

Hardware document release for devices

- Needed for future support if you/your company stops support for some reason

Working with the Linux Community

All work is done through mail lists

- Lists are found on the kernel.org website
  - Sign-up (subscribe) from there as well
  - Networking is done on “netdev”, kernel work on “lkml”
  - Lot of others, take a look

Show me the code!

- It is always better to supply code to make your point, always!
- Few get away with not doing this

Working with the Linux Community cont.

Keep posts short and to the point

- Stay on topic for any given thread
- Try not to fracture the mail threads to side topics – hard to follow later
- Only use text posts – very important, some reject all mail that is html formatted
- Don't top post if at all possible
- If supplying charts, also supply the data – use an open format for graphics

Don't be “thin skinned”

- The lists are often very blunt and offending to some – It's not personal!
- Some are not very nice at all but everybody needs to live with them!

Working with the Linux Community cont.

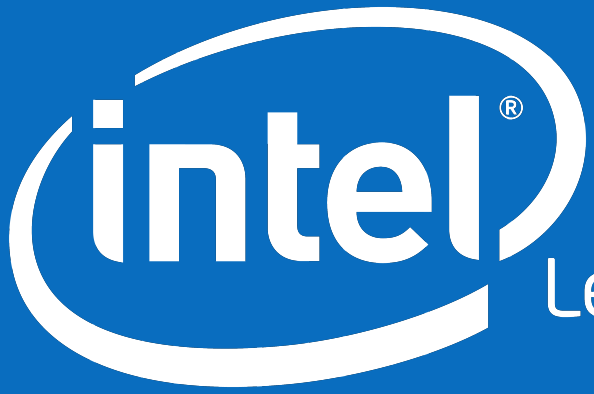
Make use of Archives

- Lots of them but http://marc.theaimsgroup.com/ is a good one
- Research your topic before posting or you may not get a response
- Get to know the active posters so you know what to expect

Questions?

Contact info:

john.ronciak@intel.com


intel
Leap ahead™