

Linux device driver staging

Learning from the sins of others

Stephen Hemminger

stephen.hemminger@vyatta.com

Japan Linux Symposium

22 October 2009

Linux driver staging

- What is it?
- Why are drivers not in mainline?
- How to avoid these problems

Staging drivers

Version	Drivers		Lines Of Code
2.6.28	11		99,293
2.6.29	26	+15	215,082
2.6.30	37	+11	277,420
2.6.31	42	+7 -2	354,752
2.6.32-rc4	45	+11 -10	365,073
Linux-next	47	+2 -1	352,751

WELCOME TO PURGATORY

HEAVEN  **3**

HELL  **666**

Staging tree maintainer



Seven deadly sins

- Style
- Excess
- Reinvention
- Bad Interface
- Incompatibility
- No hardware
- No interest

Sin #1: Style mismatch



Microsoft Hyper V driver (before)

```
/*++
Name:     netvsc_open()
Desc:     Open the specified interface device
--*/
static int netvsc_open(struct net_device *net)
{
    int ret=0;
    NETVSC_DRIVER_OBJECT *net_drv_obj = &net_drv_ctx->drv_obj;
    DEVICE_OBJECT *device_obj = &net_device_ctx->device_ctx->device_obj;

    if (netif_carrier_ok(net))
    {
        // Open up the device
        ret = net_drv_obj->Open(device_obj);
        if (ret != 0)
        {
            DPRINT_ERR(NETVSC_DRV, "unable to open device (ret %d).", ret);
            return ret;
        }

        netif_start_queue(net);
    }
    else
```


Misuse of typedef's

```
/* Counters for error rate monitoring */  
typedef struct _MP_ERR_COUNTERS {  
    u32 PktCountTxPackets;  
    u32 PktCountTxErrors;  
    u32 TimerBasedTxErrors;  
    u32 PktCountLastError;  
    u32 ErredConsecPackets;  
} MP_ERR_COUNTERS, *PMP_ERR_COUNTERS;
```



GCC

Sparse

Checkpoint

Sin #2: Unwarranted Complexity

eschew obfuscation, espouse elucidation

Fresh Fish Sold Here



篠一

八八五一室 八八二
大篠 八十四棟
岡野水産 大間 助康
やまふ水産
の運行及び船・停止禁止時間 小車 7:00-10:00 アット 6:30-11:00

やまふ水産

大間

康

北條たこほけ通り

松日海水産株式会社

3升 8升

大 30

```
//-----  
// Write Data to Driver  
//-----  
// -> write(...)  
//-----
```

```
static ssize_t EplLinWrite(struct file *pInstance_p,  
                           const char *pSrcBuff_p,  
                           size_t BuffSize_p,  
                           loff_t * pFileOffs_p)  
{  
  
    int iRet;  
  
    TRACE0("EPL: + EplLinWrite...\n");  
  
    TRACE0("EPL: Sorry, this operation isn't supported.\n");  
    iRet = -EINVAL;  
  
    TRACE1("EPL: - EplLinWrite (iRet=%d)\n", iRet);  
    return (iRet);  
}
```

Sin #3: Reinventing



```
// we assume the s1 and s2 both are strings.
BOOLEAN rtstrcasecmp(char *s1, char *s2)
{
    char *p1 = s1, *p2 = s2;

    if (strlen(s1) != strlen(s2))
        return FALSE;

    while(*p1 != '\0')
    {
        if((*p1 != *p2) &&
            ((*p1 ^ *p2) != 0x20))
            return FALSE;

        p1++;
        p2++;
    }

    return TRUE;
}
```

SMP red flags

- `atomic_t`
- Volatile
- Locking wrappers
- Too many locks

Sin #4: Wrong Interface



```
int heci_ioctl_get_version(struct iamt_heci_device *dev, int if_num,
                          struct heci_message_data __user *u_msg,
                          struct heci_message_data k_msg,
                          struct heci_file_private *file_ext)
{
    if ((if_num != HECI_MINOR_NUMBER) || (!dev)
        || (!file_ext))
        return -ENODEV;

    if (k_msg.size < (sizeof(struct heci_driver_version) - 2))
        return -EMSGSIZE;

    res_msg.data = kmalloc(sizeof(struct heci_driver_version), GFP_KERNEL);
    if (!res_msg.data)
        return -ENOMEM;

    version = (struct heci_driver_version *) res_msg.data;
    version->major = MAJOR_VERSION;
    version->minor = MINOR_VERSION;
    version->hotfix = QUICK_FIX_NUMBER;
    version->build = VER_BUILD;

```

...



Sin #5: Clashing with existing code

Existing 8139too driver

```
static struct pci_device_id rtl8139_pci_tbl[] = {  
    {0x10ec, 0x8139, PCI_ANY_ID, PCI_ANY_ID, 0, 0, RTL8139 },
```

...

```
MODULE_DEVICE_TABLE (pci, rtl8139_pci_tbl);
```

Overlapping table in Epl/Edrvr8139.c

```
static struct pci_device_id aEdrvPciTbl[] = {  
    {0x10ec, 0x8139, PCI_ANY_ID, PCI_ANY_ID, 0, 0, 0},  
    {0,}  
};
```

```
MODULE_DEVICE_TABLE(pci, aEdrvPciTbl);
```

Sin #6

-ENO HARDWARE



Sin#7: No Interest



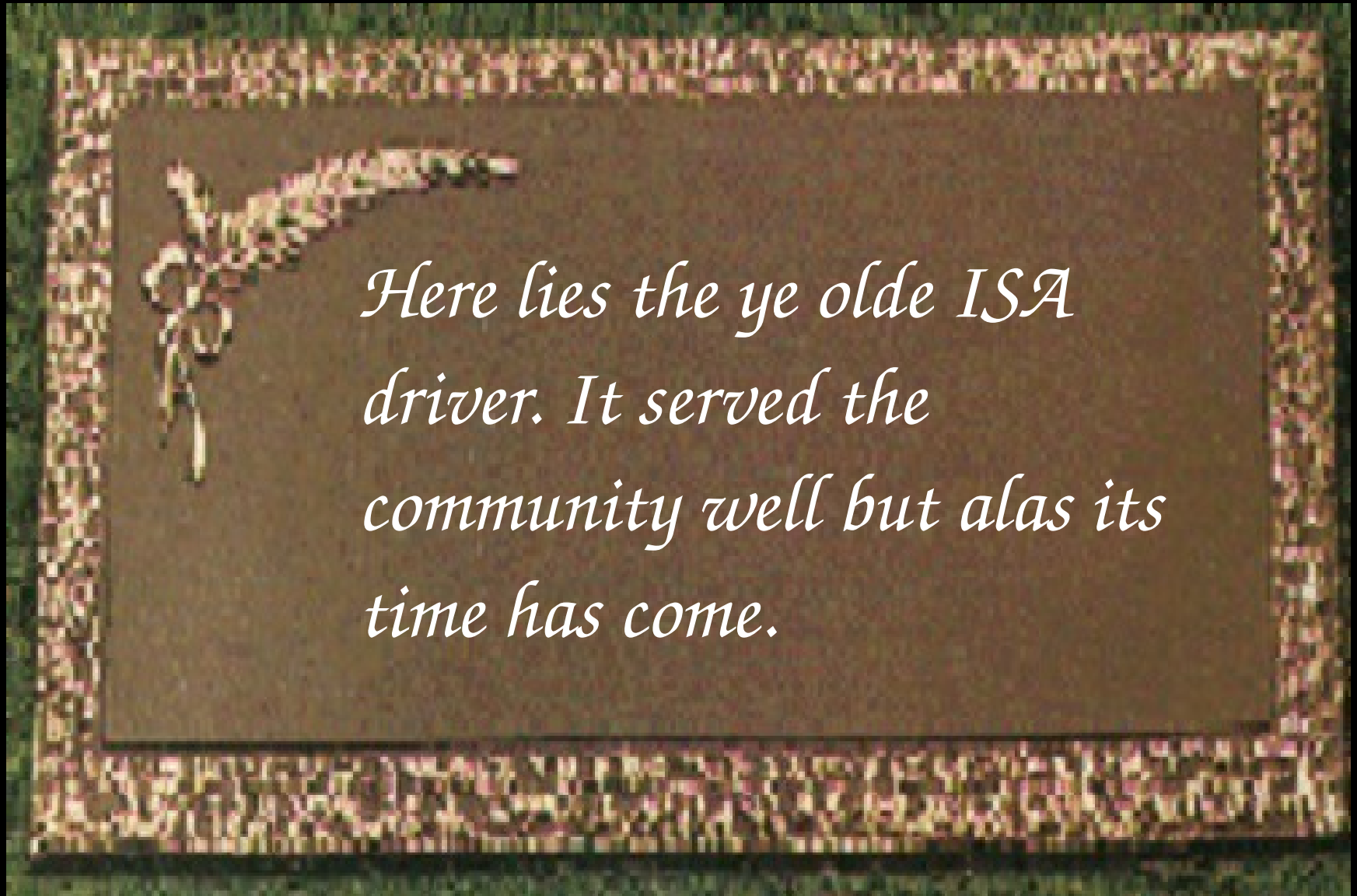
Seven deadly sins

- Style
- Excess
- Reinvention
- Bad Interface
- Incompatibility
- No hardware
- No interest

Work in progress



Next mission



*Here lies the ye olde ISA
driver. It served the
community well but alas its
time has come.*