Лекция 5.2 Использование sysfs в драйверах устройств

Разработал: Максимов А.Н.

Содержание

- Общая информация по sysfs
- Атрибуты устройства
- Атрибуты драйвера
- Добавление драйвера в ядро

sysfs

Sysfs — виртуальная файловая система располагающаяся в памяти.

Основное назначение: обеспечивать явное предоставление инфомации о структурах ядра (шины, драйвера, устройства и т.д.) и их атрибутах в пользовательском пространстве.

Директории в в sysfs соответсвуют структура kobject в модели устройств.

Дополнительная информация может быть найдена в Documentation/sysfs.txt

Атрибуты

Атрибуты предстявляются в sysfs в виде обычных текстовых ASCII файлов.

Sysfs перенаправляет операции ввода/вывода над атрибутами и позволяет ядру читать и писать атрибуты.

Определиени атрибута:

```
struct attribute {
        char                    * name;
        struct module           *owner;
        mode_t                  mode;
};
```

Функции создания/удаления атрибута:

```
int sysfs_create_file(struct kobject * kobj, const struct attribute * attr);
void sysfs_remove_file(struct kobject * kobj, const struct attribute * attr);
```

(для абстрактоного атрибута функции чтения/записи не определены)

Атрибуты для устройства(device)

Возможно добавление атрибутов для объектов модели устройств типа device

Тип данных для атрибута устройства:

```
struct device_attribute {
   struct attribute   attr;
   ssize_t (*show)(struct device *dev, struct device_attribute *attr, char *buf);
   ssize_t (*store)(struct device *dev, struct device_attribute *attr, const char *buf,
     size_t count);
};
```

Функции для создания:

```
int device_create_file(struct device *, const struct device_attribute *);
void device_remove_file(struct device *, const struct device_attribute *);
```

Определенов в include/linux/device.h

Атрибуты для устройства(device). п.2

Для облегчения создания утрибутов устойств определен макрос:

```
#define DEVICE_ATTR(_name, _mode, _show, _store) \
struct device_attribute dev_attr_##_name = __ATTR(_name, _mode, _show,
    _store)
```

Пример.

```
static DEVICE_ATTR(foo, S_IWUSR | S_IRUGO, show_foo, store_foo);
```

Это эквивалентно:

```
static struct device_attribute dev_attr_foo = {
    .attr    = {
        .name = "foo",
        .mode = S_IWUSR | S_IRUGO,
        .show = show_foo,
        .store = store_foo,
    },
};
```

Пример.

```
// Определение функций чтения и записи параметров
int _port_reg;
static ssize_t foo_port_reg_show(struct device *dev,    struct device_attribute *attr, char *buf){
    return sprintf(buf, "%x\n", _port_reg);
}
static ssize_t foo_port_reg_set(struct device *dev,    struct device_attribute *attr,const char *buf, size_t
       count) {
    _port_reg=simple_strtoul(buf, NULL, 0);
    return count;
}
static DEVICE_ATTR(port_reg,0644,foo_port_reg_show,foo_port_reg_set);
int rtl8139_probe (struct pci_dev *pdev, const struct pci_device_id *id) {
   res = device_create_file(&pdev->dev,&dev_attr_port_reg);
  return 0;
}
```

simple_strtoul

simple_strtoul — преобразует string в unsigned long

Определение функции:

unsigned long simple_strtoul (const char *cp, char **endp, unsigned int base);

Аргументы:

cp - начало строки

endp - указатель на конец разбираемой строки

base - база числа

Атрибуты для драйвера устройства.

Структура для представления атрибута:

```
struct driver_attribute {
        struct attribute        attr;
        ssize_t (*show)(struct device_driver *, char * buf);
        ssize_t (*store)(struct device_driver *, const char * buf,
                        size_t count);
};
```

Определение:

DRIVER_ATTR(_name, _mode, _show, _store)

Создание/уничтожение:

int driver_create_file(struct device_driver *, const struct driver_attribute *);
void driver_remove_file(struct device_driver *, const struct driver_attribute *);

Определено в include/linux/device.h

Практическое задание

Добавить в драйвер последовательного порта возможность читать статистику по число переданных и принятых байт через sysfs, а также возможность сброса статистики.

Практическое задание

Добавить в разработанный ранее драйвер атрибут для задания размера области памяти отображаемой через proc

Часть 2. Добавление драйвера в ядро.

Для добавления драйвера в ядро необходимо:

- Поместить драйвер сетевого интерфейса в католок drivers\net\bhnet
- Создать файл drivers\net\bhnet\Makefile
- Изменить файл drivers\net\Kconfig
- drivers\net\Makefile

drivers\net\bhnet\Makefile

```
#############################################################################
# Makefile for the BHNET foo driver net driver
#
obj-$(CONFIG_FOO_BHNET) += bhnet.o
```

Kconfig и Makefile

Изменения в Kconfig

config **FOO_BHNET**

tristate "Add support dor my foo black hole network driver"

depends on NET_PCI && PCI

---help---

  This is a driver for foo network device

  If you have one of those, say Y and

Изменения в Makefile

obj-$(CONFIG_FOO_BHNET) += bhnet/

Пересборка ядра с новым драйвером

Необходимо пересобрать ядро:

make menuconfig

make

Вопросы