Writing Device Drivers Under Linux

Network Device Drivers

Network buffer management

- BSD uses mbuf to store network packets
  - mbufs are changed together to minimize copying
- Linux uses sk_buffs
  - Linear buffer
  - Header space is preallocated
  - Makes many operations much faster

sk_buff allocation routines

- alloc_skb(size, priority)
- skb_reserve(skbuff, size)
- skb_push(skbuff, size)
- skb_pull(skbuff, size)

January 7, 1997 Network Device Drivers 3

alloc_skb()

- Allocates an sk_buff structure
- Data area can hold *size* bytes
- Initially, all the data is "tail" space
- skb->data points at beginning of data area

January 7, 1997 Network Device Drivers 4

skb_reserve()

- Reserves *size* bytes of space for header
- Must be called before adding data to the sk_buff
- Moves skb->data forward *size* bytes
- Increases the amount of "head" space

skb_put()

- Used to append data to the end of a packet
- Returns pointer to newly allocated space
- Reduces the amount of "tail" space by *size* bytes
- Increases the packet size by *size* bytes

skb_push()

- Used to prepend data to a packet
- Returns pointer to newly allocated space
  - (which is always skb->data)
- Reduces the amount of "head" space

sk_buff examples

sk_buff after alloc_skb()

sk_buff after skb_reserve()

More sk_buff examples

Head Room	Data	Tail Room

sk_buff containing data

Head Room	Data	skb_put area	Tail Room

After sk_put() has been called on the buffer

Head Room	skb_push area	Data	Tail Room

After sk_push() has been called

Ethernet performance hack

- Ethernet header is 14 bytes
- Better performance on many architectures when IP header is 16-byte aligned
- Solution
  - Allocate an sk_buff 2 bytes larger than the packet requires
  - Call skb_reserve(skb, 2) to reserve 2 bytes

Example source code

- From drivers/net/skeleton.c (see page 122)
- Designed to make it easy to write a new network device driver
- Written by Donald Becker, who wrote most of the network drivers
- Copyright by the NSA: fun, fun

Parts of a network driver

- Probe code
- Open/init code
- Transmit function
- Interrupt/receive functions
- Set multicast list

Probe code

- See netcard_probe() (skeleton.c:129) and netcard1_probe() (skeleton.c:156)
- Hardware/irq detection
- Read Ethernet address from EEPROM

Open/init code

- See net_open() (skeleton.c:317)
- Set Ethernet address
- Other card-specific initialization

Transmit function

- See net_send_packet() (skeleton.c:356)
- Normal processing
  - Send packet
  - call dev_kfree_skb()
  - Return 0
- If temporarily busy
  - Set dev->tbusy non-zero
  - Return 1

Interrupt/receive function

- See net_interrupt (skeleton.c:416)
- Checks device for TX and RX interrupt flags
- See net_rx() (skeleton.c:455)
- Gets packet from hardware and calls netif_rx() to process it

Set multicast list

- See set_multicast_list (skeleton.c:567)
- Used to set the ethernet card
  - to listen to multicast transmissions
  - to enter promiscuous mode