IN 3230 / IN 4230
Oracle Session – Week 2 – Raw Sockets

Praveensankar Manimaran

09/09/2021

Outline

1. Raw Socket Flow
2. Different Components needed
    1. Packet socket
    2. sockaddr_ll
    3. Ifaddrs
    4. iovec
    5. ether_frame
    6. msghdr

1 . Raw Socket Overview

Steps

1. Create Packet Socket (AF_PACKET)
2. Create physical layer address (sockaddr_ll)
3. Construct message header (msghdr)
    1. Construct ethernet frame (ether_frame)
    2. Point iovec to the ethernet frame (iovec)
    3. Point iovec to the data
    4. Add physical layer address

2 . 1 Packet Socket (AF_PACKET)

```
#include <sys/socket.h>
#include <linux/if_packet.h>
#include <net/ethernet.h> /* the L2 protocols */

packet_socket = socket(AF_PACKET, int socket_type, int protocol);
```

1. **AF_PACKET:**
   - Also called as "packet socket".
   - used to receive or send raw packets at the device driver
2. **SOCK_RAW:**
   - for raw packets

[1] https://man7.org/linux/man-pages/man7/packet.7.html

2 . 2 Sockaddr_ll

```
struct sockaddr_ll {
unsigned short sll_family;   /* Always AF_PACKET */
unsigned short sll_protocol; /* Physical-layer protocol */
int  sll_ifindex;  /* Interface number */
unsigned short sll_hatype;   /* ARP hardware type */
unsigned char  sll_pkttype;  /* Packet type */
unsigned char  sll_halen;    /* Length of address */
unsigned char  sll_addr[8];  /* Physical-layer address */
};
```

[1] https://man7.org/linux/man-pages/man7/packet.7.html

[2] https://github.com/spotify/linux/blob/master/include/linux/if_ether.h

2 . 3 ifaddrs

```
struct ifaddrs {
    struct ifaddrs  *ifa_next;    /* Next item in list */
    char            *ifa_name;    /* Name of interface */
    unsigned int     ifa_flags;   /* Flags from SIOCGIFFLAGS */
    struct sockaddr *ifa_addr;    /* Address of interface */
    struct sockaddr *ifa_netmask; /* Netmask of interface */
    union {
        struct sockaddr *ifu_broadaddr;
                         /* Broadcast address of interface */
        struct sockaddr *ifu_dstaddr;
                         /* Point-to-point destination address */
    } ifa_ifu;
#define              ifa_broadaddr ifa_ifu.ifu_broadaddr
#define              ifa_dstaddr   ifa_ifu.ifu_dstaddr
    void            *ifa_data;    /* Address-specific data */
};
```

- creates a linked list of structures describing the network interfaces of the local system, and stores the address of the first item of the list in *ifap.
- Important fields: ifa_next, ifa_name and ifa_addr

[2] https://man7.org/linux/man-pages/man3/getifaddrs.3.html

2 . 4 iovec

- struct iovec {
  ptr_t iov_base; /* Starting address */
  size_t iov_len; /* Length in bytes */
  };

- iov_base – stores the starting address
- Iov_len – stores the length

Why it's needed ?

- It's used by the message header (msghdr)
- To point to ethernet frame header and the payload

2 . 5 ether_frame

```
struct ether_frame {
uint8_t dst_addr[6];
 uint8_t src_addr[6];
 uint8_t eth_proto[2];
 uint8_t contents[0]; } __attribute__((packed));
```

- It's frame header we use in the raw socket
- **__attribute__ ((packed)):** it packs the ether_frame in such a way that it preserves memory (removes automatic padding between the structure members)

2 . 6 msghdr

```
void          *msg_name        optional address
socklen_t      msg_namelen     size of address
struct iovec  *msg_iov         scatter/gather array
int            msg_iovlen      members in msg_iov
void          *msg_control     ancillary data, see below
socklen_t      msg_controllen  ancillary data buffer len
int            msg_flags       flags on received message
```

- **Msg_name :** sockaddr pointer will be used here (struct sockaddr_ll *)
- **Msg_namelen:** size of sockaddr pointer
- **Msg_iov:** array of io vector structures (we use 2 arrays. One to point to the frame header and the 2nd one to point to the frame payload)
- **Msg_iovlen:** no of numbers in msg_iov (2 in our example)

References

1. https://man7.org/linux/man-pages/man7/packet.7.html
2. https://man7.org/linux/man-pages/man3/getifaddrs.3.html
3. http://www.ccplusplus.com/2012/02/struct-iovec-iov.html
4. msghdr : https://pubs.opengroup.org/onlinepubs/7908799/xns/syssocket.h.html

Thank You