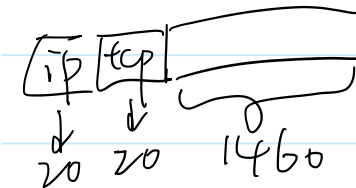


# tcp段

2023年5月6日

9:23

16bit 源端口号								16bit 目的端口号									
32bit 序号																SEQ	
32bit 确认序号																ACK	
4bit首部长度		6bit 保留		U R G	A C K	P S H	R S T	S Y N	F I N	16bit 窗口大小							
16bit 校验和										16bit 紧急指针							
选项																	
数据																	



MSS = Max Segment Size

MTU - 40

↓  
不触发分片

# udp

2023年5月6日

10:28

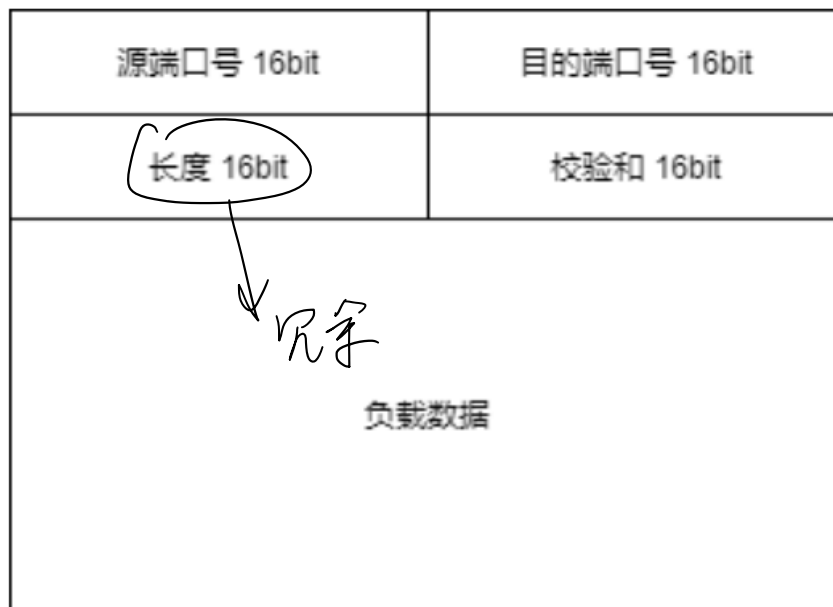
无连接 不可靠

UDP 面向消息

数据有边界

TCP 面向数据流

数据无边界



## KCP - 快速可靠网络传输协议

KCP 是一个快速可靠协议，能以比 TCP 浪费 10%-20% 的带宽的代价，换取平均延迟降低 30%-40%，且最大延迟降低三倍的传输效果。纯算法实现，并不负责底层协议（如 UDP）的...

软件类型：协议和规范 | 授权协议：GPLv2 | 开发语言：C/C++

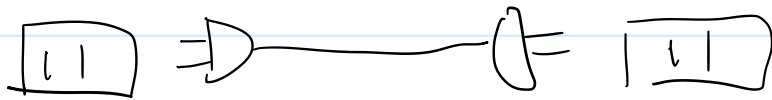
[相关博客](#) [软件文档](#)

<https://www.oschina.net/p/kcp?hmsr=aladdin1e1>

# 网络编程

2023年5月6日 10:56

Berkley Socket 伯克利套接字



地址. ip = port

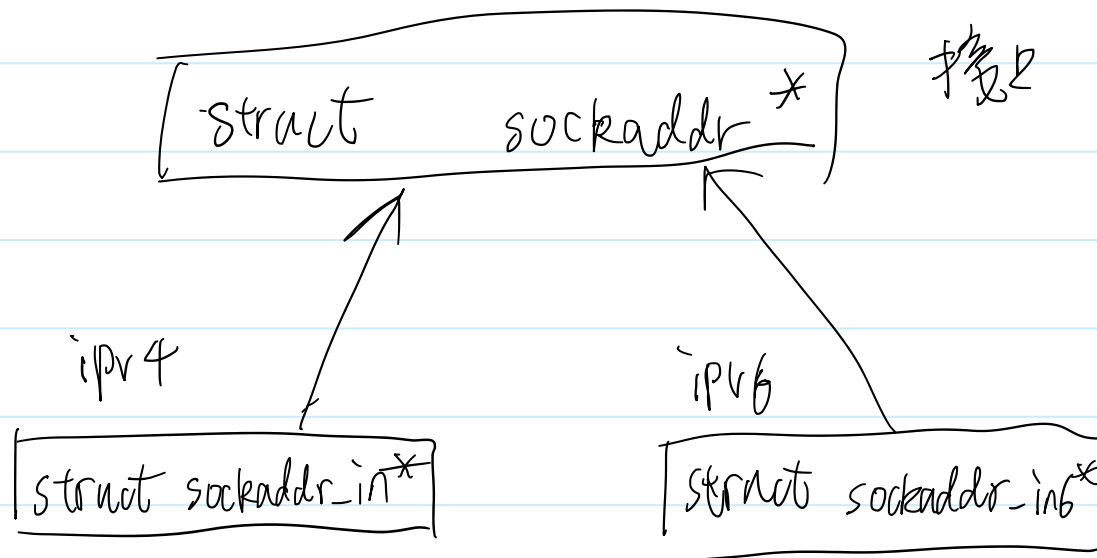
```
struct sockaddr_in {  
    sa_family_t    sin_family; /* address family: AF_INET */  
    in_port_t      sin_port;   /* port in network byte order */  
    struct in_addr sin_addr;    /* internet address */  
};  
  
/* Internet address. */  
struct in_addr {  
    uint32_t        s_addr;     /* address in network byte order */  
};
```

↓  
ipv4 32bit

# 不同的版本ip设计不一样

2023年5月6日 11:04

```
struct sockaddr_in6 {  
    sa_family_t    sin6_family; /* AF_INET6 */  
    in_port_t      sin6_port;   /* port number */  
    uint32_t       sin6_flowinfo; /* IPv6 flow information */  
    struct in6_addr sin6_addr;   /* IPv6 address */  
    uint32_t       sin6_scope_id; /* Scope ID (new in 2.4) */  
};  
  
struct in6_addr {  
    unsigned char  s6_addr[16]; /* IPv6 address */  
};
```



# 真值 机器数 内存布局

2023年5月6日

11:09

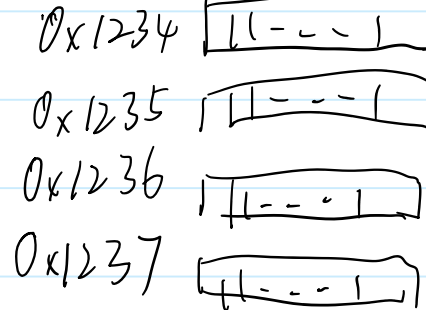
人类视角

int i = -1

用0/1描述

11...1 11...1 11...1 11...1  
△

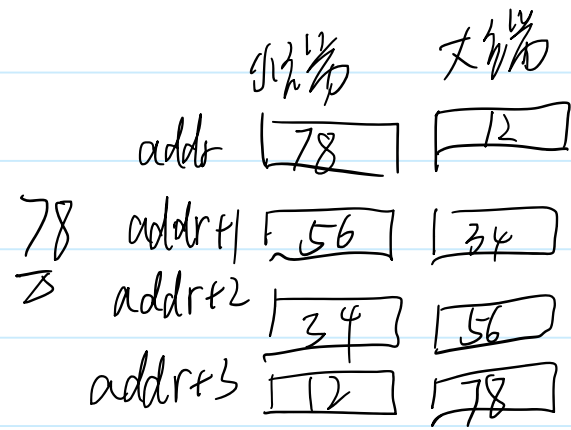
1 Byte / 8 bit



字节序. 机器数 → 内存  
(大小端序)

→ 低位低地址 高位高地址 小端  
低位高地址 高位低地址 大端

机器数 0x 12 34 56 78



# 主机字节序 网络字节序

2023年5月6日

11:19

↓  
大端法 (网络设备...)

↓  
x86 小端法

ARM 可大可小.

int i = 0x12345678

主机字节序

0\_endian.c

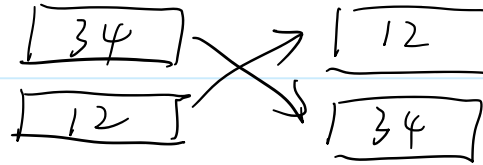
```
1 #include <stdio.h>
2 int main(){
3     int i = 0x12345678;
4     printf("i = %x\n", i);
5     char * p = (char *)&i;
6     printf("*p = %x\n", *p);
7 }
```

# 字节序转换

2023年5月6日 11:27

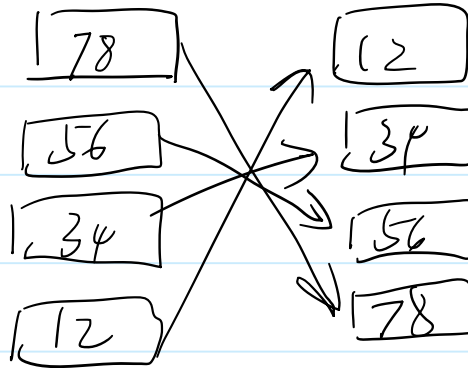
16bit

0x 12 34



32bit

0x 12 34 56 78



h host 主机

n net 网络

long 32 bit

short 16 bit

```
uint32_t htonl(uint32_t hostlong);
```

```
uint16_t htons(uint16_t hostshort);
```

```
uint32_t ntohl(uint32_t netlong);
```

```
uint16_t ntohs(uint16_t netshort);
```

# ip的转换

man inet

2023年5月6日 11:31

ip "192.168.118.128"  $\longrightarrow$  4byte 大端 ip地址  
字符串 点分十进制

`int inet_aton(const char *cp, struct in_addr *inp);`  
`in_addr_t inet_addr(const char *cp);` } 点分十进制  $\longrightarrow$  32bit大端

`char *inet_ntoa(struct in_addr in);`  $\longrightarrow$  32bit大端  $\longrightarrow$  点分十进制

```
int main(int argc, char *argv[]){
    // ./1_addr 192.168.118.128 1234
    ARGS_CHECK(argc,3);
    struct sockaddr_in addr;
    addr.sin_family = AF_INET; //ipv4
    addr.sin_port = htons(atoi(argv[2]));
    // argv[2] char * --> short 再从小端h 到大端n
    // inet_aton
    // inet_aton(argv[1], &addr.sin_addr);
    // inet_addr
    addr.sin_addr.s_addr = inet_addr(argv[1]);
}
```



# 域名

2023年5月6日 11:47

www.baidu.com

14.215.177.39

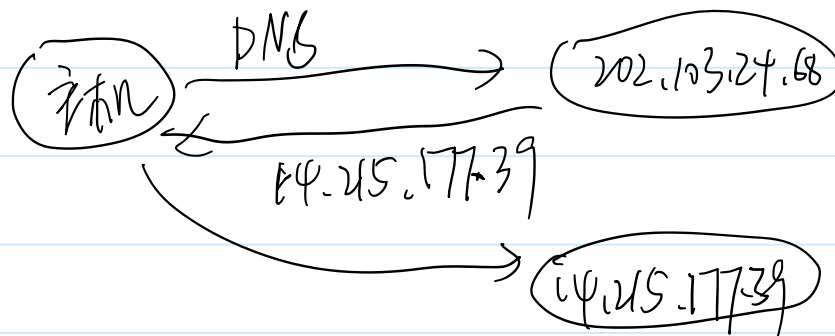
① 查 hosts 文件.

```
port = 8080, ip = 0.0.0.0
[liao@ubuntu Linuxday 20]$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    ubuntu
140.82.112.3 github.com
```

```
# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

② DNS 协议

IPv4 DNS 服务器	202.103.24.68
	202.103.44.150



# nslookup

2023年5月6日 11:52

```
[liao@ubuntu Linuxday 20]$ nslookup www.baidu.com
```

```
Server:          127.0.0.53
```

```
Address:         127.0.0.53#53
```

```
Non-authoritative answer:
```

```
www.baidu.com    canonical name = www.a.shifen.com.
```

```
Name:   www.a.shifen.com
```

```
Address: 14.119.104.189
```

```
Name:   www.a.shifen.com
```

```
Address: 14.119.104.254
```

```
[liao@ubuntu Linuxday 20]$ nslookup www.bilibili.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
www.bilibili.com   canonical name = a.w.bilicdn1.com.
Name:   a.w.bilicdn1.com
Address: 183.131.147.27
Name:   a.w.bilicdn1.com
Address: 116.207.137.68
Name:   a.w.bilicdn1.com
Address: 171.214.10.140
Name:   a.w.bilicdn1.com
Address: 117.21.179.19
Name:   a.w.bilicdn1.com
Address: 171.214.10.141
Name:   a.w.bilicdn1.com
Address: 183.131.147.48
```

# 域名转换的系统调用

2023年5月6日 11:53

```
struct hostent *gethostbyname(const char *name);
```

底层使用了 DNS，断网不可用。