

计算机网络编程

第11章 TCP数据包的封装与发送

信息工程学院 方徽星

fanghuixing@hotmail.com

大纲

- 设计目的
- 相关知识
- 例题分析

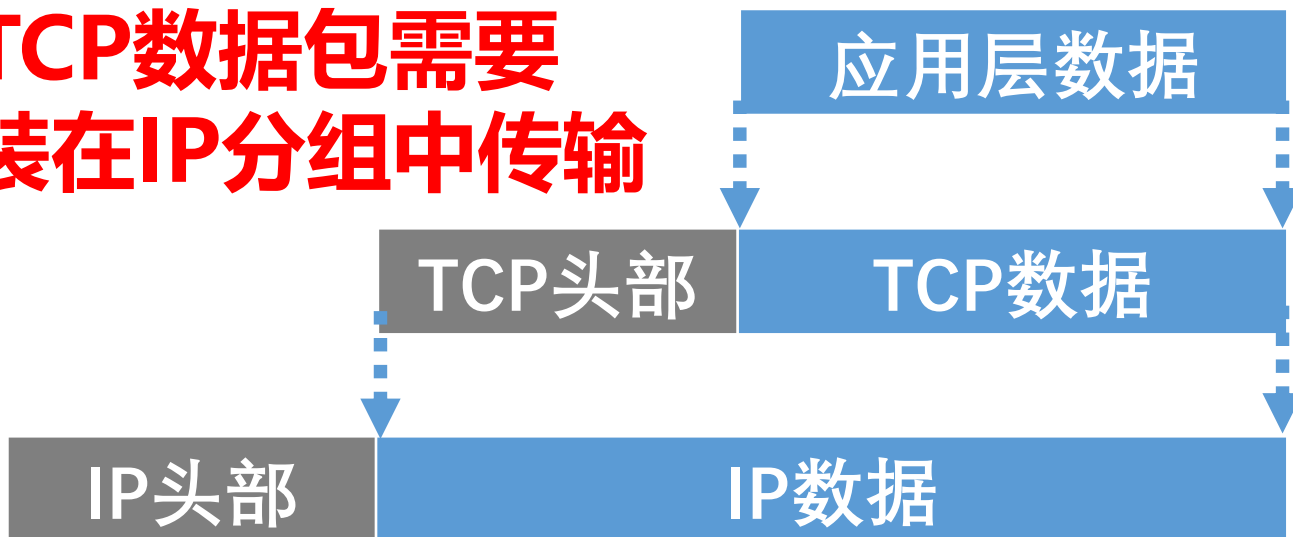
1. 设计目的

- 熟悉TCP包结构对于理解网络层次结构，以及TCP协议与IP协议的关系有着重要意义
- 通过封装与发送一个标准的TCP数据包，了解TCP包结构中各字段的含义与用途
- 深入理解传输层与下面各层的关系

2. 相关知识：TCP协议的基本概念

- 传输层协议分为
 - TCP：可靠、面向连接
 - UDP：不可靠、无连接

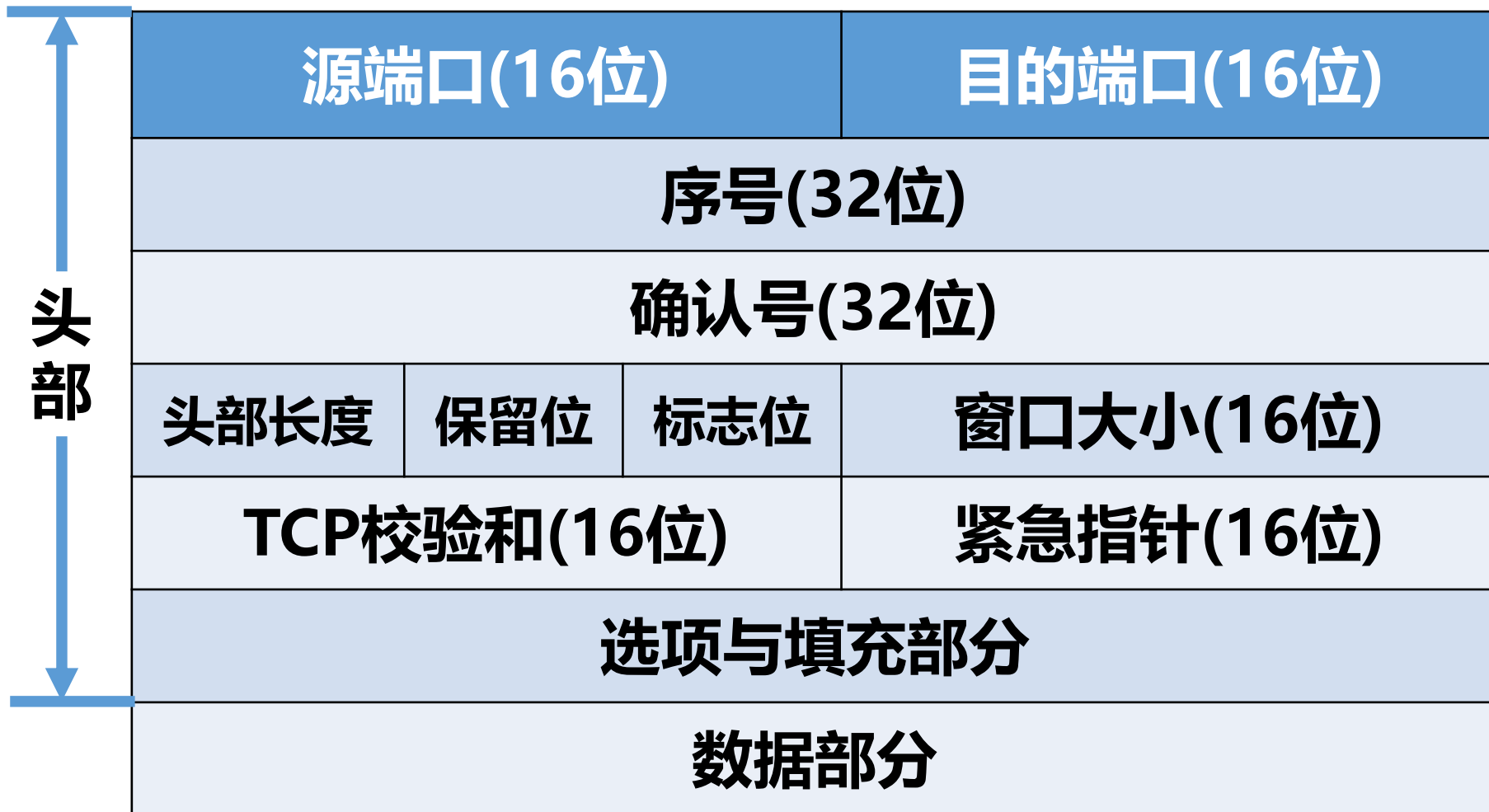
TCP数据包需要封装在IP分组中传输



2. 相关知识：TCP协议的基本概念

- TCP协议允许通信双方的应用程序在任何时候传输数据
- 通信双方都设置有相应的发送与接收缓冲区，用于缓存数据流
- TCP协议使用以字节为单位的滑动窗口机制，用于控制字节流的发送、接收、确认与重传过程

2. 相关知识：TCP数据包的结构



2. 相关知识：TCP数据包的结构

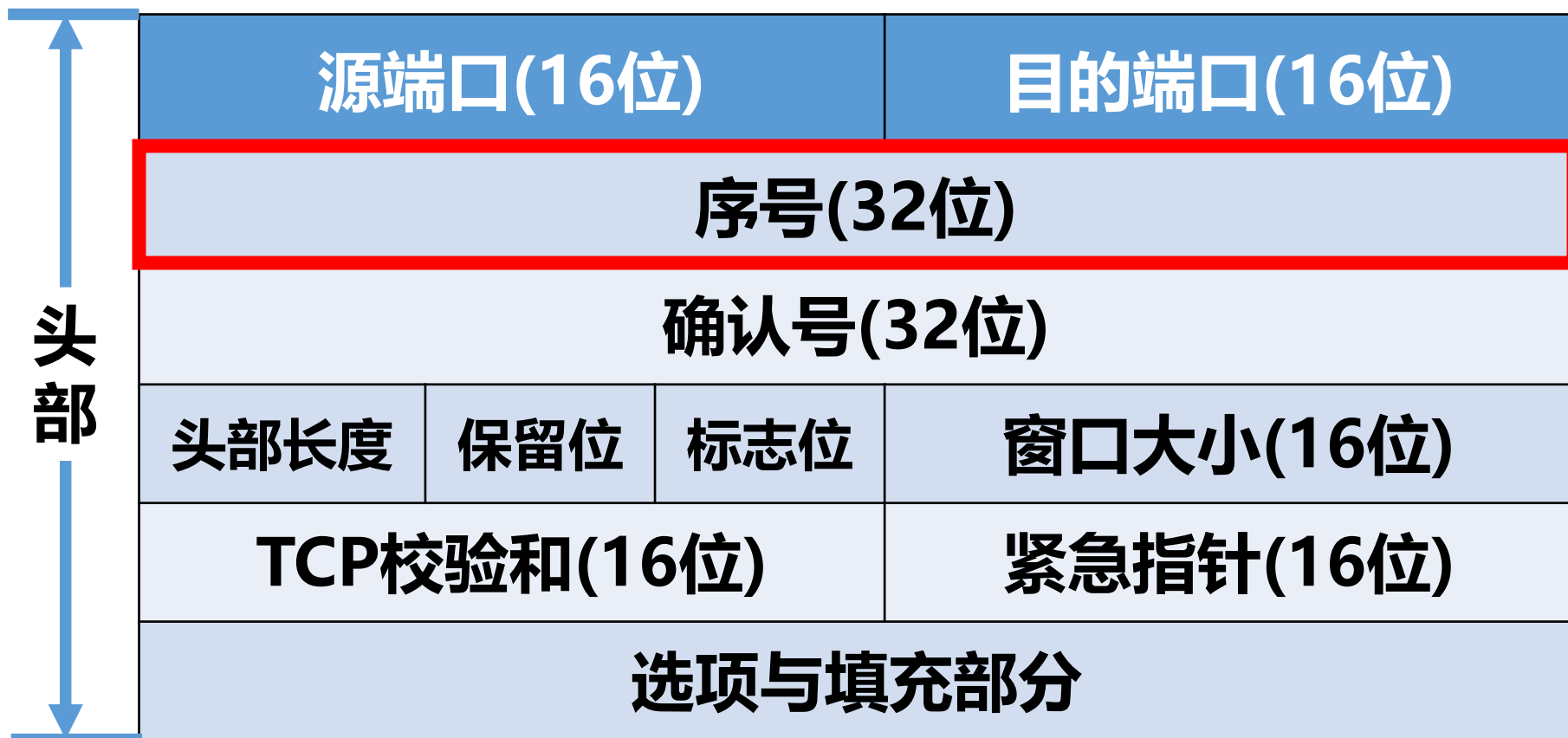
源端口：发送方应用程序使用的TCP端口号

目的端口：接收方应用程序使用的TCP端口号



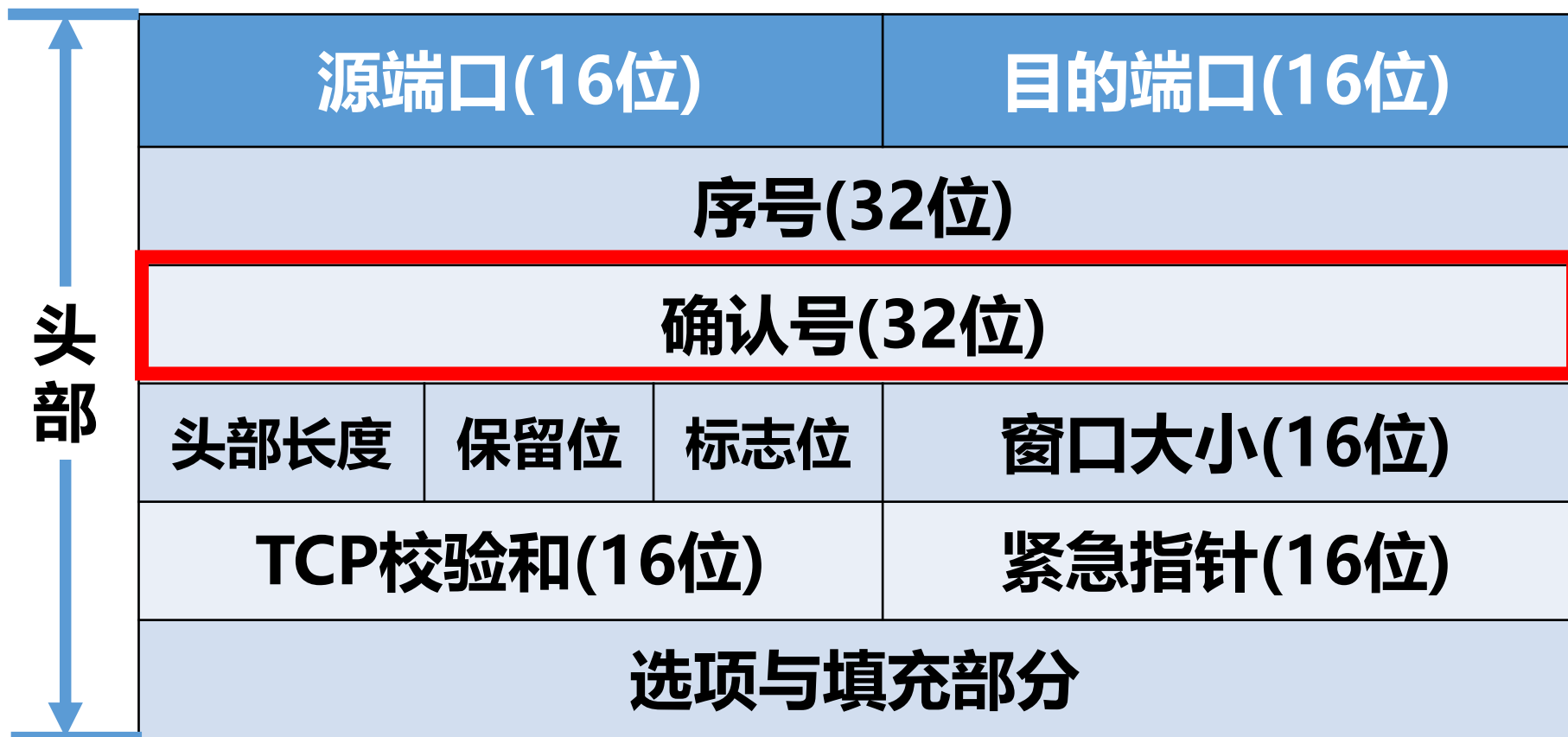
2. 相关知识：TCP数据包的结构

序号：TCP包第一字节序号。TCP 连接中传送的数据流中的每一个字节都编上一个序号



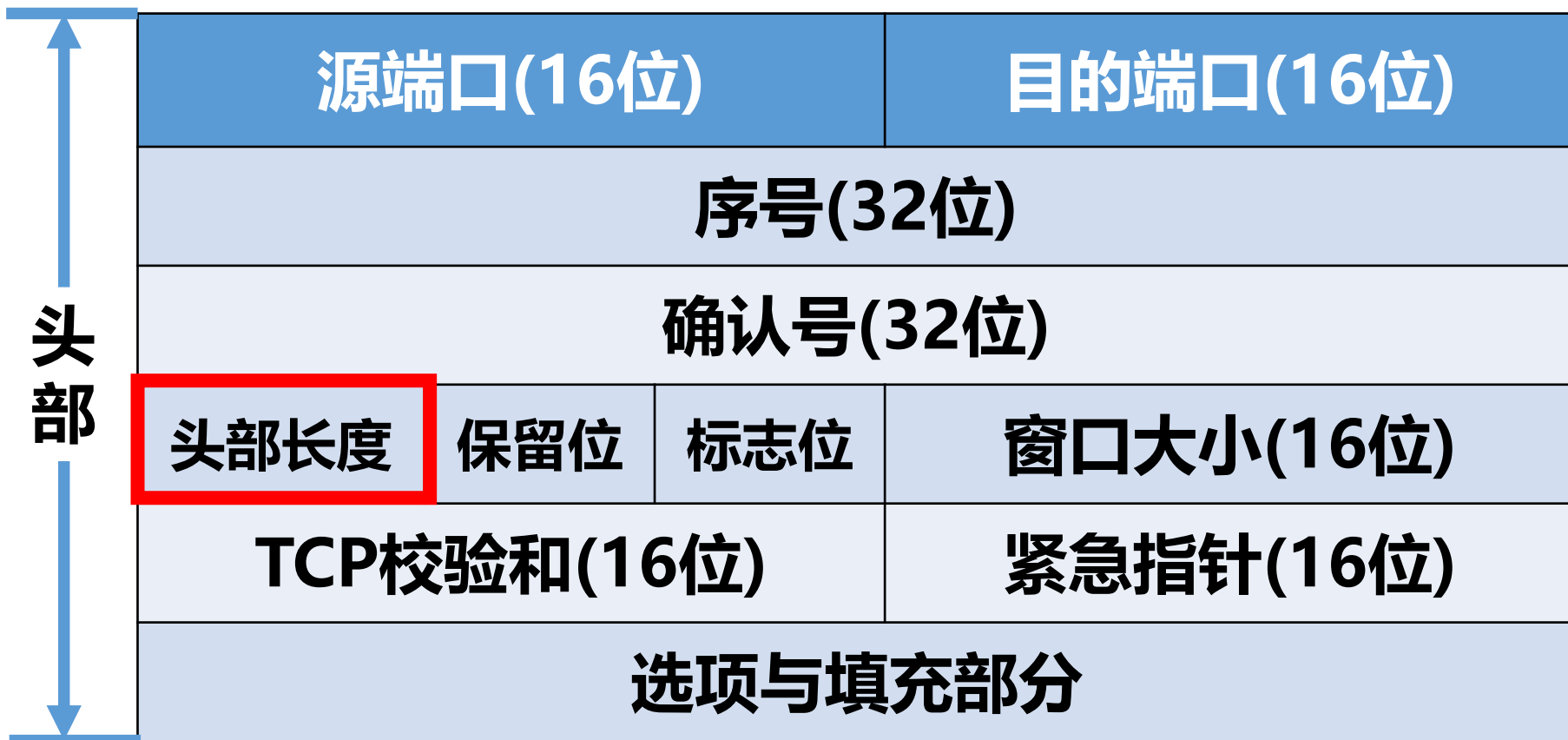
2. 相关知识：TCP数据包的结构

确认号：是期望收到对方的下一个报文段的数据的
第一个字节的序号



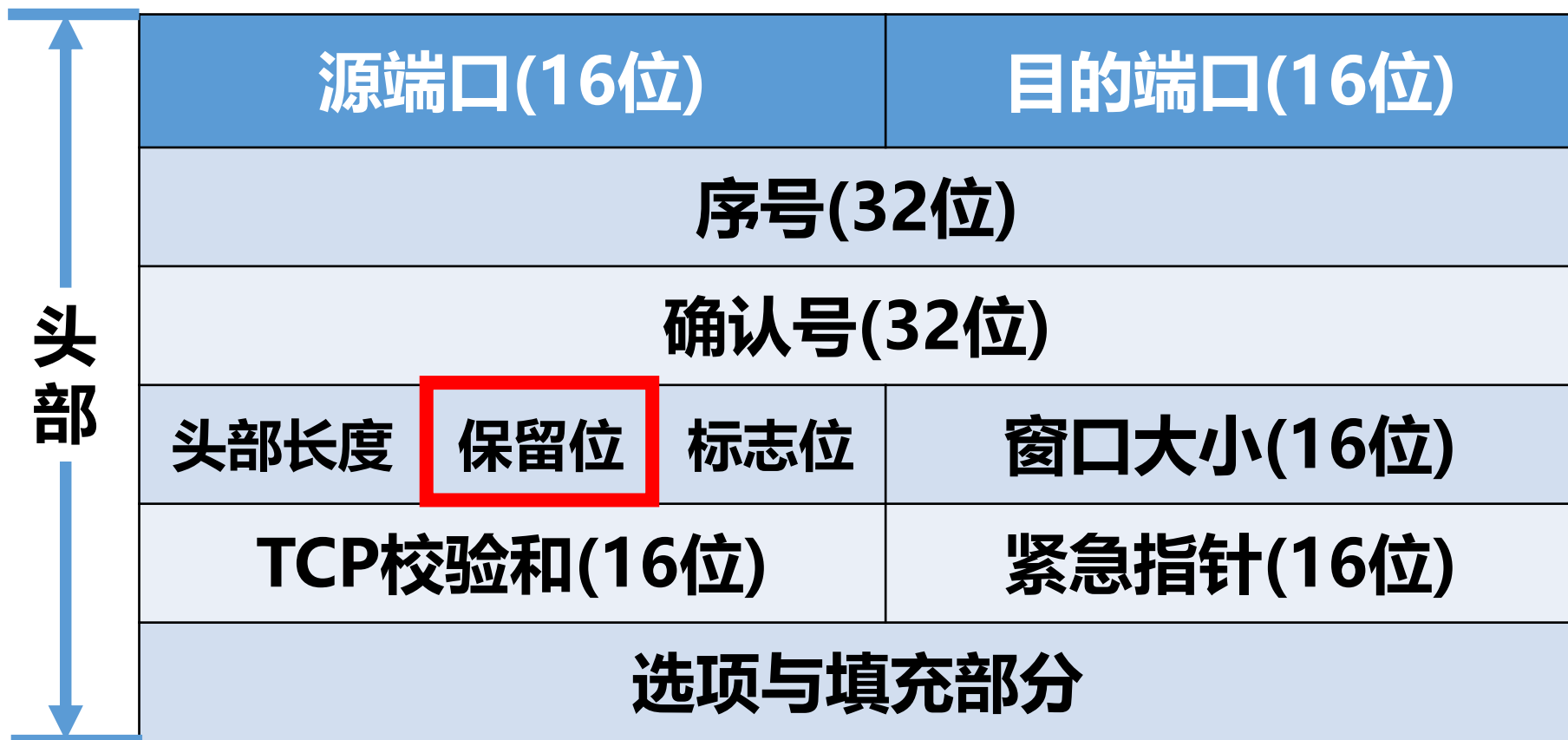
2. 相关知识：TCP数据包的结构

头部长度：4位，表示TCP数据包的头部长度，取值范围是5~15（以4字节为计算单位）



2. 相关知识：TCP数据包的结构

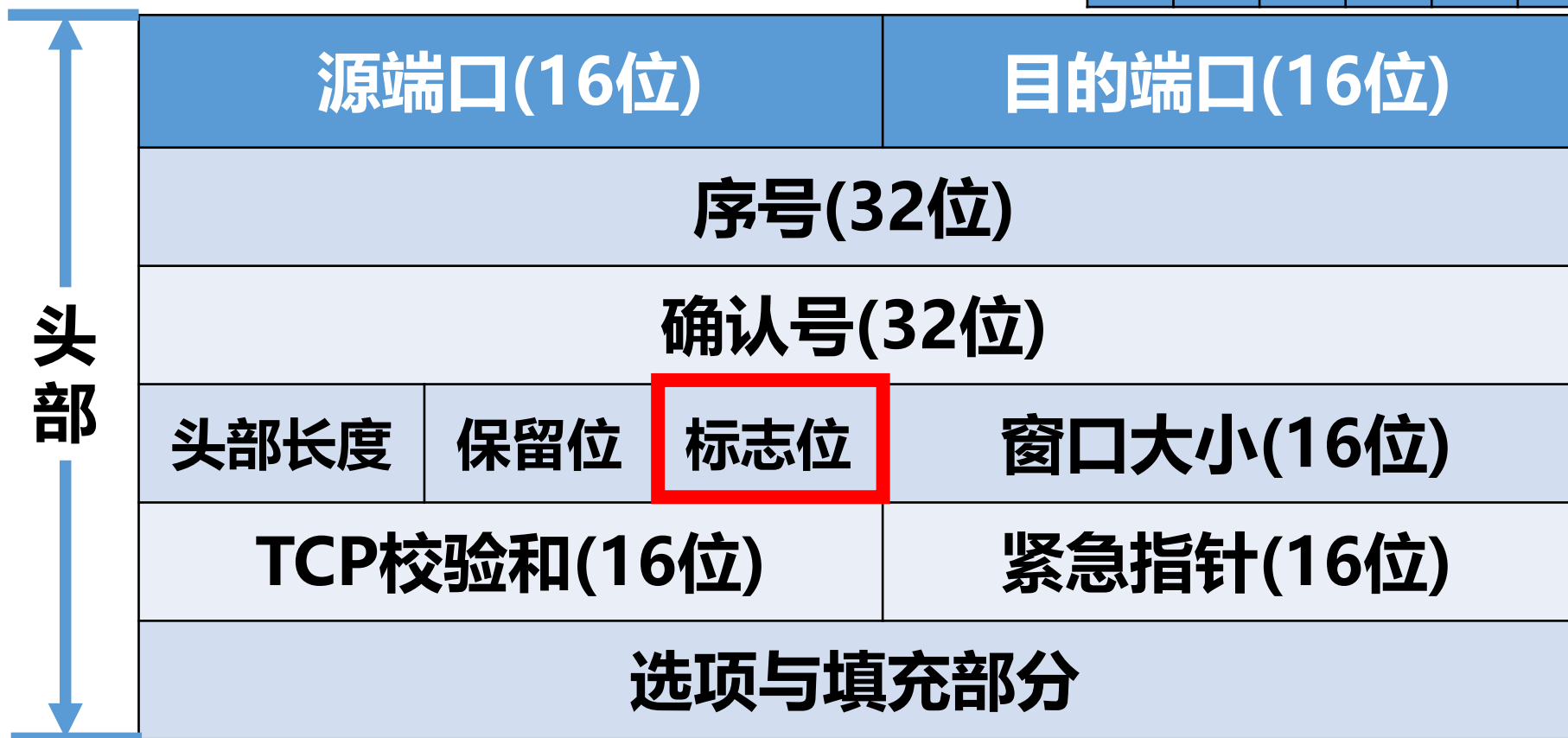
保留位：6位，保留为今后使用，默认置为 0



2. 相关知识：TCP数据包的结构

标志位：6位，设置6种不同的标志位

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N



2. 相关知识：TCP数据包的结构



- ① **紧急 URG** : 当 $URG = 1$ 时, 紧急指针字段有效, 数据的优先级高
- ② **确认 ACK** : 只有当 $ACK = 1$ 时确认号字段才有效; 当 $ACK = 0$ 时, 确认号无效
- ③ **推送 PSH (PuSH)** : 接收 TCP 收到 $PSH = 1$ 的报文段, 就尽快地交付接收应用进程, 而不再等到整个缓存都填满了后再向上交付

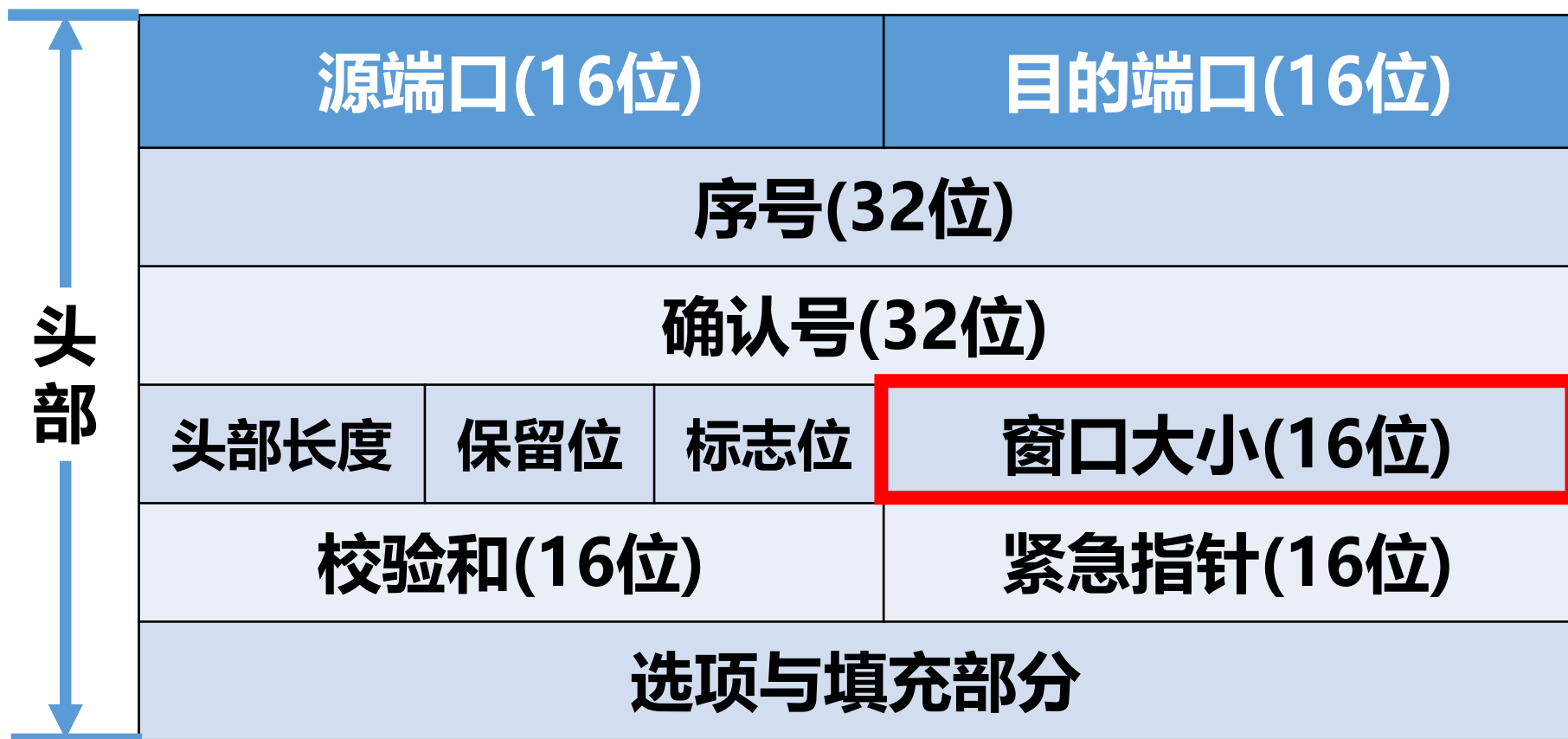
2. 相关知识：TCP数据包的结构



- ④ **复位 RST (ReSeT) :** 当 $RST = 1$ 时, 表明 TCP 连接中出现严重差错 (如由于主机崩溃或其他原因), 必须释放连接, 然后再重新建立运输连接
- ⑤ **同步 SYN :** 同步 $SYN = 1$ 表示这是一个连接请求或连接接受报文。 用于同步序号
- ⑥ **终止 FIN (FINish) :** 用来释放一个连接。 $FIN = 1$ 表明此报文段的发送端的数据已发送完毕, 并要求释放连接

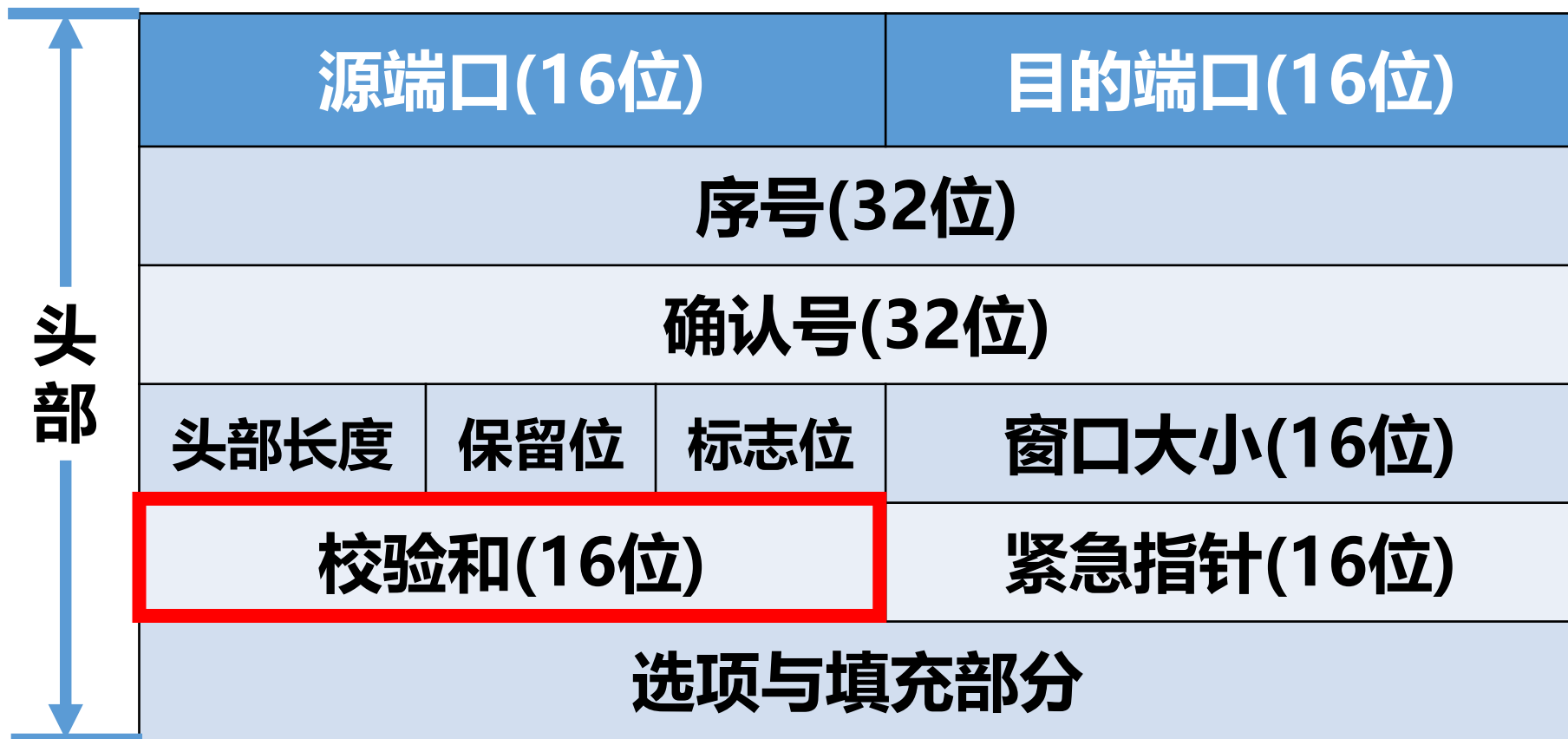
2. 相关知识：TCP数据包的结构

窗口大小：用来让对方设置发送窗口的依据，单位为字节，最大值是 $2^{16} - 1 = 65535$ 字节



2. 相关知识：TCP数据包的结构

校验和：校验范围包括伪头部、头部和数据



2. 相关知识：TCP数据包的结构

在计算检验和时，临时在 TCP 报文段的前面加上 12 字节的伪头部 (pseudo header)

源IP地址(32位)		
目的IP地址(32位)		
保留位(全0)	协议(8位)	TCP长度(16位)

值为6



2. 相关知识：TCP数据包的结构

紧急指针：指出在本报文段中紧急数据共有多少个字节（紧急数据放在本报文段数据的最前面）



2. 相关知识：TCP数据包的结构

选项：0~40字节，如果TCP头部长度不是32位的整数倍，就需要使用填充位（0）凑齐



2. 例题分析：设计要求

- 根据协议规定的TCP数据包的标准格式，编写程序构造TCP包结构，填写各个字段，并将封装后的TCP包内容写入输出文件**
- 为简便起见，数据字段通过为字符串赋值来获得**
- 需要计算头部校验和**

2. 例题分析：设计要求

- 具体要求
 - 要求为命令程序


TcpEncap output_file



输出文件名

2. 例题分析：设计要求

- 具体要求

- 要求将部分字段内容显示在控制台上，具体格式为 

IP头部字段

总长度：xx

IP校验和：xx

源IP地址：xx.xx.xx.xx

目的IP地址：xx.xx.xx.xx

TCP头部与数据字段

TCP长度：xx

源端口：xx

目的端口：xx

TCP校验和：xx

数据字段：xx

2. 例题分析：关键问题

- 定义TCP头部结构

```
typedef struct TCP_HEAD
{
    unsigned short SourcePort;//源端口16位
    unsinged short DestinPort;//目的端口16位
    unsinged int Sequence;//序号
    unsigned int Acknowledge;//确认号
    ... ..
}tcp_head;
```

2. 例题分析：关键问题

```
typedef struct TCP_HEAD
{
    ... ..
    union
    {
        unsigned short HeadLen;//头部长度的4位
        unsigned short Reserved;//保留位6位
        unsigned short Flags;//标志位6位
    };
    ... ..
}tcp_head;
```


2. 例题分析：关键问题

- 定义TCP头部结构

```
typedef struct TCP_HEAD
{
    ... ..
    unsigned short WindowsLen;//窗口大小16位
    unsigned short TcpChecksum;//检验和16位
    unsigned short UrgePoint; //紧急指针16位
}tcp_head;
```

2. 例题分析：关键问题

- 定义TCP伪头部结构

```
typedef struct PSD_HEAD
{
    unsigned int SourceAddr;//源IP地址32位
    unsigned int DestinAddr;//目的IP地址32位
    unsigned char Reserved;//保留位8位
    unsigned char Protocal;//协议8位
    unsinged short TcpLen;//TCP长度16位
}psd_head;
```

2. 例题分析：关键问题

- 填充数据包

```
//初始化相关对象
psd_head psd = {0};//伪头部初始化为全0填充
tcp_head tcp = {0};//tcp头部也全0填充
unsigned short check[65535];//校验缓冲区
const char tcp_data[] =
{ "This is a test of tcp packet encapsule!" };
```

2. 例题分析：关键问题

- 填充数据包

```
//填充TCP伪头部字段
```

```
psd.SourceAddr = ip.SourceAddr;
```

```
psd.DestinAddr = ip.DestinAddr;
```

```
psd.Reserved = 0;
```

```
psd.Protocol = ip.Protocol;
```

```
psd.TcpLen = sizeof(tcp_head) + sizeof(tcp_data);
```

2. 例题分析：关键问题

- 填充数据包

```
//填充TCP头部字段
tcp.SourcePort = 1000;
tcp.DestinPort = 1000;
tcp.Sequence = 0;
tcp.Acknowledge = 0;
tcp.HeadLen =
(sizeof(tcp_head)/sizeof(unsigned int)<<4 | 0);
tcp.WindowsLen = htons((unsigned short)10000);
tcp.TcpChecksum = 0;
tcp.UrgePoint = 0;
```

2. 例题分析：关键问题

- 计算TCP包（包括伪头部）的校验和

```
memset(check, 0, 65535); //全部重置为0
memcpy(  check,
        &psd,
        sizeof(psd_head)); //复制伪头部
memcpy(  check+sizeof(psd_head),
        &tcp,
        sizeof(tcp_head)); //复制tcp头部
```

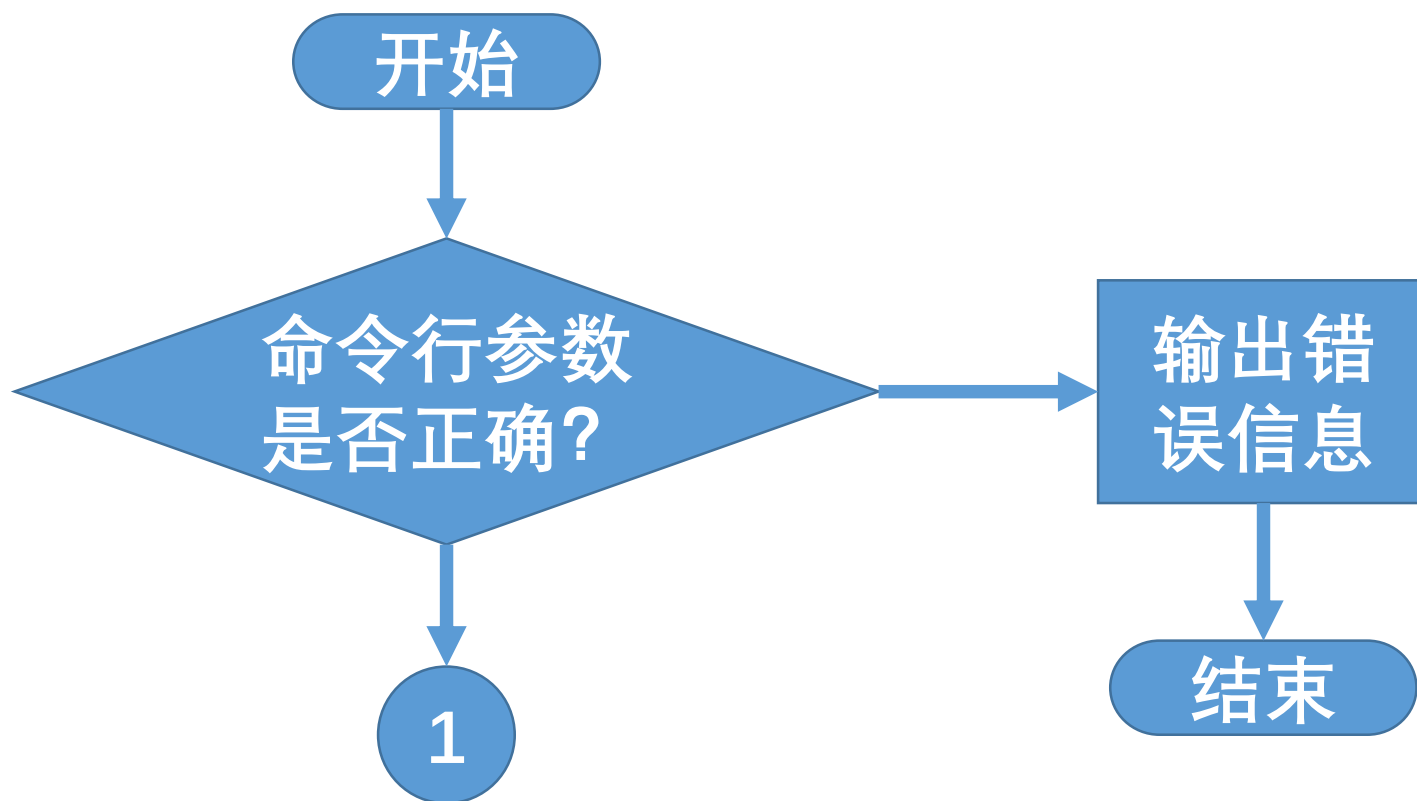
2. 例题分析：关键问题

- 计算TCP包（包括伪头部）的校验和

```
memcpy( check+sizeof(psd_head) +  
        sizeof(tcp_head),  
        tcp_data,  
        sizeof(tcp_data)); //复制tcp数据  
//计算校验和  
tcp.TcpChecksum = checksum(check,  
                             sizeof(psd_head) +  
                             sizeof(tcp_head) +  
                             sizeof(tcp_data));
```

2. 例题分析：关键问题

- 程序流程图



2. 例题分析：关键问题

- 程序流程图

1

打开输出文件

填充IP头部结构

计算IP头部校验和

填充TCP伪头部结构

填充TCP头部结构

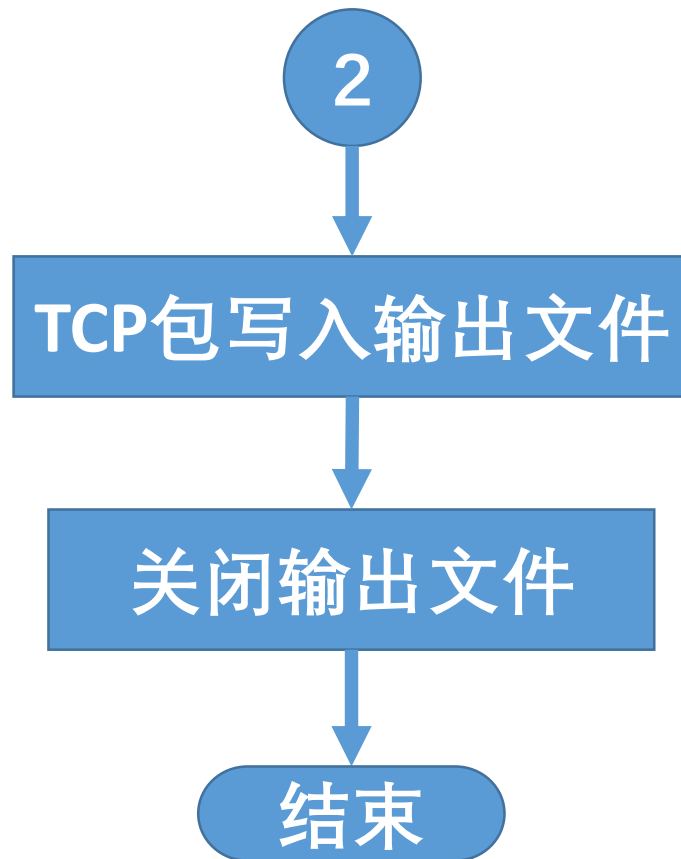
填充TCP数据部分

计算TCP头部校验和

2

2. 例题分析：关键问题

- 程序流程图



2. 例题分析：程序演示

Microsoft Visual Studio 调试控制台

IP头部字段

总长度：80

IP校验和：33511

源IP地址：192.168.1.15

目的IP地址：192.168.1.22

TCP头部与数据字段

TCP长度：60

源端口：1000

目的端口：1000

TCP校验和：1448

数据字段：This is a test of tcp packet encapsule!

TCP包封装完成

作业

- **P131-练习题，说明文档发送到
fanghuixing@hotmail.com**

本章小结

- 设计目的
 - 了解TCP包结构各字段含义用途
 - 掌握封装和发送TCP包的编程方法
- 相关知识
 - TCP协议基本概念
 - TCP数据包的结构
- 例题分析
 - 定义TCP头部、伪头部数据结构
 - 填充数据包、计算校验和