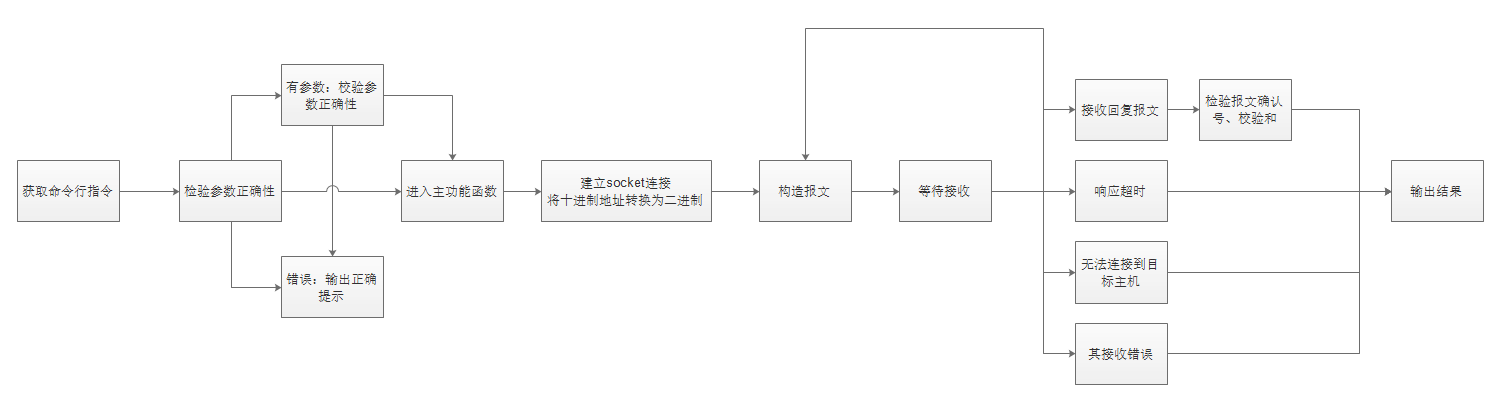
1. 代码结构



1. 主要函数
2. unsigned short checksum(unsigned char\* buf, unsigned int len)

求报文校验和，对每16位进行求和（高位溢出位会加到低位），即先对每16位求和，在将得到的和转为反码

1. int ping\_target\_by\_send\_icmp(char \*dst\_ip, int packet\_length)

主要功能函数，通过构造icmp报文ping目标主机，检测是否存活

1. void help()

帮助函数，参数出错时输出提示正确指令格式

1. int validnum(char \*src)

校验参数值是否有效，有效则返回所输入的参数

1. int main(int argc, char\*\* argv)

主函数，控制整体流程，控制输入输出

1. 功能实现
2. unsigned short checksum(unsigned char\* buf, unsigned int len)

获取报文内容首地址\*buf以及报文长度len，首先将报文首地址指针类型转换为unsigned short\*使其可以每十六位递增。如果len为偶数则将buf中每十六位相加，如果len为奇数则需要最后将多出来的八比特位加至总和。相加完成之后会得到一个不超过32位的总和，需要再将高十六位加至低十六位。这次相加依旧可能产生溢出，再次执行上一步操作，得到一个十六位数字，将它取反并转换为unsigned short类型得到最终的校验和。

1. int ping\_target\_by\_send\_icmp(char \*dst\_ip, int packet\_length)

建立连接

首先使用socket函数，使目标ip与主机尝试建立socket连接，随后设置一些初始值，将协议类型设置为ipv4套接字(AF\_INET)，将目标ip地址从十进制转换为二进制(inet\_pton)

发送报文

首先获取当前进程号pid，作为icmp包头部id，确认号从0开始计数，没发一次报文加一，code段默认置为0，获取当前时间作为计算报文回复时间的起始时间，如果发送报文长度大于10个字节，则可以将这个时间放入报文数据段中，在回复报文传回时获取使时间更为准确。随后计算报文校验和，发送报文，这其中报文总长度使用用户输入的长度或是默认长度加8的ip报文头。

接收回复

获取报文，计算长度，如果长度小于0则说明接收错误，大于0进入下面步骤。获取当前时间，作为结束时间，计算接收报文传递的时间。首先将接受的报文指针以ip格式解析，去除ip报文头部长度后再以icmp格式解析。得到icmp报文结构体，获取校验和，并重新计算校验和，比较两个结果是否相等，如果不相等则说明报文校验和有误。以上正确后，判断icmp\_type类型，如果为ICMP\_ECHOREPLY，说明回复正常，此时检验报文确认号，如果正确则说明目标主机能够ping通。如果为ICMP\_TIME\_EXCEEDED，表示响应超时，并输出提示。如果为ICMP\_DEST\_UNREACH，说明无法到达目标主机。其他情况下，显示其他接收错误。

1. void help()

提供程序功能使用方式，在输入错误时提示用户。

1. int validnum(char \*src)

通过isalnum判断用户在-n或是-l后所输入的内容是否为有效数字，如果不是则返回-2，正确则返回所对应的数值。

1. int main(int argc, char\*\* argv)

首先判断用户输入-n或-l如果有则判断后续输入，一切正确后进入主要功能函数，默认ping的次数为无限次数，默认icmp包数据段大小为56字节。