Flowcharts For Ping.c Code

checksum
----
b , len

(unsigned short) *buf = b , sum , result

sum <- 0
|| len <- len -2
----
len > 1

sum += *buf++

F host_entity ==
NULL T

sum <- sum + *buff

```
tmp1 <- sum BITAND 0xFFFF
tmp2 <- sum BİTSHİFTRİGHT 16
sum <- tmp2 + tmp1
sum <- tmp2 + sum
result <- ~sum
```

return result

stop

```
                      dns_lookup
              ---------------------------------
                 addr_host , addr_con


                    Resolving DNS


                    host_entity =
                gethostbyname(addr_host)


            F                           T
                   host_entity ==
                       NULL


         return NULL


                        ◯


    strcpy(ip, inet_ntoa(*(struct in_addr *)host_entity.h_addr));

          (*addr_con).sin_family <- host_entity.h_addrtype;
              (*addr_con).sin_port <- htons (PORT_NO);
       (*addr_con).sin_addr.s_addr <- *(long*)host_entity.h_addr;


                     return ip


                        stop
```

```
                    reverse_dns_lookup
            -----------------------------------
                          ip_addr


              temp_addr.sin_family <- AF_INET
         temp_addr.sin_addr.s_addr <- inet_addr(ip_addr)
                        buf[NI_MAXHOST]


                 getnameinfo(temp_addr, len, buf,          T
                sizeof(buf), NULL, 0, NI_NAMEREQD)

        F
                                                          return


                   tmp1 <- sum BITAND 0xFFFF
                  tmp2 <- sum BİTSHİFTRİGHT 16
                        sum <- tmp2 + tmp1
                       sum <- tmp2 + sum;
                         result <- ~sum;


                         return result


                            stop
```

```
                    ┌─────────────────────┐
                    │      send_ping      │
                    │ - - - - - - - - - - │
                    │ ping_sockfd, *ping_addr, │
                    │ *ping_dom, *ping_ip, │
                    │     *rev_host       │
                    └─────────────────────┘
                              │
                              ▼
          ┌───────────────────────────────────────┐
          │        (struct ping_pkt) pckt,        │
          │       (struct sockaddr_in) r_addr,    │
          │ (struct timespec) [time_start, time_end, tfs, tfe] │
          │                                       │
          │    ttl_val <- 64, msg_count <- 0, flag <- 1, │
          │         msg_received_count <- 0,      │
          │      rtt_msec <- 0, total_msec <- 0,  │
          │         (struct timeval) tv_out,      │
          │   tv_out.tv_sec <- 1 , tv_out.tv_usec <- 0 │
          └───────────────────────────────────────┘
                              │
                              ▼
          ┌───────────────────────────────────────┐
          │    clock_gettime(CLOCK_MONOTONIC, &tfs); │
          └───────────────────────────────────────┘
                              │
                              ▼
   F                  ╱─────────────────────╲                T
 ┌──────────────────◇  setsockopt(ping_sockfd, SOL_IP, IP_TTL, ◇──────────────┐
 │                   ╲  &ttl_val, sizeof(ttl_val)) != 0  ╱                     │
 │                    ╲─────────────────────╱                                 │
 │                                                                            │
 ▼                                                                            ▼
┌─────────────────────┐                              ┌─────────────────────┐
│ "\nSocket set to TTL..\n" │                        │ "\nSetting socket options to TTL │
│                     │                              │      failed!\n"     │
└─────────────────────┘                              └─────────────────────┘
 │                                                                │
 │                                                                ▼
 │                                                    ┌─────────────────────┐
 │                                                    │       return        │
 │                                                    └─────────────────────┘
 │                              ◯                                 │
 └──────────────────────────────┼────────────────────────────────┘
                                 │
                                 ▼
                    ┌─────────────────────────────────┐
                    │ setsockopt(ping_sockfd, SOL_SOCKET, │
                    │          SO_RCVTIMEO,            │
                    │  (const char*)&tv_out, sizeof tv_out); │
                    └─────────────────────────────────┘
                                 │
                                 ▼                        ┌───┐
                         ⬡─────────────────        ◄──────│ 2 │
                         │  pingloop != 0  │────►  ┌───┐  └───┘
                         ⬡─────────────────        │ 1 │
                                 │                 └───┘
                                 │ F
                                 ▼
                    ┌─────────────────────────────────┐
                    │ clock_gettime(CLOCK_MONOTONIC, &tfe) │
                    └─────────────────────────────────┘
                                 │
                                 ▼
          ┌───────────────────────────────────────────────────────────┐
          │ (double) timeElapsed <- ( (double) (tfe.tv_nsec -tfs.tv_nsec) ) / 1000000.0 │
          │    total_msec <- (tfe.tv_sec-tfs.tv_sec) * 1000.0 + timeElapsed │
          │  p_rec = ( (msg_count - msg_received_count) / msg_count) * 100.0 │
          └───────────────────────────────────────────────────────────┘
                                 │
                                 ▼
                         ┌─────────────────────┐
                         │     "\n===%s ping   │
                         │   statistics===\n", ping_ip │
                         └─────────────────────┘
                                 │
                                 ▼
                    ┌───────────────────────────────────┐
                    │ "\n%d packets sent, %d packets received, %f │
                    │  percent packet loss. Total time: %Lf ms.\n\n", │
                    │     msg_count, msg_received_count, │
                    │          p_rec,total_msec          │
                    └───────────────────────────────────┘
                                 │
                                 ▼
                         ┌─────────────┐
                         │    stop     │
                         └─────────────┘
```

```
                                    ( 1 )
                                      |
                                 flag <- 1
                                      |
                         bzero(&pckt, sizeof(pckt ) )
                                      |
            pckt.hdr.type <- ICMP_ECHO , pckt.hdr.un.echo.id <- getpid()
                                      |
                    +--------------------------------+
                    | i <- 0    || +2                |<---------------+
                    |--------------------------------|                |
                    | i < sizeof(pckt.msg) - 1       |                |
                    +--------------------------------+                |
                                      |          |                    |
                                      |      +------------------------+
                                      |      | pckt.msg[i] <- i+'0'   |
                                      |      +------------------------+
                                      |
                    +--------------------------------------------------+
                    | pckt.msg[i] <- 0                                 |
                    | pckt.hdr.un.echo.sequence <- msg_count           |
                    | msg_count <- msg_count + 1                       |
                    | pckt.hdr.checksum <- checksum(&pckt, sizeof(pckt))|
                    +--------------------------------------------------+
                                      |
                         usleep(PING_SLEEP_RATE)
                                      |
                    clock_gettime(CLOCK_MONOTONIC, &time_start)
                                      |
    F                 sendto(ping_sockfd, &pckt, sizeof(pckt), 0,           T
   <----------------  (struct sockaddr*) ping_addr,                 ------------->
                      sizeof(*ping_addr)) <= 0
                                                              "\nPacket Sending
                                                                  Failed!\n"
                                                                      |
                                                                 flag <- 0
                                                                      |
                    +-------------O<---------------------------------+
                                  |
                         addr_len <- sizeof(r_addr)
                                  |
    F                 recvfrom(ping_sockfd, &pckt, sizeof(pckt), 0,         T
   <----------------  (struct sockaddr*)&r_addr, &addr_len) <= 0   ------------->
                      && msg_count>1
    clock_gettime(CLOCK_MONOTONIC,&time_end);                    "\nPacket receive failed!\n"
                                  |
    timeElapsed <- ( (double) (time_end.tv_nsec - time_start.tv_nsec) ) / 1000000.0
    rtt_msec <- (time_end.tv_sec - time_start.tv_sec) * 1000.0 + timeElapsed
                                  |
              F          flag > 0          T
             <----------    <>       --------->
             |                                  F    NOT( pckt.hdr.type ==69    T
             |                                 <---    AND pckt.hdr.code == 0 )  --->
             |                                 |
             |          "%d bytes from %s (h: %s) (%s) msg_seq=%d ttl=%d rtt = %Lf ms.\n",   "Error..Packet received with
             |          PING_PKT_S , ping_dom , rev_host , ping_ip , msg_count , ttl_val, rtt_msec   ICMP type %d code %d\n",
             |                                 |                                              pckt.hdr.type, pckt.hdr.code
             |          msg_received_count <- msg_received_count + 1
             |                                 |
             |                           O<----+---------------------+
             |                           |
             +-----------O<--------------+
                         |
                         O<--------------------------------------------+
                         |
                       ( 2 )
```

```mermaid
flowchart TD
    mainStart(["mainStart
    ----------------
    argc , **argv"])

    A["net_buf[NI_MAXHOST] ,
    (struct sockaddr_in) addr_con ,
    addrlen <- sizeof(addr_con)"]

    B{"argc != 2"}

    C["\nFormat %s
    <address>\n", argv[0]"]

    D["return 0"]

    E["ip_addr <- dns_lookup(argv[1], &addr_con)"]

    F{"ip_addr == NULL"}

    G["\nDNS lookup failed!
    Could not resolve
    hostname!\n"]

    H["return 0"]

    I["reverse_hostname <- reverse_dns_lookup(ip_addr)"]

    J["\nTrying to connect to '%s' IP:
    %s\n", argv[1], ip_addr"]

    K["\nReverse Lookup domain:
    %s",reverse_hostname"]

    L["sockfd <- socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)"]

    M{"sockfd < 0"}

    N["\nSocket file
    descriptor %d
    received\n", sockfd);"]

    O["\nSocket file descriptor
    not received!!\n"]

    P["return 0"]

    Q["signal(SIGINT, intHandler)"]

    R["send_ping(sockfd, &addr_con, reverse_hostname,ip_addr, argv[1])"]

    stop(["stop"])
```

mainStart
argc , **argv

net_buf[NI_MAXHOST] ,
(struct sockaddr_in) addr_con ,
addrlen <- sizeof(addr_con)

argc != 2   F / T

"\nFormat %s
<address>\n", argv[0]

return 0

ip_addr <- dns_lookup(argv[1], &addr_con)

ip_addr == NULL   F / T

"\nDNS lookup failed!
Could not resolve
hostname!\n"

return 0

reverse_hostname <- reverse_dns_lookup(ip_addr)

"\nTrying to connect to '%s' IP:
%s\n", argv[1], ip_addr

"\nReverse Lookup domain:
%s",reverse_hostname

sockfd <- socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)

sockfd < 0   F / T

"\nSocket file
descriptor %d
received\n", sockfd);

"\nSocket file descriptor
not received!!\n"

return 0

signal(SIGINT, intHandler)

send_ping(sockfd, &addr_con, reverse_hostname,ip_addr, argv[1])

stop