Advanced Programming in the UNIX Environment

Week 09, Segment 3:
socket(PF_INET, SOCK_DGRAM, 0)

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

socket(2)

```
#include <sys/socket.h>

int socket(int domain, int type, int protocol);
```

Returns: fd if ok, -1 otherwise

socket(2) creates an endpoint for communication and returns a descriptor.

The *domain* specified selects the address- or name space of the socket, which selects the protocol family.

The *type* selects the semantics of communication; *protocol* selects specific rules / formats for this type. In practice, selecting the default protocol by specifying 0 is generally sufficient.

Terminal — 159×33

```
    The hexadecimal set:

    00 NUL   01 SOH   02 STX   03 ETX   04 EOT   05 ENQ   06 ACK   07 BEL
    08 BS    09 HT    0a LF    0b VT    0c FF    0d CR    0e SO    0f SI
    10 DLE   11 DC1   12 DC2   13 DC3   14 DC4   15 NAK   16 SYN   17 ETB
    18 CAN   19 EM    1a SUB   1b ESC   1c FS    1d GS    1e RS    1f US
    20 SP    21  !    22  "    23  #    24  $    25  %    26  &    27  '
    28  (    29  )    2a  *    2b  +    2c  ,    2d  -    2e  .    2f  /
    30  0    31  1    32  2    33  3    34  4    35  5    36  6    37  7
    38  8    39  9    3a  :    3b  ;    3c  <    3d  =    3e  >    3f  ?
    40  @    41  A    42  B    43  C    44  D    45  E    46  F    47  G
    48  H    49  I    4a  J    4b  K    4c  L    4d  M    4e  N    4f  O
    50  P    51  Q    52  R    53  S    54  T    55  U    56  V    57  W
    58  X    59  Y    5a  Z    5b  [    5c  \    5d  ]    5e  ^    5f  _
    60  `    61  a    62  b    63  c    64  d    65  e    66  f    67  g
    68  h    69  i    6a  j    6b  k    6c  l    6d  m    6e  n    6f  o
    70  p    71  q    72  r    73  s    74  t    75  u    76  v    77  w
    78  x    79  y    7a  z    7b  {    7c  |    7d  }    7e  ~    7f DEL

    The decimal set:

      0 NUL     1 SOH     2 STX     3 ETX     4 EOT     5 ENQ     6 ACK     7 BEL
      8 BS      9 HT     10 LF     11 VT     12 FF     13 CR     14 SO     15 SI
     16 DLE    17 DC1    18 DC2    19 DC3    20 DC4    21 NAK    22 SYN    23 ETB
     24 CAN    25 EM     26 SUB    27 ESC    28 FS     29 GS     30 RS     31 US
     32 SP     33  !     34  "     35  #     36  $     37  %     38  &     39  '
     40  (     41  )     42  *     43  +     44  ,     45  -     46  .     47  /
     48  0     49  1     50  2     51  3     52  4     53  5     54  6     55  7
     56  8     57  9     58  :     59  ;     60  <     61  =     62  >     63  ?
jschauma@apue$ ./send panix.netmeister.org 54670
jschauma@apue$
  0 bash
```

```
        0x0000:  4500 004c 04d7 0000 4011 bc04 0a00 020f  E..L....@.......
        0x0010:  a654 0763 ffd2 d57d 0038 5e0e 5468 6520  .T.c...}.8^.The.
        0x0020:  7365 6120 6973 2063 616c 6d20 746f 6e69  sea.is.calm.toni
        0x0030:  6768 742c 2074 6865 2074 6964 6520 6973  ght,.the.tide.is
        0x0040:  2066 756c 6c20 2e20 2e20 2e00            .full.......
15:31:06.936341 IP 10.0.2.15.65488 > 166.84.7.99.54653: UDP, length 48
        0x0000:  4500 004c 04e1 0000 4011 bbfa 0a00 020f  E..L....@.......
        0x0010:  a654 0763 ffd0 d57d 0038 5e10 5468 6520  .T.c...}.8^.The.
        0x0020:  7365 6120 6973 2063 616c 6d20 746f 6e69  sea.is.calm.toni
        0x0030:  6768 742c 2074 6865 2074 6964 6520 6973  ght,.the.tide.is
        0x0040:  2066 756c 6c20 2e20 2e20 2e00            .full.......
15:31:06.979406 IP 166.84.7.99 > 10.0.2.15: ICMP 166.84.7.99 udp port 54653 unr
eachable, length 56
        0x0000:  45c0 004c 085d 0000 3f01 b8ce a654 0763  E..L.]..?....T.c
        0x0010:  0a00 020f 0303 001b 0000 0000 4500 004c  ............E..L
        0x0020:  04e1 0000 3f11 bcfa 0a00 020f a654 0763  ....?........T.c
        0x0030:  ffd0 d57d 0038 5e10 5468 6520 7365 6120  ...}.8^.The.sea.
        0x0040:  6973 2063 616c 6d20 746f 6e69            is.calm.toni
15:35:05.506554 IP 10.0.2.15.65486 > 166.84.7.99.54670: UDP, length 48
        0x0000:  4500 004c 05b4 0000 4011 bb27 0a00 020f  E..L....@..'....
        0x0010:  a654 0763 ffce d58e 0038 5e01 5468 6520  .T.c.....8^.The.
        0x0020:  7365 6120 6973 2063 616c 6d20 746f 6e69  sea.is.calm.toni
        0x0030:  6768 742c 2074 6865 2074 6964 6520 6973  ght,.the.tide.is
        0x0040:  2066 756c 6c20 2e20 2e20 2e00            .full.......
15:35:05.552014 IP 166.84.7.99 > 10.0.2.15: ICMP 166.84.7.99 udp port 54670 unr
eachable, length 56
        0x0000:  45c0 004c 09e2 0000 3f01 b749 a654 0763  E..L....?..I.T.c
        0x0010:  0a00 020f 0303 001b 0000 0000 4500 004c  ............E..L
        0x0020:  05b4 0000 3f11 bc27 0a00 020f a654 0763  ....?..'.....T.c
        0x0030:  ffce d58e 0038 5e01 5468 6520 7365 6120  .....8^.The.sea.
        0x0040:  6973 2063 616c 6d20 746f 6e69            is.calm.toni

  3 bash
```

Sockets: Datagrams in the Internet Domain

- Unlike UNIX domain names, Internet socket names are not entered into the file system and, therefore, they do not have to be unlinked after the socket has been closed.
- The local machine address for a socket can be any valid network address of the machine, or it can be the wildcard value `INADDR_ANY`.
- request any ephemeral port by calling `bind(2)` with a port number of 0
- "well-known" ports (range 1 - 1023) can only be bound by `euid` 0
- determine used port number (or other information) using `getsockname(2)`
- convert between network byte order and host byte order using `htons(3)` and `ntohs(3)` (which may be noops)
- UDP is connectionless / unreliable: you can (try to) send packets without anything listening

Questions

- Change the reader program to accept as command-line arguments an IP address or hostname as well as a port number to use.
- What happens if you don't use `htons(3)`/`ntohs(3)`?
- A host may have multiple IP addresses - how do our programs handle those cases?
- What happens if you specify a host that has both an IPv4 and an IPv6 address? What if it has only an IPv6 address? (You can experiment by manually adding entries in `/etc/hosts`.)
- Practice using `tcpdump(8)` to observe the network traffic. Use e.g. https://www.wireshark.org/ to help read the output using a GUI.