Network Programming for Windows 03:

Internet Protocol

jintaeks@dongseo.ac.kr
Division of Digital Contents, DSU

Outline

- ✓ IPv4
- ✓ IPv6
- ✓ Address and Name Resolution
- ✓ ~~Writing IP Version-independent Programs~~

- ✓ IPv4 is commonly known as the **network protocol** that the Internet uses.
  - The background,
  - addressing scheme,
  - name resolution,
  - and Winsock specifics for both IPv4 and ~~IPv6~~.

IPv4

- ✓ IPv4 was developed by the U.S. Department of Defense's Advanced Research Project Agency (ARPA).
- ✓ This research eventually led to IPv4 as well as TCP.

Addressing

- ✓ In IPv4, computers are assigned an address that is represented as a 32-bit number, formally known as an **IPv4 address**.
- ✓ IPv4 addresses are divided into **classes** that describe the portion of the address assigned to the network and the portion assigned to endpoints.

Class	Network Portion	First Number	Number of Endpoints	Default Subnet Mask
A	8 bits	0–127	16,777,216	255.0.0.0
B	16 bits	128–191	65,536	255.255.0.0
C	24 bits	192–223	256	255.255.255.0
D	N/A	224–239	N/A	n/a
E	N/A	240–255	N/A	n/a

```
C:₩Users₩13FGames>ipconfig /all
```

```
Wireless LAN adapter Wi-Fi 5:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : 802.11ac Wireless LAN Card
   Physical Address. . . . . . . . . : 88-36-6C-F9-D8-CB
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::6422:e3b0:3046:2588%2(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.130(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Sunday, May 24, 2020 2:00:25 PM
   Lease Expires . . . . . . . . . . : Sunday, May 24, 2020 7:00:24 PM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 42481260
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-C4-59-85-CC-B0-DA-76-86-8D
   DNS Servers . . . . . . . . . . . : 168.126.63.1
                                       168.126.63.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:₩Users₩13FGames>
```

slash notation

- ✓ The address 172.31.28.120**/16** indicates that the first 16 bits make up the **network portion** of the address.
- ✓ This is equivalent to a **subnet mask** of 255.255.0.0
- ✓ Class D addresses are reserved for IPv4 **multicasting**
- ✓ Class E addresses are experimental.
- ✓ Reserved for private:
  - 10.0.0.0–10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0–172.31.255.255 (172.16.0.0/12)
  - **192.168.0.0**–192.168.255.255 (192.168.0.0/16)
- ✓ The **loopback** address (**127.0.0.1**)
  - special address that refers to the local computer

Unicast, Multicast addressed

- ✓ **Unicast addresses** are those addresses that are assigned to an individual computer interface.
- ✓ Classes A, B, and C comprise the unicast address space for IPv4.
- ✓ Typically, an interface on a host is assigned an IPv4 (unicast) address either statically or by a configuration protocol like Dynamic Host Configuration Protocol (**DHCP**).
- ✓ **Multicast addresses** are not assigned to a specific interface.
  - Instead, multiple computers may "join" a **multicast group** listening on a particular multicast address.

Broadcast

- ✓ The data sent to the limited broadcast address, **255.255.255.255**, will be received and processed by every machine on the **local network**.
  - bad practice
- ✓ If applications require broadcasting, it is better to use **subnet directed broadcasts**.
  - UDP datagram

IPv4 Management Protocols

- ✓ The IPv4 protocol relies on several other protocols to function.
- ✓ The three support protocols we are most interested in is the **Address Resolution Protocol** (ARP), the **Internet Control Message Protocol** (ICMP), and the **Internet Group Management Protocol** (IGMP).
- ✓ **ARP** is used to resolve the 32-bit IPv4 address into a physical or hardware address so the IPv4 packet can be wrapped in the appropriate media frame (such as an Ethernet frame).
  - C:₩> arp -a
- ✓ **ICMP** is designed to send status and error messages between IPv4 hosts.
  - C:₩> ping 127.0.0.1
- ✓ **IGMP** is used to manage multicast group membership.

```
C:₩Users₩13FGames>nslookup www.google.com
Server:  kns.kornet.net
Address:  168.126.63.1

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4004:81c::2004
          172.217.174.100


C:₩Users₩13FGames>ping 172.217.174.100

Pinging 172.217.174.100 with 32 bytes of data:
Reply from 172.217.174.100: bytes=32 time=39ms TTL=54
Reply from 172.217.174.100: bytes=32 time=41ms TTL=54
Reply from 172.217.174.100: bytes=32 time=39ms TTL=54
Reply from 172.217.174.100: bytes=32 time=42ms TTL=54

Ping statistics for 172.217.174.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 39ms, Maximum = 42ms, Average = 40ms
```

ARP

- ✓ **ARP** is used to resolve the 32-bit IPv4 address into a physical or hardware address so the IPv4 packet can be wrapped in the appropriate media frame (such as an Ethernet frame).

C:\WINDOWS\system32\cmd.exe

```
C:\>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                table are displayed.
  -g            Same as -a.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr. inet_addr may be
                wildcarded with * to delete all hosts.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.

C:\>_
```

ICMP

✓ The **ping** command is based on the ICMP protocol.

C:\WINDOWS\system32\cmd.exe

```
C:\>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet.
    -i TTL         Time To Live.
    -v TOS         Type Of Service.
    -r count       Record route for count hops.
    -s count       Timestamp for count hops.
    -j host-list   Loose source route along host-list.
    -k host-list   Strict source route along host-list.
    -w timeout     Timeout in milliseconds to wait for each reply.

C:\>
```

IPv6

✓ **netsh** interface ipv6 show interface

C:\WINDOWS\system32\cmd.exe - netsh

```
C:\>netsh
netsh>?

The following commands are available:

Commands in this context:
..              - Goes up one context level.
?               - Displays a list of commands.
abort           - Discards changes made while in offline mode.
add             - Adds a configuration entry to a list of entries.
alias           - Adds an alias.
bridge          - Changes to the `netsh bridge' context.
bye             - Exits the program.
commit          - Commits changes made while in offline mode.
delete          - Deletes a configuration entry from a list of entries.
diag            - Changes to the `netsh diag' context.
dump            - Displays a configuration script.
exec            - Runs a script file.
exit            - Exits the program.
firewall        - Changes to the `netsh firewall' context.
help            - Displays a list of commands.
interface       - Changes to the `netsh interface' context.
offline         - Sets the current mode to offline.
online          - Sets the current mode to online.
popd            - Pops a context from the stack.
pushd           - Pushes current context on stack.
quit            - Exits the program.
ras             - Changes to the `netsh ras' context.
routing         - Changes to the `netsh routing' context.
set             - Updates configuration settings.
show            - Displays information.
unalias         - Deletes an alias.
winsock         - Changes to the `netsh winsock' context.

The following sub-contexts are available:
 bridge diag firewall interface ras routing winsock

To view help for a command, type the command, followed by a space, and
 type ?.

netsh>
```

tracert

```
C:₩Users₩13FGames>tracert www.google.com

Tracing route to www.google.com [172.217.24.132]
over a maximum of 30 hops:

  1     1 ms     1 ms     3 ms  192.168.0.1
  2    81 ms     3 ms     3 ms  222.96.24.1
  3     3 ms     3 ms     3 ms  112.191.9.233
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     9 ms    11 ms     9 ms  112.174.5.118
  7    45 ms    45 ms    49 ms  72.14.194.194
  8    38 ms    40 ms    38 ms  108.170.242.161
  9    41 ms    40 ms    40 ms  72.14.234.229
 10    40 ms    40 ms    40 ms  nrt20s01-in-f4.1e100.net [172.217.24.132]

Trace complete.

C:₩Users₩13FGames>
```

Download and Install Curl.exe

curl.se/windows/

seojt 주소록 캘린더 Gmail mgdsu 검색 진택 위키 Wiki 동서대 GP101 학생 개발

Download Documentation libcurl Get Help Developr

curl / Download / **Windows downloads**

curl 7.86.0 for Windows

These are the latest and most up to date **official** curl binary builds for Microsoft Windo

curl version: 7.86.0
Build: 7.86.0_2
Date: 2022-10-26
Changes: 7.86.0 changelog

curl for 64-bit

Size: 10.1 MB
sha256: 1175599e2c8a26fdfa981064367bfe8e07cb17457eb6027f640f5a6de

curl for 64-bit (ARM)

Size: 8.7 MB

curl

```
C:₩Users₩13FGames>tracert www.google.com

Tracing route to www.google.com [172.217.24.132]
over a maximum of 30 hops:

  1     1 ms     1 ms     3 ms  192.168.0.1
  2    81 ms     3 ms     3 ms  222.96.24.1
  3     3 ms     3 ms     3 ms  112.191.9.233
  4     *        *        *     Re
  5     *        *        *     R
  6     9 ms    11 ms     9 ms  1
  7    45 ms    45 ms    49 ms  7
  8    38 ms    40 ms    38 ms  1
  9    41 ms    40 ms    40 ms  7
 10    40 ms    40 ms    40 ms  nrt20s0

Trace complete.

C:₩Users₩13FGames>curl ipinfo.io/222.96.24.1
{
  "ip": "222.96.24.1",
  "city": "Busan",
  "region": "Busan",
  "country": "KR",
  "loc": "35.1017,129.0300",
  "org": "AS4766 Korea Telecom",
  "postal": "48926",
  "timezone": "Asia/Seoul",
  "readme": "https://ipinfo.io/missingauth"
}
```

curl ipinfo.io/222.96.24.1

```
C:₩Users₩13FGames>curl ipinfo.io/112.174.5.118
{
  .118",
  ",
  126.9784",
  orea Telecom",
  "postal": "03186",
  "timezone": "Asia/Seoul",
  "readme": "https://ipinfo.io/missingauth"
}
C:₩Users₩13FGames>curl ipinfo.io/72.14.194.194
{
  "ip": "72.14.194.194",
  "city": "Mountain View",
  "region": "California",
  "country": "US",
  "loc": "37.4056,-122.0775",
  "org": "AS15169 Google LLC",
  "postal": "94043",
  "timezone": "America/Los_Angeles",
  "readme": "https://ipinfo.io/missingauth"
}
C:₩Users₩13FGames>
```

Download wget.exe

eternallybored.org/misc/wget/

주소록 캘린더 Gmail mgdsu 검색 진택 위키 Wiki 동서대 GP101 학생 개발

Windows binaries of GNU Wget

A command-line utility for retrieving files using HTTP, HTTPS and FTP protocols.

Warning: some antivirus tools recognise wget-1.21.3-win32.zip as potentially dangerous. The file that triggers the warning is wget.exe.debug, which contains debugging symbols for wget.exe, and isn't even executable. If your AV is giving you trouble, and you don't need the documentation or debug symbols, you can download wget.exe directly, or switch to a less broken security product.

All of the binaries are compiled statically, meaning that wget.exe doesn't require any other files to work.

Version	32-bit		64-bit		Notes
1.21.3	ZIP	EXE	ZIP	EXE	OpenSSL 1.1.1m, ZLib 1.2.11, gpgme-1.17.1, pcre2 10.39, libpsl 0.21.1, c-ares 1.18.1, taskbar progressbar, Windows certificate store support, manual
1.21.2	ZIP	EXE	ZIP	EXE	OpenSSL 1.1.1l, ZLib 1.2.11, gpgme-1.16.0, pcre2 10.38, libpsl 0.21.1, c-ares 1.17.2, taskbar progressbar, Windows certificate store support, manual
1.21.1	ZIP	EXE	ZIP	EXE	OpenSSL 1.1.1k, ZLib 1.2.11, gpgme-1.15.1, pcre2 10.36, libpsl 0.21.1, c-ares 1.17.1, taskbar progressbar, Windows certificate store support, fix for downloading files >2GB, manual
1.20.3	ZIP	EXE	ZIP	EXE	OpenSSL 1. [illegible] re2 10.32, libpsl 0.20.2, taskbar progressbar, Windows certificate store support, manual

store: 가게, 저장, 상점, 점포 shop, 식료품점

Windo
Gifsic
NetC
PCI U
GNU
Scripts
Autoh
for (h
for VI
Other
GIMP
Provided


```
D:₩Work₩Temp2>wget https://en.wikipedia.org/wiki/Spacetime
```

Download and Install youtube-dl.exe

ytdl-org.github.io/youtube-dl/download.html

seojt 주소록 캘린더 Gmail mgdsu 검색 진택 위키 Wiki 동서대 GP101 학

youtube-dl

Remember *youtube-dl* requires Python version 2.6, 2.7, or 3.2+ to work except for Windows e

Windows exe requires Microsoft Visual C++ 2010 Service Pack 1 Redistributable Package (x86
embedded into the binary.

2021.12.17 (sig)

SHA256: 7880e01abe282c7fd596f429c35189851180d6177302bb215be1cdec78d6d06d

Windows exe (sig - SHA256 26e5c00c35c5c3edc86dfc0a720aed109a13b1b7c67ac654a0ce8ff82
Full source + docs + binary tarball (sig - SHA256 9f3b99c8b778455165b4525f21505e86c7ff56

To install it right away for all UNIX users (Linux, OS X, etc.), type:

```
D:₩Work₩Temp2>youtube-dl https://www.youtube.com/watch?v=u8PGpl7v20s
[youtube] u8PGpl7v20s: Downloading webpage
[download] Destination: How to Build Games that Keep Players Happy & Engaged _ Unite 2022-u8PGpl7v20s.mp4
```

Addressing IPv6 from Winsock

```
struct sockaddr_in6 {
    short          sin6_family;
    u_short                sin6_port;
    u_long         sin6_flowinfo;
    struct in6_addr sin6_addr;
    u_long         sin6_scope_id;
};
```

Address and Name Resolution

- ✓ How to assign both **literal string addresses** and resolve names to the address specific **structures** for both IP protocols.
- ✓ Name resolution APIs
  - getaddrinfo()
  - getnameinfo()
- ✓ Winsock APIs for converting between string literal addresses and socket address structure.
  - WSAAddressToString()
  - WSAStringToAddress()

Name Resolution Routines

- ✓ The legacy functions like **gethostbyname**() and **inet_addr**() work with IPv4 addresses only.
- ✓ New name resolution routines are defined in **WS2TCPIP.H**.
- ✓ The **getaddrinfo**() function provides protocol-independent name resolution.

```
int getaddrinfo(
    const char FAR *nodename,
    const char FAR *servname,
    const struct addrinfo FAR *hints,
    struct addrinfo FAR *FAR *res
    );
```

```
struct addrinfo {
    int                 ai_flags;
    int                 ai_family;
    int                 ai_socktype;
    int                 ai_protocol;
    size_t              ai_addrlen;
    char                *ai_canonname;
    struct sockaddr *ai_addr;
    struct addrinfo *ai_next;
};
```

- ✓ **ai_flags**: AI_PASSIVE, AI_CANONNAME, or AI_NUMERICHOST.
  - AI_CANONNAME indicates that nodename is a computer name like www.microsoft.com
  - AI_NUMERICHOST indicates that it is a literal string address such as "10.10.10.1".
  - AI_PASSIVE will be discussed later.
- ✓ **ai_family**: AF_INET, AF_INET6, or AF_UNSPEC.
  - if AF_UNSPEC is given, then the addresses returned could be either IPv4 or IPv6 or both.
- ✓ **ai_socktype**: specifies the desired socket type, such as SOCK_DGRAM, SOCK_STREAM.
- ✓ **ai_protocol**: specifies the desired protocol, such as IPPROTO_TCP.

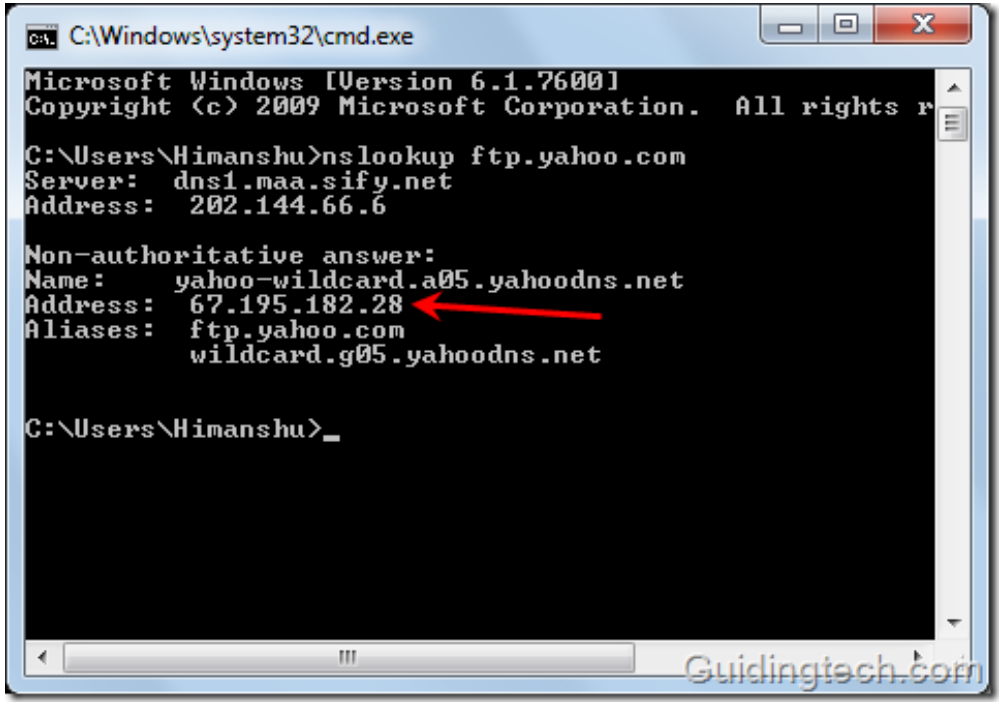
```
// Declare and initialize variables.
char* ip = "127.0.0.1";
char* port = "7777";
struct addrinfo aiHints;
struct addrinfo *aiList = NULL;
int retVal;

// Setup the hints address info structure
// which is passed to the getaddrinfo() function
memset(&aiHints, 0, sizeof(aiHints));
aiHints.ai_family = AF_INET;
aiHints.ai_socktype = SOCK_STREAM;
aiHints.ai_protocol = IPPROTO_TCP;

// Call getaddrinfo(). If the call succeeds, the aiList variable
// will hold a linked list f addrinfo structures containing
// response information about the host
if ((retVal = getaddrinfo(ip, port, &aiHints, &aiList)) != 0)
{
    printf("getaddrinfo() failed with error code %d.₩n", retVal);
}
```

- ✓ **getnameinfo**() takes a socket address structure already initialized and returns the host and service name corresponding to the address and port information.

```
int getnameinfo(
    const struct sockaddr FAR *sa,
    socklen_t salen,
    char FAR *host,
    DWORD hostlen,
    char FAR *serv,
    DWORD servlen,
    int flags
    );
```



- ✓ **nslookup** command line tool.
  - get ip from domain name

Simple Address Conversion

- ✓ To convert between string literal addresses and socket address structures, the **WSAStringToAddress**() and **WSAAddressToString**() helper APIs are available.

```
INT WSAStringToAddress(
    LPTSTR AddressString,
    INT AddressFamily,
    LPWSAPROTOCOL_INFO lpProtocolInfo,
    LPSOCKADDR lpAddress,
    LPINT lpAddressLength
    );
```

- ✓ The API functions **getservbyname**() and WSAAsyncGetServByName() take the name of a well-known service like “FTP” and return the port number that the service uses.

```
struct servent FAR * getservbyname(
    const char FAR * name,
    const char FAR * proto
    );
```

```
struct servent {
    char FAR *          s_name;
    char FAR * FAR *    s_aliases;
    short               s_port;
    char FAR *          s_proto
};
```

find a process which uses a specific port

```
tions, listening ports, and bound
rts. Bo
an acti
g table
ol stat
 ICMP,
e used
t conne
ect con
```

D:\github\DiconServer\chapter01\Debug\tcpserver.exe

```
We are awaiting a connection on port 5150.
```

```
C:₩Users₩13FGames>netstat -a -o | find "5150"
  TCP    0.0.0.0:5150           NotebookGram-Jintaeks:0  LISTENING       22048

C:₩Users₩13FGames>_
```

```
C:₩Users₩13FGames>netstat -a -o | find "5150"
  TCP    0.0.0.0:5150           NotebookGram-Jintaeks:0  LISTENING       22048

C:₩Users₩13FGames>_
```

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	PID	Status	User name	CPU	Memo
devenv.exe	19320	Running	13FGames	00	316,
KakaoTalk.exe	19448	Running	13FGames	00	34,
CAudioFilterAgent64...	19712	Running	13FGames	00	
CDASrv.exe	19884	Running	13FGames	00	1,
QtWebEngineProces...	20016	Running	13FGames	00	9,
chrome.exe	20056	Running	13FGames	00	9,
chrome.exe	20536	Running	13FGames	00	42,
OSPPSVC.EXE	20560	Running	NETWORK...	00	
chrome.exe	20764	Running	13FGames	01	66,
svchost.exe	20792	Running	SYSTEM	00	3,
SystemSettingsBroke...	20872	Running	13FGames	00	1,
QtWebEngineProces...	21060	Running	13FGames	00	
WindowsInternal.Co...	21304	Running	13FGames	00	2,
RuntimeBroker.exe	21688	Running	13FGames	00	3,
ShellExperienceHost....	21744	Suspended	13FGames	00	
SearchProtocolHost....	21812	Running	13FGames	00	1,
ServiceHub.Host.CL...	21972	Running	13FGames	00	8,
tcpserver.exe	22048	Running	13FGames	00	
audiodg.exe	22108	Running	LOCAL SE...	00	4,
RuntimeBroker.exe	22280	Running	13FGames	00	

~~Writing IP Version-independent Program~~

- ✓ Windows IPv6 stack is a dual stack.
- ✓ That is, there is a separate stack for IPv4 and IPv6, so if a server wishes to accept both IPv4 and IPv6 connections, it must create a socket for each one.
  - Call **getaddrinfo**() with hints containing AI_PASSIVE, AF_UNSPEC, and the desired socket type and protocol along with the desired local port to listen or receive data on.
  - This will return **two addrinfo** structures: one containing the listening address for IPv4 and the other containing the listening address for IPv6.
  - For every addrinfo structure returned, **create a socket** with the ai_family, ai_socktype, and ai_protocol fields followed by calling bind() with the ai_addr and ai_addrlen members

Practice

- ✓ chapter03 → resolve project
  - ~~getaddrinfo~~
  - getnameinfo
  - getservbyname
  - gethostbyname

```
D:₩github₩DiconServer₩chapter03₩Debug>resolve -n www.google.com

NOTE: Without the NI_NUMERICSERV flag (-f numericserv) getnameinfo will
      fail if the port information does not resolve to a well known service
      like ftp (21), etc.


Attempting to resolve the host: www.google.com
Service/port information is   : NULL
Requested address family      : NONE (default is AF_UNSPEC)
Requested socket type         : NONE
Requested protocol            : NONE
Requested flags to getnameinfo:

Hostname 'www.google.com' resolved to 1 addresses

[1] Numeric address resolved: 172.217.24.132
[1] Host:    nrt20s01-in-f4.1e100.net
[1] Service: 0


D:₩github₩DiconServer₩chapter03₩Debug>_
```

Practice: Find and Kill a process with port number

- ✓ Launch the tcpserver.exe in chap01
  - tcpserver uses port number 5150
- ✓ In the command line, find the process which uses port 5150
  - netstat -a -o | find "5150"
- ✓ Kill the process with taskkill.exe
  - taskkill /F /PID [process-id]

References

- ✓ http://www.winsocketdotnetworkprogramming.com/winsock2programming/winsock2advancedInternet3chap.html

MY **BRIGHT** FUTURE

DSU Dongseo University 동서대학교