

Socket Programming

Introduction (Part 2)

TA: Cristian Rodriguez

COMP 445 – Winter 2021

Agenda

- Understanding hostnames.
- Port numbers.
- TCP Socket Flow
- Blocking vs Non-Blocking
 - Selector
 - Select

Understanding hostnames and port numbers

- Hostnames are names that can be translated into IP addresses.
 - A single hostname can map to different IP addresses at different times.
 - Multiple hostnames can map to the same IP address.
- `nslookup` is the tool to translate hostnames into IP addresses.
 - `nslookup` stands for “name server look up”.
- What is the protocol behind this command?
 - DNS (Domain Name System) specified in RFCs 1034, 1035, 8499, 2929.
- It is possible to find the domain name that corresponds to an IP address; this process is known as reverse DNS.

Exercise 1:

- nslookup localhost

Server's name

Server's IP and port

Lookup information for localhost

```
cristian@cristian-VirtualBox:~$ nslookup localhost
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   localhost
Address: 127.0.0.1
Name:   localhost
Address: ::1
```

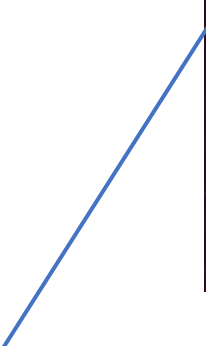
What is non-authoritative answer?

An answer is authoritative only if our DNS server has the complete zone file information for the domain. Normally, our DNS has a cache which represents the last authoritative answer it received when it made a similar query. When the cache information is passed as a response then it qualifies as non-authoritative.

Exercise 1:

- nslookup localhost

- Why the DNS server is my local machine?
 - Answer for Ubuntu users.
 - It is running `systemd-resolved` as a service.
- How it works?
 - `systemd-resolved` generates two configuration files.
 - `/run/systemd/resolve/stub-resolv.conf`
 - `/run/systemd/resolve/resolv.conf`
 - The former tells to send DNS queries to 127.0.0.53, then it forwards on,
 - The second one avoids querying 127.0.0.53 and uses the DNS server provided by DHCP.

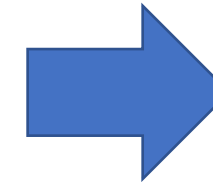
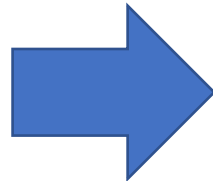


```
cristian@cristian-VirtualBox:~$ nslookup localhost
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   localhost
Address: 127.0.0.1
Name:   localhost
Address: ::1
```

Exercise 1:

- nslookup localhost
 - Let's check if there is a UDP process running at 127.0.0.53.



```
cristian@cristian-VirtualBox:~$ sudo netstat -tulpn
[sudo] password for cristian:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*                LISTEN      461/systemd-resolve
tcp        0      0 127.0.0.1:631           0.0.0.0:*                LISTEN      3385/cupsd
tcp        0      0 0.0.0.0:23              0.0.0.0:*                LISTEN      1020/inetd
tcp        0      0 0.0.0.0:17500            0.0.0.0:*                LISTEN      2255/dropbox
tcp        0      0 127.0.0.1:17600          0.0.0.0:*                LISTEN      2255/dropbox
tcp        0      0 127.0.0.1:17603          0.0.0.0:*                LISTEN      2255/dropbox
tcp6       0      0 :::21                   :::*                    LISTEN      946/vsftpd
tcp6       0      0 ::1:631                  :::*                    LISTEN      3385/cupsd
tcp6       0      0 :::17500                  :::*                    LISTEN      2255/dropbox
udp        0      0 127.0.0.53:53           0.0.0.0:*                461/systemd-resolve
udp        0      0 0.0.0.0:68              0.0.0.0:*                3279/dhclient
udp        0      0 0.0.0.0:17500            0.0.0.0:*                2255/dropbox
udp        0      0 0.0.0.0:5353             0.0.0.0:*                751/avahi-daemon: r
udp        0      0 0.0.0.0:43614            0.0.0.0:*                751/avahi-daemon: r
udp        0      0 0.0.0.0:631              0.0.0.0:*                3386/cups-browsed
udp6       0      0 :::57455                 :::*                    751/avahi-daemon: r
udp6       0      0 :::5353                  :::*                    751/avahi-daemon: r
```

Exercise 2:

- nslookup google.ca

```
cristian@cristian-VirtualBox:~$ nslookup google.ca
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   google.ca
Address: 172.217.1.163
Name:   google.ca
Address: 2607:f8b0:400b:809::2003
```

Exercise 3:

- nslookup concordia.ca

```
cristian@cristian-VirtualBox:~$ nslookup concordia.ca
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   concordia.ca
Address: 132.205.244.185
```


Exercise 4:

- `nslookup -type=ns concordia.ca`
 - What is `-type=ns`?
 - The NS record of a domain is a map of all name servers that are authoritative for that domain.

```
cristian@cristian-VirtualBox:~$ nslookup concordia.ca
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   concordia.ca
Address: 132.205.244.185

cristian@cristian-VirtualBox:~$ nslookup -type=ns concordia.ca
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
concordia.ca     nameserver = ns-a.concordia.ca.
concordia.ca     nameserver = ns1.zonerisq.ca.
concordia.ca     nameserver = ns-b.concordia.ca.
concordia.ca     nameserver = ns1.cc.umanitoba.ca.
concordia.ca     nameserver = ns2.zonerisq.ca.

Authoritative answers can be found from:
```

Exercise 5:

- `nslookup -type=mx concordia.ca`
 - What is `-type=mx`?
 - The MX record of a domain is a map of mail exchange servers for a domain.

```
cristian@cristian-VirtualBox:~$ nslookup concordia.ca
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   concordia.ca
Address: 132.205.244.185

cristian@cristian-VirtualBox:~$ nslookup -type=ns concordia.ca
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
concordia.ca    nameserver = ns-a.concordia.ca.
concordia.ca    nameserver = ns1.zonerisq.ca.
concordia.ca    nameserver = ns-b.concordia.ca.
concordia.ca    nameserver = ns1.cc.umanitoba.ca.
concordia.ca    nameserver = ns2.zonerisq.ca.

Authoritative answers can be found from:
```

Exercise 6:

- `nslookup -type=cname www.ibm.com`
 - What is `-type=cname`?
 - The MX record of a domain is an alias for some canonical name.

```
cristian@cristian-VirtualBox:~$ nslookup -type=cname www.ibm.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
www.ibm.com      canonical name = www.ibm.com.cs186.net.

Authoritative answers can be found from:

cristian@cristian-VirtualBox:~$
```

Exercise 7:

- Let's capture some DNS traffic (request).

```
‣ Frame 21: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0
‣ Ethernet II, Src: PcsCompu_f7:77:7b (08:00:27:f7:77:7b), Dst: Zte_4a:4a:20 (98:13:33:4a:4a:20)
‣ Internet Protocol Version 4, Src: 192.168.1.23, Dst: 192.168.1.1
‣ User Datagram Protocol, Src Port: 33781, Dst Port: 53
‣ Domain Name System (query)
  Transaction ID: 0x4e58
  ‣ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ‣ Queries
    ‣ concordia.ca: type A, class IN
  ‣ Additional records
    ‣ <Root>: type OPT
    \[Response In: 23\]
```

Exercise 7:

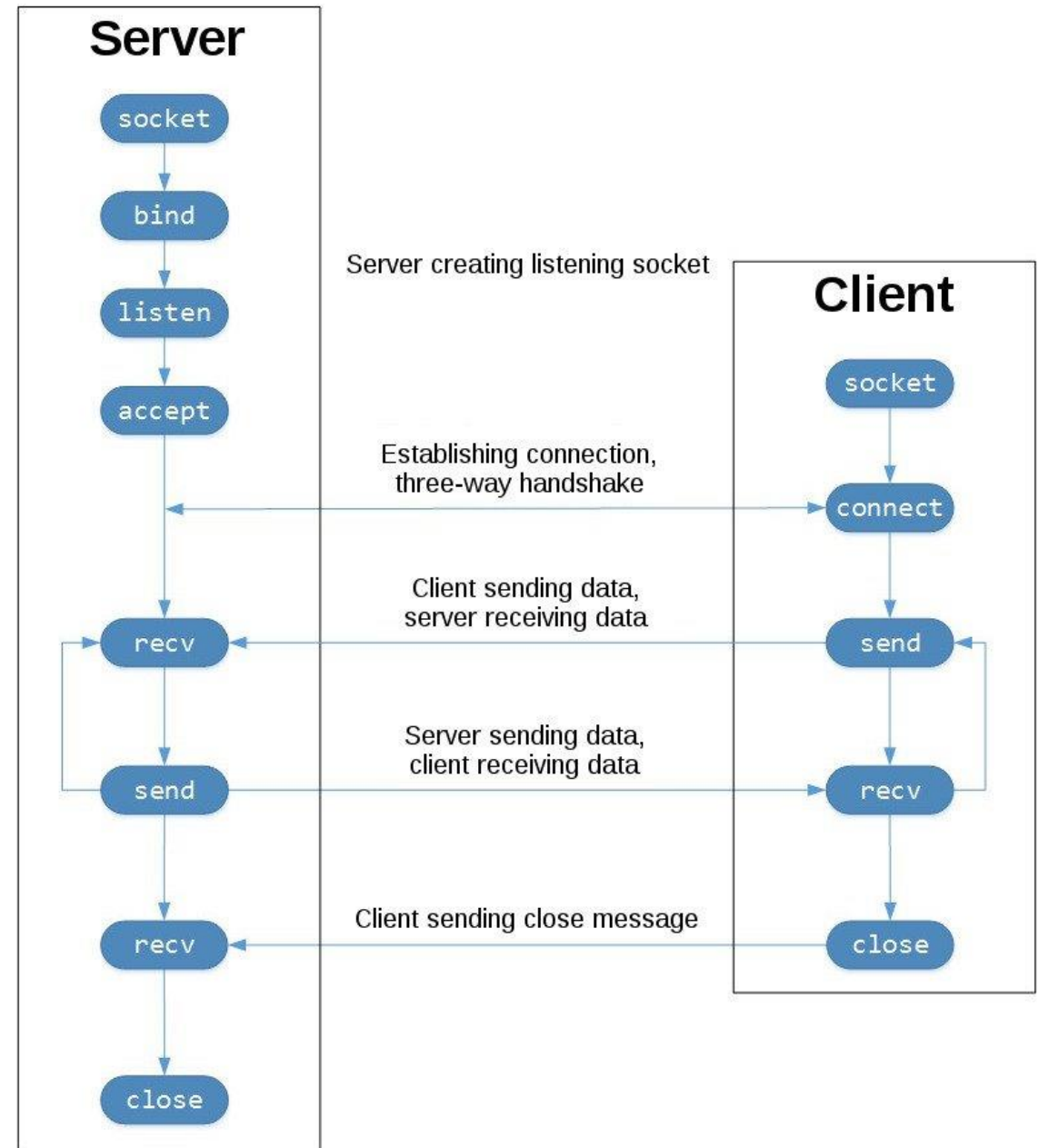
- Let's capture some DNS traffic (response).

```
▸ Frame 23: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface 0
▸ Ethernet II, Src: Zte_4a:4a:20 (98:13:33:4a:4a:20), Dst: PcsCompu_f7:77:7b (08:00:27:f7:77:7b)
▸ Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.23
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33781
▾ Domain Name System (response)
  Transaction ID: 0x4e58
  ▸ Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 1
▾ Queries
  ▸ concordia.ca: type A, class IN
▾ Answers
  ▸ concordia.ca: type A, class IN, addr 132.205.244.185
▾ Additional records
  ▸ <Root>: type OPT
\[Request In: 21\]
[Time: 0.031295583 seconds]
```

Port numbers

- A single machine might have multiple server applications that clients wish to connect to, so we need a way to direct traffic on the same network interface to different processes.
- Network interfaces have multiple ports identified by a 16-bit number from 0 to 65535.
- The well-know ports are numbered from 0 through 1023.
 - Examples?
 - FTP Data transfer:
 - FTP Command Control:
 - DNS:
 - HTTP:

TCP socket flow



Blocking vs Non-blocking

- Blocking means that a thread waits (without doing further work) until an event occurs.
 - When an incoming socket's buffer is empty, calling `read` blocks until data is available.
 - When the destination socket's buffer is full, calling `write` blocks until space is available.
- Do you recall the issue that we may have if we use blocking sockets?
 - Hint: deadlock concurrent programming.

Blocking vs Non-blocking

- There are two ways to implement.
 - Selectors: High-level I/O multiplexing.
 - Select: Waiting for I/O completion.

```
ready_to_read, ready_to_write, in_error = \
    select.select(
        potential_readers,
        potential_writers,
        potential_errs,
        timeout)
```

abstractmethod **register**(*fileobj*, *events*, *data=None*)

Register a file object for selection, monitoring it for I/O events.

fileobj is the file object to monitor. It may either be an integer file descriptor or an object with a `fileno()` method. *events* is a bitwise mask of events to monitor. *data* is an opaque object.

This returns a new `SelectorKey` instance, or raises a `ValueError` in case of invalid event mask or file descriptor, or `KeyError` if the file object is already registered.

abstractmethod **unregister**(*fileobj*) ¶

Unregister a file object from selection, removing it from monitoring. A file object shall be unregistered prior to being closed.

fileobj must be a file object previously registered.

This returns the associated `SelectorKey` instance, or raises a `KeyError` if *fileobj* is not registered. It will raise `ValueError` if *fileobj* is invalid (e.g. it has no `fileno()` method or its `fileno()` method has an invalid return value).

References:

<https://docs.python.org/3/library/select.html>

<https://docs.python.org/3/library/selectors.html>

Blocking vs Non-blocking

```
ready_to_read, ready_to_write, in_error = \
    select.select(
        potential_readers,
        potential_writers,
        potential_errs,
        timeout)
```

- `select.select()`: provides the I/O multiplexing.
- Parameters:
 - `potential_readers`: Sockets waiting for a condition when data is ready for read.
 - `potential_writers`: Sockets waiting for a condition when data is ready for write.
 - `potential_errs`: Sockets waiting for an exception condition.
 - `timeout`: Number of seconds to wait before coming out of blocking.
- Return values:
 - `ready_to_read`: All sockets that you might want to try reading.
 - `ready_to_write`: The sockets that you might want to try writing to.
 - `in_error`: The sockets that you might want to check for errors.

Blocking vs Non-blocking

- Let's review some code.
- Please remember, the provided code is just a small guide to help you start your assignment. Make sure to submit your own code.
- Exercise 8:
 - Play around with the provided code.
 - Please follow the instructions to understand how both implementations work.

Blocking vs Non-blocking

- Exercise 9:
 - ClientChat.py is blocking.
 - If there is enough time, then try to do your own non blocking version choosing the method that you think is appropriate.
 - Otherwise, feel free to leave.

Recap

- Understanding hostnames.
- Port numbers.
- TCP Socket Flow.
- Blocking vs Non-Blocking.
 - Selector
 - Select