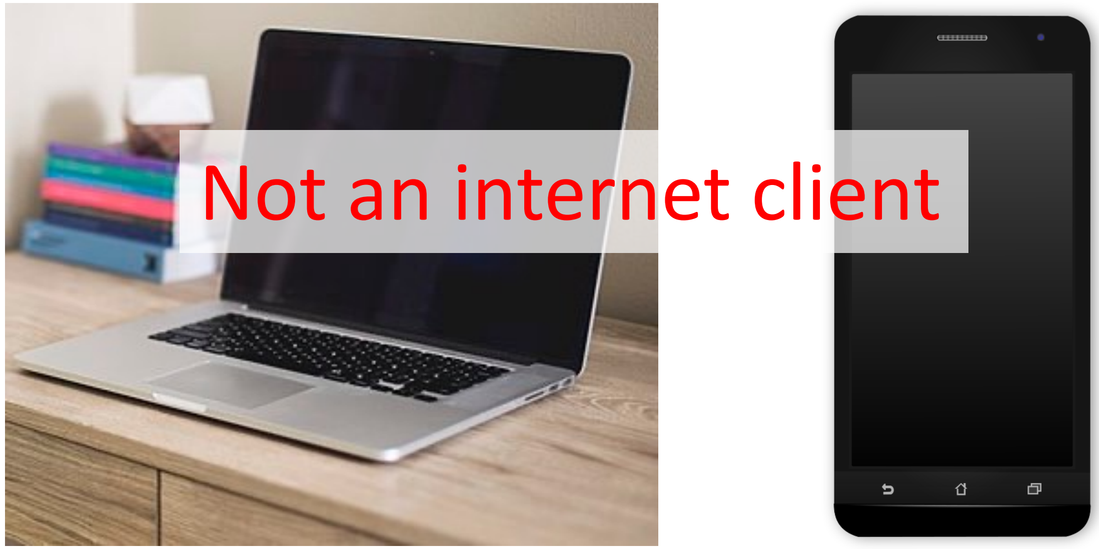
By the time you see this pptx you should have already seen this video:
https://www.youtube.com/watch?v=7_LPdttKXPc

Some considerations I disagree:

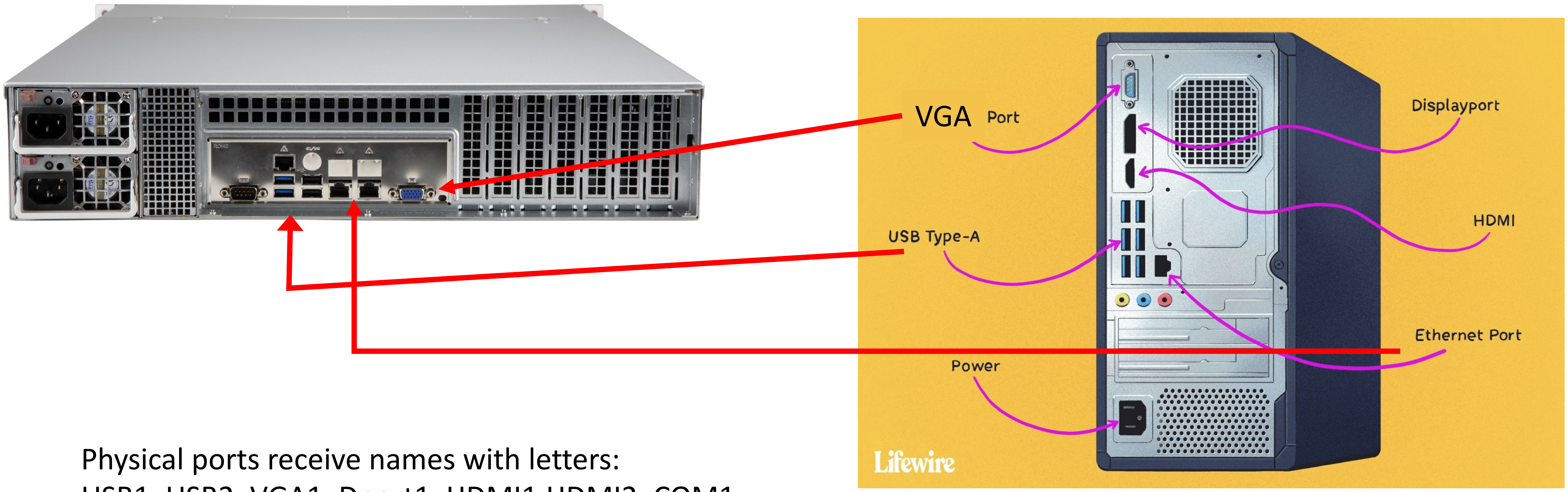
- In internet, all computers are created equal. If a computer has a routable IP it is connected to internet. It has the same “rights” as any other (where it is, how big it is, etc. It does not matter).
- What makes it a server or a client is not the type of computer it is, is the software currently in execution on it.
  If a raspberry pi has a routable IP and apache running on it, it is a web server, but if at the same time is executing curl to obtain data from other webs it is a client.

Not an internet server



Not an internet client

Internet: Client – Server architecture
Physical ports



Physical ports receive names with letters:
USB1, USB2, VGA1, Dport1, HDMI1,HDMI2, COM1…

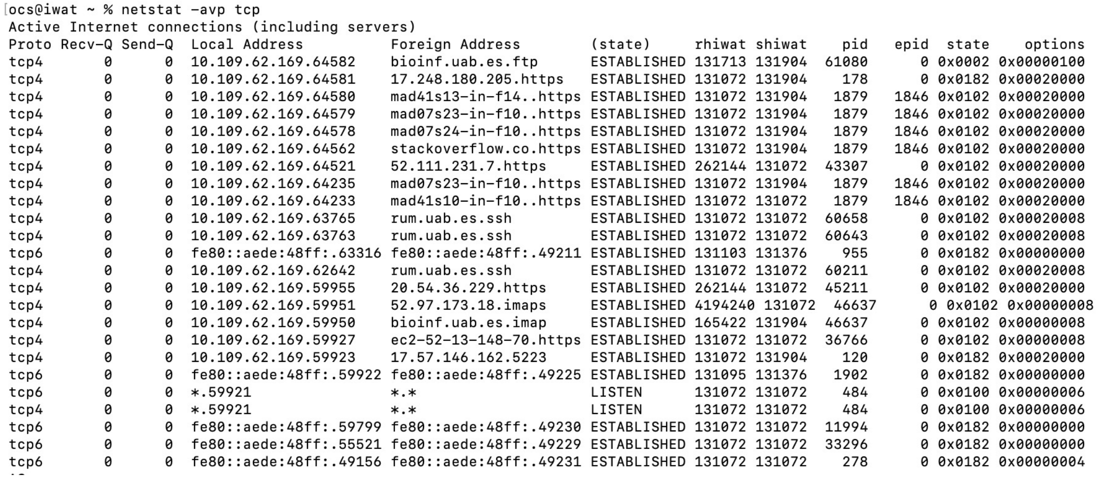
Internet: Client – Server architecture
Virtual ports

We need virtual ports to receive more connections than physical ports

Execute:

Mac: `netstat -avp tcp`
Linux: `netstat –tpa`

We can establish multiple connections at the same time through the same port.
In this case the network port (Ethernet port or Wi-Fi)



```
ocs@iwat ~ % netstat -avp tcp
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)     rhiwat shiwat    pid   epid  state    options
tcp4       0      0  10.109.62.169.64582    bioinf.uab.es.ftp      ESTABLISHED 131713 131904  61080      0 0x0002 0x00000100
tcp4       0      0  10.109.62.169.64581    17.248.180.205.https   ESTABLISHED 131072 131904    178      0 0x0182 0x00020000
tcp4       0      0  10.109.62.169.64580    mad41s13-in-f14..https ESTABLISHED 131072 131904   1879   1846 0x0102 0x00020000
tcp4       0      0  10.109.62.169.64579    mad07s23-in-f10..https ESTABLISHED 131072 131904   1879   1846 0x0102 0x00020000
tcp4       0      0  10.109.62.169.64578    mad07s24-in-f10..https ESTABLISHED 131072 131904   1879   1846 0x0102 0x00020000
tcp4       0      0  10.109.62.169.64562    stackoverflow.co.https ESTABLISHED 131072 131904   1879   1846 0x0102 0x00020000
tcp4       0      0  10.109.62.169.64521    52.111.231.7.https     ESTABLISHED 262144 131072  43307      0 0x0102 0x00020000
tcp4       0      0  10.109.62.169.64235    mad07s23-in-f10..https ESTABLISHED 131072 131904   1879   1846 0x0102 0x00020000
tcp4       0      0  10.109.62.169.64233    mad41s10-in-f10..https ESTABLISHED 131072 131072   1879   1846 0x0102 0x00020000
tcp4       0      0  10.109.62.169.63765    rum.uab.es.ssh         ESTABLISHED 131072 131072  60658      0 0x0102 0x00020008
tcp4       0      0  10.109.62.169.63763    rum.uab.es.ssh         ESTABLISHED 131072 131072  60643      0 0x0102 0x00020008
tcp6       0      0  fe80::aede:48ff:.63316 fe80::aede:48ff:.49211 ESTABLISHED 131103 131376    955      0 0x0182 0x00000000
tcp4       0      0  10.109.62.169.62642    rum.uab.es.ssh         ESTABLISHED 131072 131072  60211      0 0x0102 0x00020008
tcp4       0      0  10.109.62.169.59955    20.54.36.229.https     ESTABLISHED 262144 131072  45211      0 0x0102 0x00020000
tcp4       0      0  10.109.62.169.59951    52.97.173.18.imaps     ESTABLISHED 4194240 131072  46637      0 0x0102 0x00000008
tcp4       0      0  10.109.62.169.59950    bioinf.uab.es.imap     ESTABLISHED 165422 131904  46637      0 0x0102 0x00000008
tcp4       0      0  10.109.62.169.59927    ec2-52-13-148-70.https ESTABLISHED 131072 131072  36766      0 0x0102 0x00000008
tcp4       0      0  10.109.62.169.59923    17.57.146.162.5223     ESTABLISHED 131072 131904    120      0 0x0182 0x00020000
tcp6       0      0  fe80::aede:48ff:.59922 fe80::aede:48ff:.49225 ESTABLISHED 131095 131376   1902      0 0x0182 0x00000000
tcp6       0      0  *.59921                *.*                    LISTEN      131072 131072    484      0 0x0100 0x00000006
tcp4       0      0  *.59921                *.*                    LISTEN      131072 131072    484      0 0x0100 0x00000006
tcp6       0      0  fe80::aede:48ff:.59799 fe80::aede:48ff:.49230 ESTABLISHED 131072 131072  11994      0 0x0182 0x00000000
tcp6       0      0  fe80::aede:48ff:.55521 fe80::aede:48ff:.49229 ESTABLISHED 131072 131072  33296      0 0x0182 0x00000000
tcp6       0      0  fe80::aede:48ff:.49156 fe80::aede:48ff:.49231 ESTABLISHED 131072 131072    278      0 0x0182 0x00000004
```

Internet: Client – Server architecture
Services <-> ports

Different services use different ports by default

Mail: 25 (587 for clients)

imap 143 (imaps: 993)

www A.K.A. http 80 (https 443)

mysql: 3306

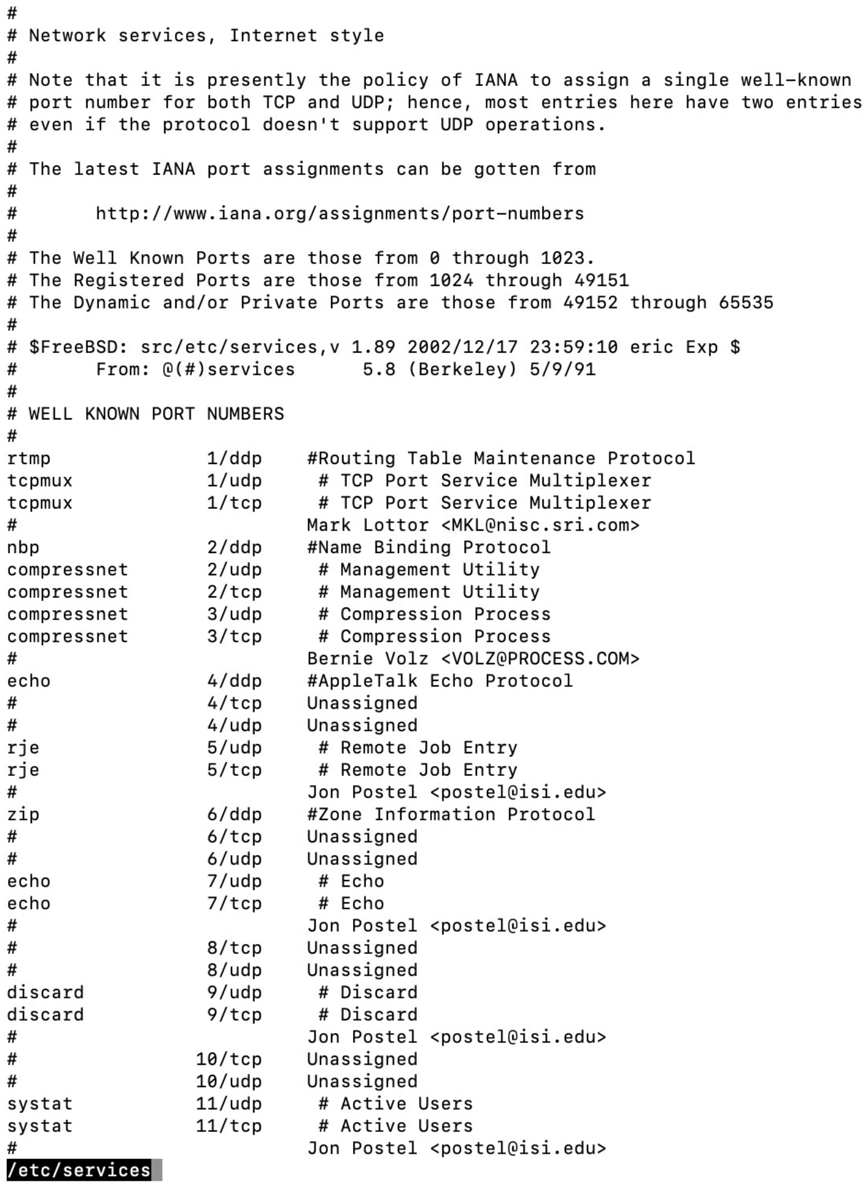
ftp: 21

Execute:

```
less /etc/services
```

THERE ARE MORE SERVICES RUNNING ON THE INTERNET BESIDES WWW!!!

The origins of Internet are in ARPANET, born in 1969. As internet in 1983. WWW was not created until 1991.



```
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
#
# The latest IANA port assignments can be gotten from
#
#       http://www.iana.org/assignments/port-numbers
#
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# $FreeBSD: src/etc/services,v 1.89 2002/12/17 23:59:10 eric Exp $
#       From: @(#)services      5.8 (Berkeley) 5/9/91
#
# WELL KNOWN PORT NUMBERS
#
rtmp              1/ddp    #Routing Table Maintenance Protocol
tcpmux            1/udp     # TCP Port Service Multiplexer
tcpmux            1/tcp     # TCP Port Service Multiplexer
#                          Mark Lottor <MKL@nisc.sri.com>
nbp               2/ddp    #Name Binding Protocol
compressnet       2/udp     # Management Utility
compressnet       2/tcp     # Management Utility
compressnet       3/udp     # Compression Process
compressnet       3/tcp     # Compression Process
#                          Bernie Volz <VOLZ@PROCESS.COM>
echo              4/ddp    #AppleTalk Echo Protocol
#                 4/tcp    Unassigned
#                 4/udp    Unassigned
rje               5/udp     # Remote Job Entry
rje               5/tcp     # Remote Job Entry
#                          Jon Postel <postel@isi.edu>
zip               6/ddp    #Zone Information Protocol
#                 6/tcp    Unassigned
#                 6/udp    Unassigned
echo              7/udp     # Echo
echo              7/tcp     # Echo
#                          Jon Postel <postel@isi.edu>
#                 8/tcp    Unassigned
#                 8/udp    Unassigned
discard           9/udp     # Discard
discard           9/tcp     # Discard
#                          Jon Postel <postel@isi.edu>
#                10/tcp    Unassigned
#                10/udp    Unassigned
systat           11/udp     # Active Users
systat           11/tcp     # Active Users
#                          Jon Postel <postel@isi.edu>
/etc/services
```

Internet: Client – Server architecture
Services: ftp

Execute in your terminal:

```
ftp ftp.ncbi.nlm.nih.gov
```

Use the following credentials (standard gest user):

User: `anonymous`

Password: [any email address]

In the new prompt, type the following command:

```
help
```

(now look around the NCBI ftp server and download something)

ftp is an internet service, older than www, designed to transfer files between computers connected to internet.
Still today many databases allow us to download (and upload) files using this services. Especially useful for download full databases.

You have used a software (ftp) which acts as a **client** and runs in your **local computer** and connects to another software (ftpd) which acts as a **server** and runs in a **remote computer.**

They communicate each other using the “File Transfer Protocol” (FTP)

Of course, there are FTP clients with GUI (Filezilla, cyberduck, etc). Apart from that, all browsers were able to connect and download files from FTP servers until 2021 (Please, ask me why, or… better don’t)

Internet: Client – Server architecture
Services: www

Open a browser and type (exactly as I'm writing it!!)
http://esci.upf.edu
What has happened? Are you viewing the URL you typed?
Use, whatever option your browser has, to view the ”page source”
Keep the source open and go to this other addresses:
http://esci.upf.edu:80
http://esci.upf.edu:443
https://www.esci.upf.edu:443
http://esci.upf.edu:[any nmber here]

Now, in the command line type:

```
wget -O- "http://esci.upf.edu"|\less
```

You are connected to different ports using different protocols, so the answer is different. (you can check the connections using the netstat command previously mentioned)
Your browser and wget are **clients.** They run in your **local computer** and connect to another software (nginx) which acts as a **server** and runs in the **remote computer.**

They communicate each other using the “Hyper Text Transfer Protocol” (HTTP) or HTTP Secure (HTTPS)
Your browser, apart of being able to communicate with the server, is capable to interpret the ”html” language (and javascript and …) and ”draw” the web page you see. `wget` only communicates and delivers the information transferred.

Internet: Client – Server architecture
Web services

You can setup a web server or a web application specially designed not to receive communications from browsers but from other programs. In that case they are call web services, the information retrieved from them is not designed to be "drawn" in a browser. It is designed to be read by other programs.

You will usually receive flat text formatted as:

- Tab separated columns (or csv): http://rest.kegg.jp/list/pathway/sml
- xml formatted data https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nuccore&id=NC_000908.2&rettype=gbwithparts
- Json formatted data https://www.ebi.ac.uk/pdbe/api/mappings/1cbs
- Custom-made formatted data (e.g.: gbff from ncbi, EMBLtxt from uniport, fasta, etc)
  https://rest.uniprot.org/uniprotkb/stream?compressed=false&format=fasta&query=(xref%3Apfam-PF05494)%20AND%20(taxonomy_id%3A40324)

The simplest way to send data to a server is through the URL, the server reads the different parts and returns information according to it. This type of web services are called "REST"
There are others (eg: SOAP) but they are more complex and difficult to use and learn so they are not as common in bioinformatics as REST

Internet: Client – Server architecture
Other internet services: example: mysql

Remember: www is not the only service that runs on internet.

- **ftp** could be used to download complete databases and process s them locally. This is something very common in bioinformatics for several reasons:
  - If many many queries must be made you avoid network lag.
  - If many many queries must be made you avoid being banned from the DataBase. (Ask me about that)
  - If the APIs provided by the DB do not allow or do not have the kind of query you want to perform.
  - The main disadvantage of that is that you will have to take care to update the DB yourself.
  - But at the same time this is and advantage since you ensure reproducibility (your results don't change due to an update you don't control)

- Others could be extremely useful for bioinformaticians.
  - **MySQL** as an example: you can connect directly to a mysql DB containing biological information and perform queries.

Execute this in your command line (mysql client must be installed):

```
mysql --host=ensembldb.ensembl.org --port=3306 --user=anonymous -D homo_sapiens_rnaseq_100_38 -e "select * from gene where gene_id= '1';"
```

Mini project for groups:
Do a script that uses the following tools and web services to perform something useful:

NCBI E-utilities
NCBI Entrez direct
Uniprot REST API
KEGG REST API
PDB REST API
SIFTS REST API