

# Week 11

DATS 6450 – FOUNDATIONS OF COMPUTER SCIENCE

# Git/GitHub Wrap-up

- <https://guides.github.com/introduction/git-handbook/>
  - Reviews almost everything we talked about
  - Example workflows for:
    - Contributing to an existing repository
    - Starting a new repository and publishing to GitHub
    - Contributing to an existing branch on GitHub
  - Collaborator model vs fork and pull model
  - Learning guides
- Extra resource on good enough practices in scientific computing:
  - <https://arxiv.org/pdf/1609.00037.pdf>

# Overview – Rest of the Semester

- 11/07
  - Markdown, ssh/scp, cloud/networking, project
- 11/14
  - Anaconda, conda, packaging your code, brew/apt-get/yum, virtual env
- 11/21 – Thanksgiving Break (No Class)
- 11/28
  - Error handling, exception raising, assert, reading error messages, IDEs/debugging
- 12/05
  - Student vote
    - Example: pyspark
  - Project due

# Telling people about your work

- Two common ways to document your work are:
  - README files
    - Quick and simple way for other users to learn more about your work.
  - Wikis
    - help you present in-depth information about your project in a useful way.
- Every project should have a README because it is the first thing people will see when they visit your repository
- Documenting your projects on GitHub:
  - <https://guides.github.com/features/wikis/>

# Project README files

- READMEs generally follow one format in order to immediately orient developers to the most important aspects of your project.
- Project Name
  - Your project's name is the first thing people will see upon scrolling down to your README, and is included upon creation of your README file.
- Description
  - A description of your project follows.
  - A good description is clear, short, and to the point. Describe the importance of your project, and what it does.

# Project README files

- Table of Contents
  - Optionally, include a table of contents in order to allow other people to quickly navigate especially long or detailed READMEs.
- Installation
  - If your repo is a Python package, then you should include installation instructions in your repo
  - Tell other users how to install your project locally. Optionally, include a gif to make the process even more clear for other people.

# Project README files

- Usage
  - The next section is usage, in which you instruct other people on how to use your project after they've installed it.
  - This would also be a good place to include screenshots of your project in action.
- Contributing
  - Larger projects often have sections on contributing to their project, in which contribution instructions are outlined.
  - Sometimes, this is a separate file. (CONTRIBUTING.md)
    - If you have specific contribution preferences, explain them so that other developers know how to best contribute to your work. To learn more about how to help others contribute, check out GitHub's guide for [setting guidelines for repository contributors](#).

# Project README files

- Credits
  - Include a section for credits in order to highlight and link to the authors of your project.
- License
  - Finally, include a section for the license of your project. For more information on choosing a license, check out GitHub's [licensing guide](#)!
  - Let's users know the terms of service for using your code and copyright info
  - If you find software that doesn't have a license, that generally means you have no permission from the creators of the software to use, modify, or share the software. Although a code host such as GitHub may allow you to view and fork the code, this does not imply that you are permitted to use, modify, or share the software for any purpose.

# Project README files

- Your README should contain only the necessary information for developers to get started using and contributing to your project. Longer documentation is best suited for wikis, outlined below.
- README files are usually written in one of two different file formats
  - Markdown (.md)
    - a lightweight and easy-to-use syntax for styling all forms of writing on the web, especially GitHub.
  - reStructured Text (.rst)
    - a file format for textual data used primarily in the Python programming language community for technical documentation.

# Examples of good documentation

- spaCy
  - <https://github.com/explosion/spaCy>
- dplyr
  - <https://github.com/tidyverse/dplyr>

# Markdown

- Markdown is a lightweight markup language designed to be converted to html and a variety of other formats using a simple plain text syntax
- What is a markup language?
  - a computer **language** that uses tags to define elements within a document. It is human-readable, meaning **markup** files contain standard words, rather than typical programming syntax
  - The most common forms are html and xml

# Markdown

- Where does it pop up?
  - GitHub project documentation or reporting
  - Slack
  - R markdown
  - Jupyter/Databricks notebooks

# Markdown

- Walkthrough together:  
<https://guides.github.com/features/mastering-markdown/>
- Exercise:
  - <https://lab.github.com/githubtraining/communicating-using-markdown>
- More than you'll probably ever need to know how to do in markdown:
  - <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- Need to insert equations?
  - <http://users.dickinson.edu/~richesod/latex/latexcheatsheet.pdf>

# Basic Networking Intro

# Outline

- Basic pieces of the internet
- IP Address, MAC Address
- Useful network-related unix commands
- What is Secure Shell?
  - Features of Secure Shell
  - Security Mechanism of Secure Shell
- SSH1 vs SSH2
- How to copy files to/from a remote server



# The pieces of the internet

What is the internet?

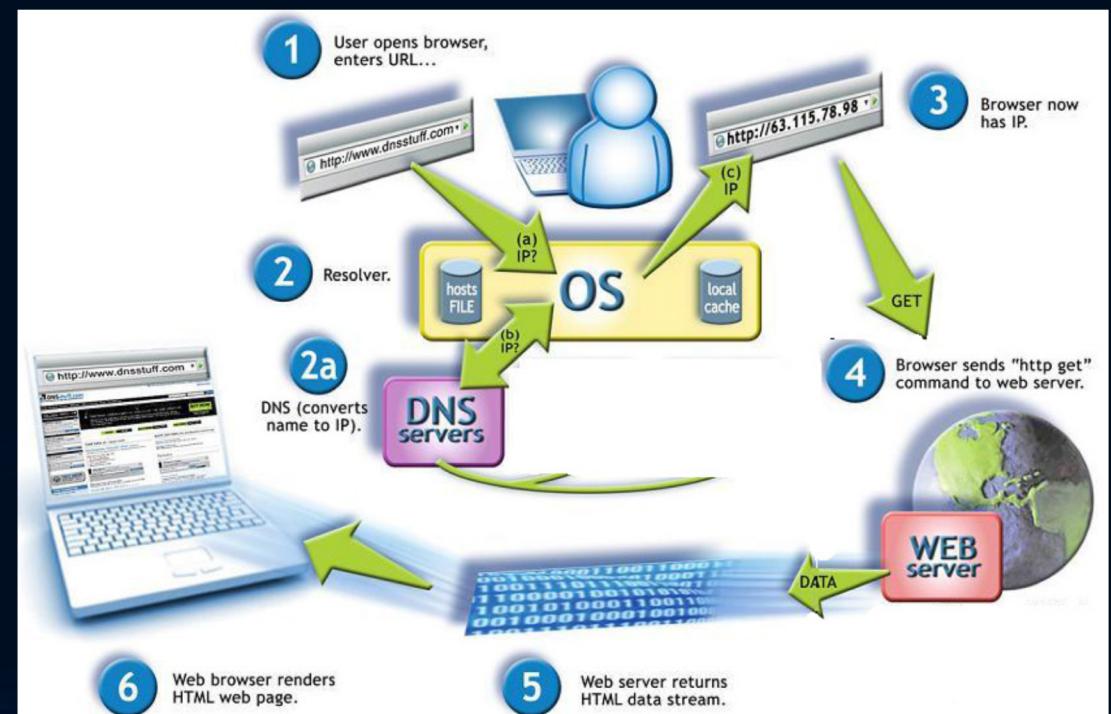
Answer: a bunch of cables

<https://www.submarinecablemap.com/#/>

Pieces of the internet:

- Router
- Server
- Modem
- Network Adapter
- Switch
- IP Address/Port
- MAC Address

It's kind of complicated



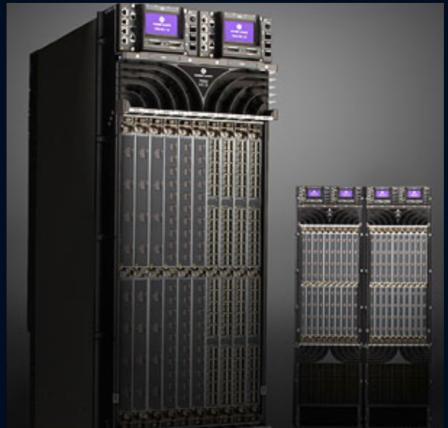
# The pieces of the internet

## Router

A router is a networking device that forwards data packets between computer networks.

Routers perform the traffic directing functions on the internet. Data sent through the internet (such as a web page, which is just html/css/javascript code) is transmitted in the form of data packets. A packet is typically forwarded from one router to another

**\*\*if you're a nerd like me, check out DD-WRT and/or OpenWrt (linux-based firmware for routers) this allows a lot of customization, you can turn old routers into wifi-repeaters or wifi-extenders to improve signal coverage in your house**



# The pieces of the internet

## Modem

A cable modem is a type of network bridge that provides bi-directional data communication via radio frequency channels on a coax or RFoG infrastructure.

Cable modems are primarily used to deliver broadband internet access in the form of cable internet, taking advantage of the high bandwidth of a coax and/or RFoG network



# The pieces of the internet

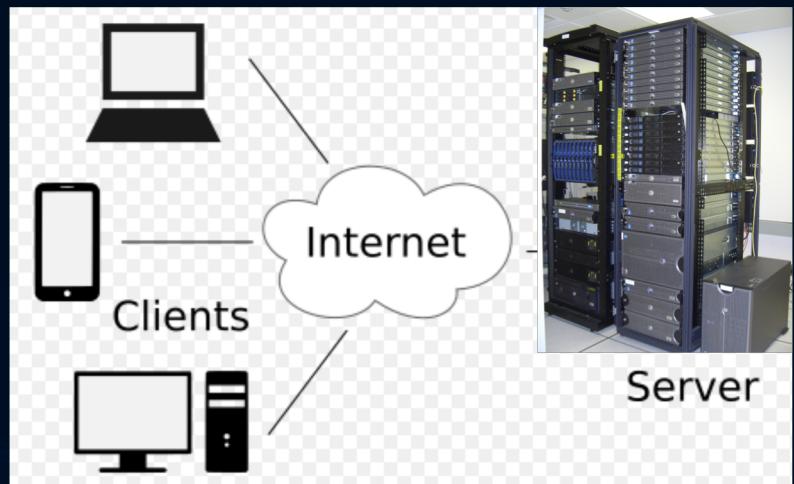
## Server

In computing, a server is a computer that provides functionality for other programs or devices, called "clients". This architecture is called the client–server model, and a single overall computation is distributed across multiple processes or devices.

Servers can provide various functionalities, often called "services", such as sharing data or resources among multiple clients, or performing computation for a client.

A single server can serve multiple clients, and a single client can use multiple servers.

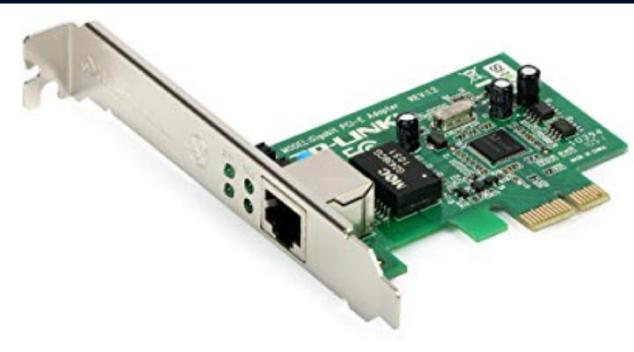
**Answer: a server is... someone else's computer**



# The pieces of the internet

## Network Adapter

A network interface controller (NIC, also known as a network interface card, network adapter, LAN adapter or physical network interface) is a piece of computer hardware that physically (or wirelessly) connects a device to a network.



## 802.11 WIFI Standards



# The pieces of the internet

## Switch

A computer networking device that connects multiple devices together and handles packet forwarding (packet switching) between them.



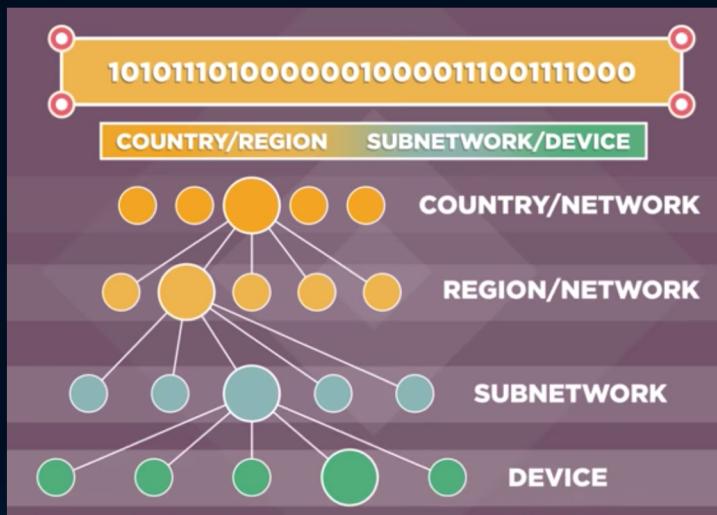
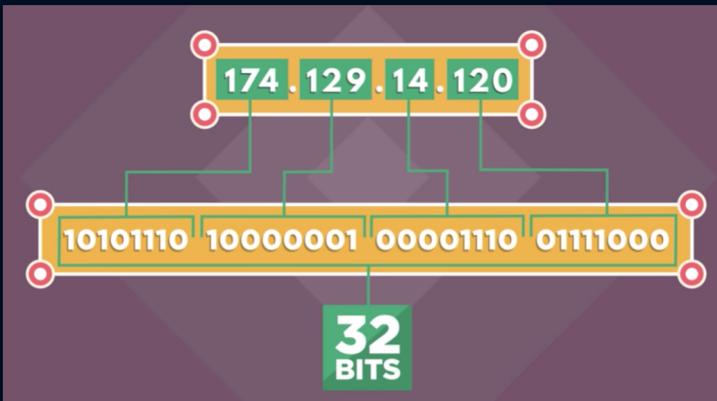
# The pieces of the internet

## IP Address

An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.

An IP address serves two principal functions: host or network interface identification and location addressing.

IP addresses are assigned to a host either dynamically at the time of booting, or permanently by fixed configuration of the host hardware or software.



## Internet protocol suite

### Application layer

BGP · **DHCP** · DNS · FTP · HTTP · HTTPS · IMAP · LDAP · MGCP · MQTT · NNTP · NTP · POP · ONC/RPC · RTP · RTSP · RIP · SIP · SMTP · SNMP · SSH · Telnet · TLS/SSL · XMPP · [more...](#)

### Transport layer

TCP · UDP · DCCP · SCTP · RSVP · [more...](#)

### Internet layer

IP (IPv4 · IPv6) · ICMP · ICMPv6 · ECN · IGMP · IPsec · [more...](#)

### Link layer

ARP · NDP · OSPF · Tunnels (L2TP) · PPP · MAC (Ethernet · DSL · ISDN · FDDI) · [more...](#)

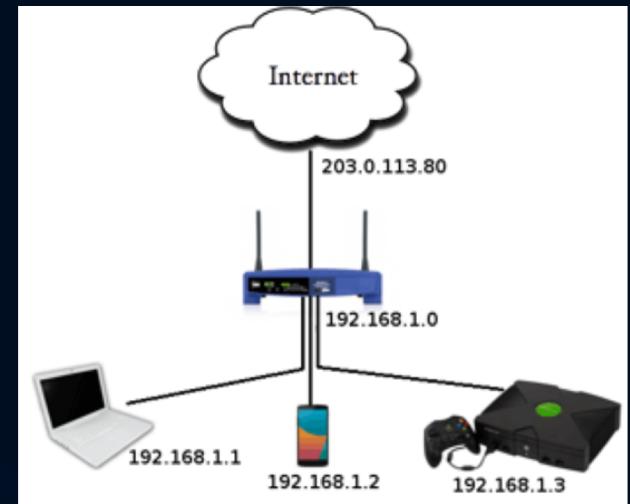
# The pieces of the internet

IP Address cont.

Persistent configuration is also known as using a **static IP address**. In contrast, when a computer's IP address is assigned newly each time it restarts, this is known as using a **dynamic IP address**.

The typical default behavior for a home router will be to function as a DHCP server, to assign a local IP address to each device connected to the network.

The Dynamic Host Configuration Protocol (DHCP) is a network management protocol where a DHCP server dynamically assigns an IP Address and other network configuration parameters to each device on a network so they can communicate with other IP networks.



# The pieces of the internet

IP Address cont.

There's no place like  
**127.0.0.1**

There are a number of "reserved IP addresses"

[https://en.wikipedia.org/wiki/Reserved\\_IP\\_addresses](https://en.wikipedia.org/wiki/Reserved_IP_addresses)

Another point to keep in mind is the fact that you can accurately be geo-located by your IP Address (only):

"**IP geolocation** is the mapping of an IP address to the geographic location of the internet from the connected device. By geographically mapping the IP address, it provides you with location information such as the country, state, city, zip code, latitude/longitude, ISP, area code, and other information." - wikipedia

<https://tools.keycdn.com/geo>

# The pieces of the internet

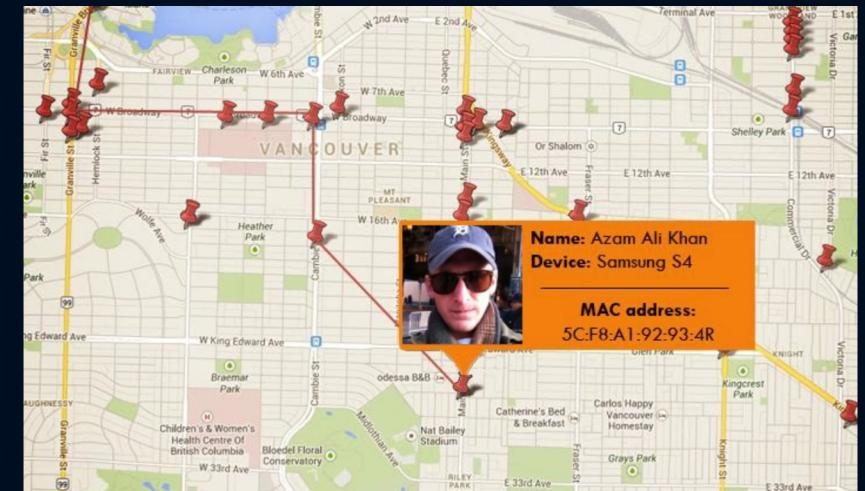


## MAC Address

A media access control address (MAC address) of a device is a unique identifier assigned to a network interface controller (NIC) for communications at the data link layer of a network segment.

A MAC may be referred to as the burned-in address (BIA). It may also be known as an Ethernet hardware address (EHA), hardware address or physical address.

MAC addresses are most often assigned by the manufacturer of a NIC and are stored in its hardware, such as the card's read-only memory (ROM) or in the device's firmware.

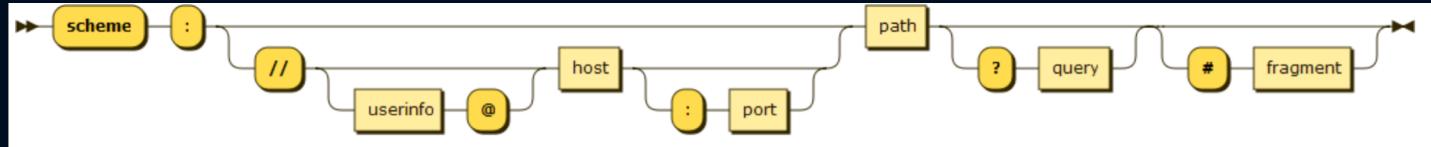


According to Edward Snowden, the US NSA has a system that tracks the movements of everyone in a city by monitoring the MAC addresses of their electronic devices.

# URL

A Uniform Resource Locator (URL a.k.a. "web address") is a reference to a web resource that specifies its location on a network and a mechanism for retrieving it.

URI = scheme:[//authority]path[?query][#fragment]



[https://www.google.com/search?q=what+is+a+url%3F&rlz=1C5CHFA\\_enUS766US766&oq=what+is+a+url%3F&aqs=chrome..69i57.2681j0j7&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=what+is+a+url%3F&rlz=1C5CHFA_enUS766US766&oq=what+is+a+url%3F&aqs=chrome..69i57.2681j0j7&sourceid=chrome&ie=UTF-8)

# DNS

An often-used analogy to explain the Domain Name System is that it serves as the "phone book" for the Internet by translating human-friendly computer hostnames into their corresponding IP addresses.

For example, the domain name `www.example.com` translates to the addresses `93.184.216.34` (IPv4) and `2606:2800:220:1:248:1893:25c8:1946` (IPv6). Unlike a phone book, DNS can be quickly updated, allowing a service's location on the network to change without affecting the end users, who continue to use the same hostname. Users take advantage of this when they use meaningful URL's without having to know how the computer actually locates the services.

# Useful Unix Networking Commands

Command	Description
<code>whoami</code>	Outputs your username
<code>passwd</code>	Changes your password
<code>hostname</code>	Outputs the computer's name/address
<code>w</code> or <code>finger</code>	See info about people logged into the server
<code>ifconfig</code>	View network settings (IP/MAC addresses, etc)
<code>ssh</code>	Connect to a remote server
<code>sftp</code> or <code>scp</code>	Transfer files to/from a remote server (after starting sftp connection, use get and put)
<code>wget</code>	Download from a URL to a file
<code>curl</code>	Download from a URL and output to console ( <code>stdout</code> ) – more options
<code>*sshfs</code>	Mount and interact with remote directories and files, once mounted, use remote directories as if they were local
<code>*netstat</code>	Shows the <i>network status</i> , active TCP connections, ports the computer is listening on, IP routing tables, etc.

# What is SSH

SSH is a widely used alternative to telnet and rlogin (old school) - utilities to login through a network in Unix environment. The problem with these tools is that they aren't secure – the user's login name and password transmitted as clear text

The Secure SHell protocol was created by Tatu Ylönen and others to provide encrypted data transfers between remote machines

Any Linux or macOS user can SSH into their remote server directly from the terminal window. Windows users can take advantage of SSH clients like Putty or Cygwin (or the more superior Gitbash)

You can execute shell commands in the same manner as you would if you were physically operating the remote computer.



# The SSH Command

The SSH command consists of 3 distinct parts:

- `ssh {user}@{host}`
- `ssh brentskoumal@login.colonialone.gwu.edu`
- `{user}` represents the account you want to access.
  - For example, you may want to access the **root** user, which is basically synonymous for system administrator with complete rights to modify anything on the system.
- `{host}` refers to the computer you want to access.
  - This can be an IP Address (**e.g. 244.420.23.19**) or a domain name (e.g. [www.xyzdomain.com](http://www.xyzdomain.com))

# The SSH Command

- `ssh brentskoumal@login.colonialone.gwu.edu`
- When you hit enter after the ssh command structured like the one above, you will be prompted to enter the password for the requested account.
- When you type it in, nothing will appear on the screen, but your password is, in fact being transmitted. Once you're done typing, hit enter once again. If your password is correct, you will be greeted with a remote terminal window.

# How does SSH work with encryption?

SSH makes use of a client-server model to allow for authentication of two remote systems and encryption of the data that passes between them.

SSH operates on TCP port 22 by default

The host (server) listens on port 22 for incoming connections. It organizes the secure connection by authenticating the client and opening the correct shell environment if the verification is successful.



# How does SSH work with encryption?

The client must begin the SSH connection by initiating the TCP handshake with the server, ensuring a secured symmetric connection, verifying whether the identity displayed by the server match previous records (typically recorded in an RSA key store file), and presenting the required user credentials to authenticate the connection.

There are two stages to establishing a connection: first both the systems must agree upon encryption standards to protect future communications, and second, the user must authenticate themselves. If the credentials match, then the user is granted access.



# SSH Keys

SSH keys serve as a means of identifying yourself to an SSH server using public-key cryptography and challenge-response authentication.

One immediate advantage this method has over traditional password authentication is that you can be authenticated by the server without ever having to send your password over the network.

Anyone eavesdropping on your connection will not be able to intercept and crack your password because it is never actually transmitted. Additionally, using SSH keys for authentication virtually eliminates the risk posed by brute-force password attacks by drastically reducing the chances of the attacker correctly guessing the proper credentials.



# SSH1 vs SSH2

SSH2 has separate transport, authentication, and connection protocols. SSH1 has one monolithic protocol

SSH2 has strong cryptographic integrity check using MAC, SSH1 has weak checking using CRC-32

SSH2 supports any number of session channels per connection (including none), SSH1 exactly one

Servers running SSH-2 can also run SSH-1 to take care of clients running SSH-1

SSH2 allows more than one form of authentication per session, SSH1 allows only one.

# Copying files to/from a remote server

## SFTP

### **Option 1: using secure file transfer protocol (sftp):**

sftp <psuid>@<servername>

#connect to server to  
transfer or download files

Once logged in via sftp:

put <filename>

#copies file to the server  
#copies all files in folder on  
local directory (i.e. your  
computer) to server

put -r <directoryname>

#copies file from server to  
local directory

get <filename>

#copies all files in directory  
to local directory

get -r <directoryname>

#tells you current local  
directory

lpwd

#change local directory

lcd <local directory path/name>

# Copying files to/from a remote server

## SCP

### **Option 2: using secure copy protocol (scp)**

```
scp <localfilename> <psuid>@<servername>:~/
```

#copies file from  
local directory to  
server

```
scp -r <localdirname> <psuid>@<servername>:~/
```

#copies all files in  
local directory to  
server

```
scp <psuid>@<servername>:~/ <localfilename>
```

#copies file from  
server to local  
directory

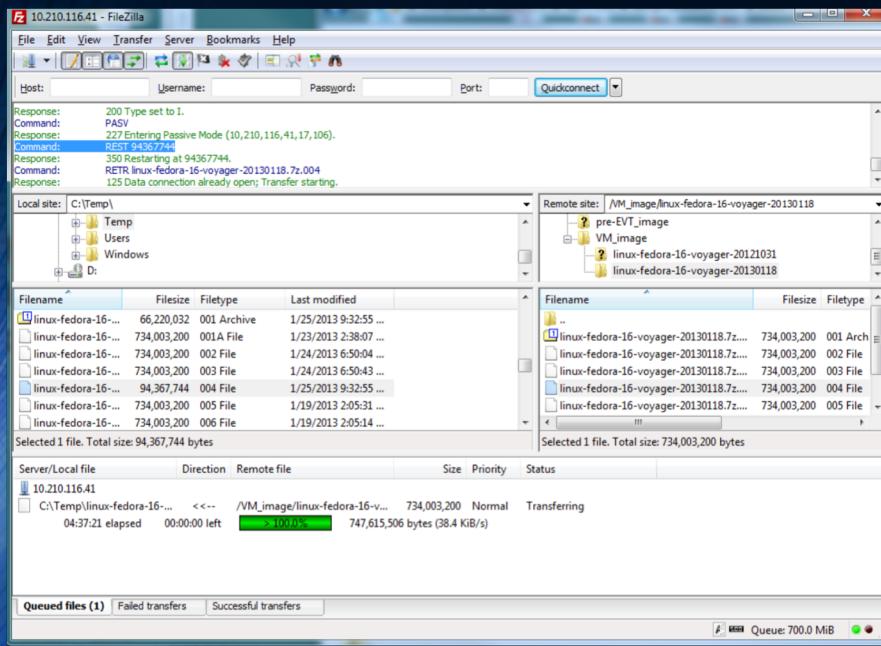
```
scp -r <psuid>@<servername>:~/ <localdirname>
```

#copies all files in directory  
from server to local  
directory

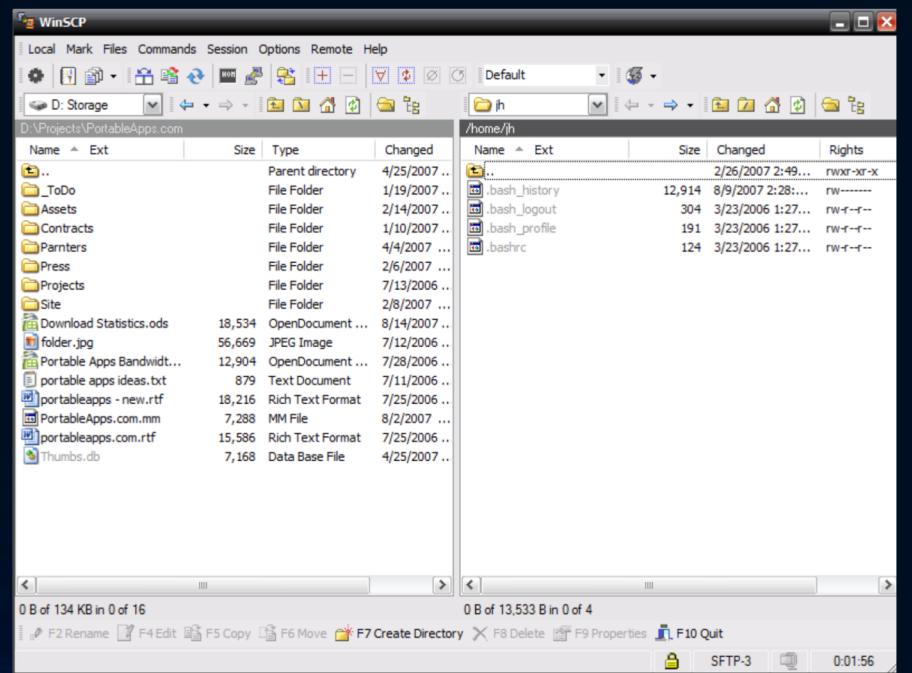
# Copying files to/from a remote server

## Filezilla / Winscp

<https://filezilla-project.org/>



<https://winscp.net/eng/download.php>



(for the GUI inclined...)