

Networking

networking lecture & practice for CSC420 Operating Systems Spring 2022 Lyon College

README

- This file accompanies lectures on the shell and `bash(1)`. To gain practice, you should type along in your own Org-mode file. You have to have Emacs and my `.emacs` file installed on your PC or the Pi you're working with.
- This section is based on chapter 16 of Shotts, The Linux Command Line (2e), NoStarch Press (2019) - "Networking".
- To make this easier, use the auto expansion (`<s>`). This will only work if you have my `.emacs` file ([from GDrive](#)) installed and the `org-tempo` library loaded.
- Add the following two lines at the top of your file, and activate each line with `C-c C-c` (this is confirmed in the echo area as `Local setup has been refreshed`):

```
#+PROPERTY: header-args:bash :results output :exports both
```

- Remember that `C-M-\` inside a code block indents syntactically (on Windows, this may only work if you have a marked region - set the mark with `C-SPC`).
- To **not** see the emphatic characters like `~` or `*` or `/` in the Org file text, run the following code chunk (or put the code in your `/.emacs` file): if successful, you should see "t" in the minibuffer.

```
(setq-default org-hide-emphasis-markers t)
```

If you don't put it in your `/.emacs` file, the command will only work for the current Emacs session.

- If you have difficulty distinguishing the code blocks from the documentation, change your Emacs theme with `M-x custom-themes` - `Leuven` is great if you like a light theme, or `Manoj-dark` if you like it dark.

Overview

- When it comes to networking, there is nothing that cannot be done with Linux.
- Linux is used to build networking systems like firewalls, routers, name servers, network-attached storage (NAS) boxes, etc.
- There is a vast number of commands - we focus on the most frequently used ones to monitor networks, transfer files and facilitate remote work:

COMMAND	MEANING
<code>ping</code>	Send an echo request to network hosts
<code>traceroute</code>	Print the route packets trace to a host
<code>ip</code>	Show/manipulate routing, devices, tunnels
<code>netstat</code>	Print network connections, routing tables, interface statistics, masquerade connections and multicast memberships
<code>ftp</code>	Internet file transfer program

COMMAND	MEANING
irc	Internet relay chat program
wget	Non-interactive network downloader
ssh	OpenSSH secure shell client (remote login)
nmap	Map subnets

- This section also assumes some familiarity with the concepts
 - Internet Protocol (IP) address
 - Domain and host name
 - Uniform Resource Identifier (URI)

Internet addresses

- An IP address is a numerical label like 192.168.1.10
- It identifies a network interface and enables connections to host computers (computers with an OS and user access)
- There are two types of (standard) Internet protocols, IPv4 (32-bit) and IPv6 (128-bit)

← Settings



Properties

SSID:	MyAltice 63d435
Protocol:	Wi-Fi 5 (802.11ac)
Security type:	WPA2-Personal
Network band:	5 GHz
Network channel:	48
Link speed (Receive/Transmit):	866/866 (Mbps)
Link-local IPv6 address:	fe80::a8e1:be74:d100:29db%3
IPv4 address:	192.168.1.116
IPv4 DNS servers:	192.168.1.1
Manufacturer:	Intel Corporation
Description:	Intel(R) Wi-Fi 6 AX201 160MHz
Driver version:	22.100.0.3
Physical address (MAC):	04-56-E5-25-D2-5D

Copy

Figure 1: WiFi information on a Windows Box

- The IPv4 and IPv6 addresses are dynamical
- The MAC address is static
- The DHCP (Dynamical Host Configuration Protocol) assigns IP addresses to devices connected to the network
- [] Check if a `dhcp` process runs on your computer. Use two methods on the terminal or on an Emacs shell:

1. process check with `ps aux` - use `grep` to search for `dhcp`.

```
ps aux | grep --exclude=grep dhcp
```

```
marcus 12 0.0 0.0 16208 1284 tty1 S 22:05 0:00 grep --exclude=grep dhcp
```

2. search for `dhcp` in the output of the `systemctl` service program (the program that talks to `systemd`). Pipe the output of `systemctl status` into `grep`.

```
systemctl status | grep dhcp
```

3. []

Find out what the active flags `-b` `-q` for the `dhcpcd(8)` program mean.

`-b` stands for "background" (startup scripts) `-q` stands for "quiet" (level of system messages)

4. On the man page, you find the information that this daemon program implements an RFC - a Request For Comment. This is the traditional (since 1969) title for standard-setting documents for the Internet, or more specifically for TCP/IP (Transmission Control Protocol/Internet Protocol)

The Internet and the Web

- The World-Wide Web is a collection of web pages on the Internet
- Similar to a shopping mall with road access
- Web locations and Internet addresses are linked but not identical
- The Internet's name space is structured by standardized strings: Universal Resource Identifiers (URI), a Universal Resource Locators (URL) and a Universal Resource Name (URN).
- A URI contains both URL and URN.
- URI syntax:

```
scheme://authority]path[?query][#fragment]
```

SYNTAX ELEMENT	EXAMPLES
Scheme	http, file, ftp, data, irc
Authority	userinfo@, host (IP), port (80)
Path	path to the resource
Query	query string (optional)
Fragment	direction to secondary resource (optional)

- URI Examples:

URI	WHAT
-----	------

URI	WHAT
mailto:birkenkrahe@lyon.edu	user mail
https://github.com/birkenkrahe/os420/.../README.org#my-first-pi	GitHub link
http://ftp.gnu.org/gnu/emacs/	GNU Emacs file server
irc.freenode.net	Internet Relay Chat

Network address on Windows

- Go to Settings > Network & Internet > Wi-Fi > Hardware properties
Here you see the IPv4 address, e.g. for my computer: 192.168.1.116
- On Windows, the `hostname` command only gives you the name of the windows box

Examining and monitoring a network

Look at the man page for each of these programs if you're interested in learning more, and try some of the many options.

Knock-knock who's there: `ping`

- The `ping` command sends a special network packet called an ICMP ECHO_REQUEST to a specified host.
- Most network devices receiving this packet will reply to it, allowing the network connection to be verified.
- You can configure the network connection to ignore these packets (for security reasons).
- A typical packet reply looks like this:

```
64 bytes from ord38s29-in-f14.1e100.net (142.250.191.142): icmp_seq=3 ttl=57 time=31.9 ms
```

It contains the packet size, the target IP, time to live and transmitting time information.

- When you interrupt the communication with `c-c` `C-c` or `CTRL-c`, you get some stats:

```
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 31.358/31.437/31.517/0.079 ms
```

- A properly performing (transparent) network will exhibit 0 % packet loss and indicate that different network elements (interface cards, cabling, routing, gateways) are OK.
- [X]

Go to the terminal or to an Emacs shell and ping Lyon College, and then Google. Do you see a difference?

```
ping lyon.edu
ping google.com
```

What's the route: `traceroute`

- This program lists all the hops network traffic takes to get from the local system to the specified host. Here is the route from my house to lyon.edu:

```
~/GitHub $ traceroute lyon.edu
traceroute to lyon.edu (40.119.1.254), 30 hops max, 60 byte packets
 1 Docsis-Gateway (192.168.1.1)  8.721 ms  8.383 ms  8.302 ms
 2 * * *
 3 173-219-255-40.suddenlink.net (173.219.255.40)  18.387 ms  18.322 ms  18.240 ms
 4 173-219-221-143.suddenlink.net (173.219.221.143)  40.121 ms  39.998 ms  39.903 ms
 5 173-219-221-138.suddenlink.net (173.219.221.138)  39.556 ms  39.471 ms  39.343 ms
 6 173-219-17-110.suddenlink.net (173.219.17.110)  38.896 ms  42.320 ms  41.960 ms
 7 173-219-152-172.suddenlink.net (173.219.152.172)  41.850 ms  41.737 ms  41.700 ms
 8 66-76-232-151-chic.tex.sta.suddenlink.net (66.76.232.151)  41.498 ms  35.773 ms  35.771 ms
 9 ae35-0.icr01.ch4.ntwk.msn.net (104.44.237.19)  45.126 ms  45.163 ms ae30-0.icr01.ch2.ntwk.msn.r
10 be-100-0.ibr01.ch2.ntwk.msn.net (104.44.11.252)  58.979 ms be-120-0.ibr02.ch2.ntwk.msn.net (104
11 be-6-0.ibr02.dsm05.ntwk.msn.net (104.44.18.217)  60.892 ms be-4-0.ibr01.dsm05.ntwk.msn.net (104
12 be-7-0.ibr02.sn1.ntwk.msn.net (104.44.16.38)  53.861 ms  53.767 ms be-9-0.ibr01.sn1.ntwk.msn.ne
13 ae100-0.icr01.sn6.ntwk.msn.net (104.44.23.78)  64.181 ms ae124-0.icr03.sn1.ntwk.msn.net (104.44
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *

~/GitHub $
```

- Traversing the entire route requires 13 routers.
- Each router stop is accompanied by 3 round-trip times.
- For routers that do not give identifying information (because of network congestion, firewalls, etc.) you see asterisks
- The `-T` and `-I` options (different probes) sometimes gives more information (and requires `sudo` rights)
- [] Check the traceroute to `lyon.edu` yourself, and contrast it again with the traceroute to `google.com`. Check if `-T` or `-I` make a difference.

What's interfaced: ip and ifconfig

- The `ip(8)` program is a multi-purpose network configuration tool
- The `ifconfig(8)` program is the older (deprecated) version of `ip`
- []

Run `ip a` in the code block below.

```
ip a
```

```
18: eth0: <> mtu 1500 group default qlen 1
    link/ether 04:56:e5:25:d2:61
    inet 169.254.0.79/16 brd 169.254.255.255 scope global dynamic
        valid_lft forever preferred_lft forever
    inet6 fe80::5809:6379:fd0e:4f/64 scope link dynamic
        valid_lft forever preferred_lft forever
14: eth1: <> mtu 1500 group default qlen 1
    link/ether c0:25:a5:5a:cd:6b
    inet 169.254.118.60/16 brd 169.254.255.255 scope global dynamic
        valid_lft forever preferred_lft forever
```

```

        inet6 fe80::7df2:c4f3:9a7e:763c/64 scope link dynamic
            valid_lft forever preferred_lft forever
10: eth2: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1
    link/ether 0a:00:27:00:00:0a
        inet 192.168.56.1/24 brd 192.168.56.255 scope global dynamic
            valid_lft forever preferred_lft forever
        inet6 fe80::69da:73c3:7432:ff6a/64 scope link dynamic
            valid_lft forever preferred_lft forever
1: lo: <LOOPBACK,UP> mtu 1500 group default qlen 1
    link/loopback 00:00:00:00:00:00
        inet 127.0.0.1/8 brd 127.255.255.255 scope global dynamic
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host dynamic
            valid_lft forever preferred_lft forever
3: wifi0: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1
    link/ieee802.11 04:56:e5:25:d2:5d
        inet 192.168.1.117/24 brd 192.168.1.255 scope global dynamic
            valid_lft 2095sec preferred_lft 2095sec
        inet6 fe80::a8e1:be74:d100:29db/64 scope link dynamic
            valid_lft forever preferred_lft forever
2: wifi1: <> mtu 1500 group default qlen 1
    link/ieee802.11 04:56:e5:25:d2:5e
        inet 169.254.118.103/16 brd 169.254.255.255 scope global dynamic
            valid_lft forever preferred_lft forever
        inet6 fe80::5fc:9e9a:14eb:7667/64 scope link dynamic
            valid_lft forever preferred_lft forever
9: wifi2: <> mtu 1500 group default qlen 1
    link/ieee802.11 06:56:e5:25:d2:5d
        inet 169.254.5.2/16 brd 169.254.255.255 scope global dynamic
            valid_lft forever preferred_lft forever
        inet6 fe80::a58b:f6a0:f53d:502/64 scope link dynamic
            valid_lft forever preferred_lft forever

```

- My system has three network interfaces:
- `lo` for loopback, used by the OS to talk to itself
- `eth` for Ethernet interface
- `wlan0` for the WiFi interface
- When performing diagnostics, look for the word `UP` in the first line (which means it's enabled), and a valid IP address in the `inet` field. E.g. I only have WiFi right now, and no Ethernet.
- []

Run `ifconfig -a` in the code block below. This command is a little easier to understand, I think.

```
ifconfig -a
```

What's on the net: `netstat`

- This program is used to examine network settings and statistics.
- []

Run `netstat -ie` to examine network interfaces.

```
netstat -ie
```

- The output of `netstat -ie` looks similar to the `ifconfig` command because it focuses on network interfaces. Only `lo` and `wlan0` transport any packets.
- [X]

Run `netstat -r` to see the routing table, which shows how the network is configured to send packets from network to network:

```
netstat -r
```

- This is a typical table for a client on a local area network (LAN) behind a firewall/router. The first line shows the destination IP, 192.168.1.0. The last 0 means that the address refers to multiple hosts.
- The Gateway is the name or router to go from the current host to the destination network.
- The I[nter]face to connect is WiFi (wlan0).
- []

Run the `hostname -I` command to see your own host on your LAN:

```
hostname -I
```

```
192.168.56.1 192.168.1.117
```

Mine is 192.168.1.160. One of about 20 network devices in my house, including: PS5, SmartTV, Kindle tablets, PCs, Mac, Raspberry Pi (this is it), network printers etc.

Transporting files over a network with `ftp` and `wget`

FTP

- `ftp` (File Transfer Protocol) is a "classic" program. It is supported by all web browsers.
- []

Check if the `ftp` daemon `ftpd` is awake.

```
systemctl status | grep ftpd
```

- []

Check if the program `ftp` is even available/installed.

```
which ftp
```

- FTP in its original form is **not safe** because it sends account names and passwords in clear text, i.e. unencrypted. Anyone sniffing the network can see them.
- Therefore, all FTP traffic on the Internet is done by *anonymous* FTP servers that allow you to login using the username *anonymous* and a meaningless password.
- Try `lftp` - a better FTP client. On Windows, I used Cyberduck years ago. Dedicated clients offer a lot more features.

An FTP session

```
pi@raspberrypi:~/GitHub/os420$ ftp ftp.gnu.org
Connected to ftp.gnu.org.
220 GNU FTP server ready.
Name (ftp.gnu.org:pi): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> status
Connected to ftp.gnu.org.
No proxy connection.
Connecting using address family: any.
Mode: stream; Type: binary; Form: non-print; Structure: file
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Quote control characters: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
Tick counter printing: off
ftp> 230 Login successful.
?Invalid command
ftp>
```

Figure 2: FTP session example

- [] Install `ftp` as `sudo` - use an Emacs shell or a terminal for that.
- [] In a terminal or on an Emacs shell, run `ftp`
- [] On the `ftp>` shell, type `help` to see the available commands
- [] Open a connection with `open`
- [] As target IP address, enter `ftp.gnu.org`
- [] Login as `anonymous`
- [] List the current directory with `ls`
- [] Get the `README` file with `get README`
- [] Rename `README` on your computer to `README1`
- [] Send `README1` to the other location with `send README1`
- [] Close the connection with `close` and quit with `quit`

wget

- `wget` is a tool for file downloading both from web and FTP sites. It exhibits network resilience, e.g. it will keep trying to get the job done even if the network is slow or unstable. It does the job in the background
- `wget` uses "recursive downloading" and recreates the entire file structure that it finds remotely at the local site while respecting the local "Robot Exclusion Standard" of the `robots.txt` file¹.
- You can download files, directories, and entire sites.
- []

Download the Lyon College landing page with `wget` using `lyon.edu` as the only command. The program will substitute any other information necessary.

- Check the current directory for the result (`index.html`).

```
wget lyon.edu
```

- Go to a terminal (or the Emacs shell) and run the command there again to see the full screen message:

```
pi@raspberrypi:~/GitHub/os420$ wget lyon.edu
--2022-04-21 08:25:02-- http://lyon.edu/
Resolving lyon.edu (lyon.edu)... 40.119.1.254
Connecting to lyon.edu (lyon.edu)|40.119.1.254|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.lyon.edu/ [following]
--2022-04-21 08:25:02-- http://www.lyon.edu/
Resolving www.lyon.edu (www.lyon.edu)... 40.119.1.254
Reusing existing connection to lyon.edu:80.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.lyon.edu/ [following]
--2022-04-21 08:25:02-- https://www.lyon.edu/
Connecting to www.lyon.edu (www.lyon.edu)|40.119.1.254|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 146047 (143K) [text/html]
Saving to: 'index.html'

index.html      100%[=====] 142.62K --.- KB/s   in 0.1s

2022-04-21 08:25:03 (1.08 MB/s) - 'index.html' saved [146047/146047]

pi@raspberrypi:~/GitHub/os420$ 
```

Figure 3: wget terminal screen message

- The man page is better-than-average and highly readable.
- In Emacs, you can also go to the `Dired` buffer with `c-x d` and type `! chromium browser RET` on the file `index.html`. This will open the page locally in a browser.

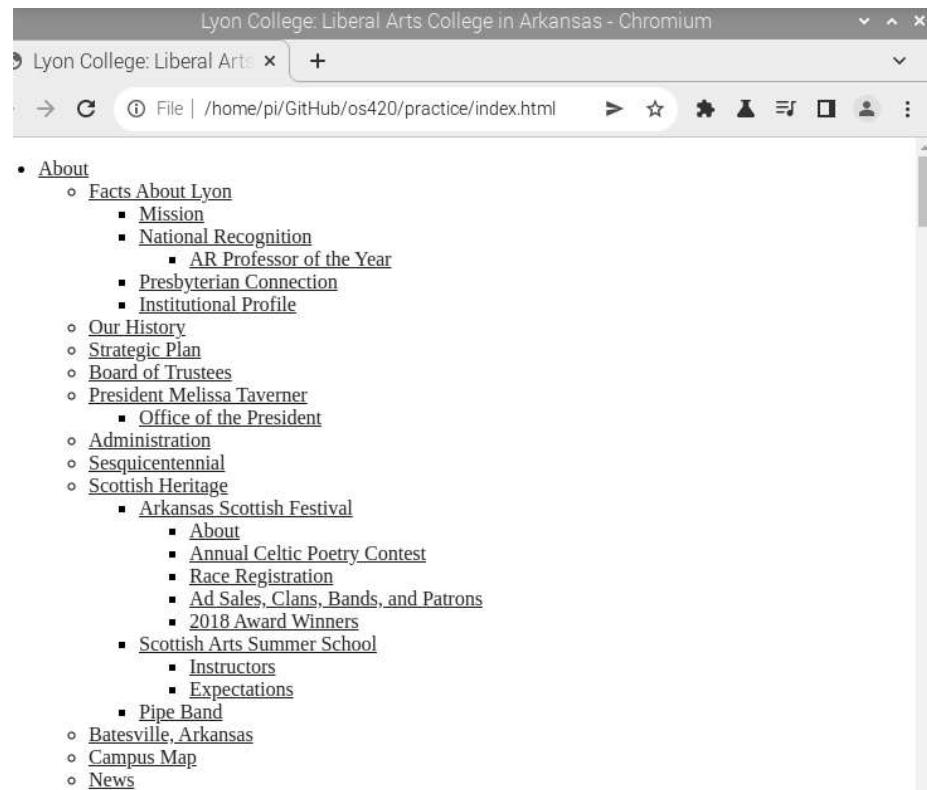


Figure 4: local copy of lyon.edu/index.html in Chromium browser

Secure communication with ssh

Overview

- Popular remote access programs included `rlogin` and `telnet`
- Like `FTP` these transmit all their communications in cleartext
- `ssh` (Secure SHell) on the other hand:
 - **authenticates** that the remote host is who it says it is (preventing so-called "man-in-the-middle" attacks)
 - **encrypts** all of the communication between the local and remote hosts
- SSH consists of two parts:
 - An **SSH server** that runs on the *remote host*, listening for incoming connections by default on port 22
 - An **SSH client** that runs on the *local system* to communicate with the remote server.
- To enable a system to receive remote connections, it must have the OpenSSH-server package installed, configured, and running, and (if the system is behind a firewall) it must allow incoming connections on TCP port 22.
- `ssh` creates a sort of virtual private network between client and host computer.

Enable ssh on your Pi

- Update your Pi now: run these commands in a shell

```
sudo apt update -y
sudo apt upgrade -y
```

Call me if either of them are unsuccessful.

- On Raspberry Pi, you may need to enable `ssh` using the `sudo raspi-config` terminal command. This opens a screen dialog.
 - Go to **Interfacing options** and **Enable ssh**, then **Save and Finish**.



Figure 5: Raspi Config screen dialog (ssh)

- []

Once enabled, check if `ssh` is active by searching for its daemon, `sshd`.

```
ps aux | grep sshd
```

```
marcus      45  0.0  0.0  16208  1276 tty1      S    22:05   0:00 grep sshd
```

When logged in on my Pi from my Windows box, I get this result, which shows that I (as pi) have one active secure shell connections. It is controlled by root and asleep:

```
: root      527  0.0  0.1  12356  5768 ?          Ss  20:32   0:00 sshd: /usr/sbin/sshd -D [listene
: root     1972  2.5  0.1  14452  6936 ?          Ss  20:57   0:00 sshd: pi [priv]
: pi       1978  0.0  0.1  14452  4404 ?          S    20:57   0:00 sshd: pi@pts/0
: pi       1993  0.0  0.0   7452   572 ?          S    20:57   0:00 grep sshd
```

When I used rclone to connect to my GDrive from within Emacs, I get these buffers in my buffer list:

```
% /rclone:pi@gdrive:/      1971 Dired by name      /rclone:pi@gdrive:/
* *tramp/rclone pi@gdrive*      0 Fundamental
```

Map your network neighborhood with nmap

- []

Find your own and the other party's hostname with `hostname -I`: this is the only information you need, apart from the username.

```
hostname -I
```

```
192.168.56.1 192.168.1.117
```

- [] Install the nmap tool to map the whole subnet (all computers in your network vicinity): run `sudo apt install nmap` in a shell.
- []

Run `nmap -sn` on your whole subnet range. You need to take the network address that you obtained earlier with `hostname` and replace the last number by `0/24`.

```
nmap -sn 192.168.1.0/24
```

This is what I get at home at the moment:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-25 21:10 CDT
Nmap scan report for Docsis-Gateway (192.168.1.1)
Host is up (0.011s latency).
Nmap scan report for BRW5C6199417261 (192.168.1.19)
Host is up (0.030s latency).
Nmap scan report for Carlyes-MacMini (192.168.1.30)
Host is up (0.080s latency).
Nmap scan report for RE230 (192.168.1.97)
Host is up (0.0087s latency).
Nmap scan report for Galaxy-A32-5G (192.168.1.98)
Host is up (0.039s latency).
Nmap scan report for raspberrypi (192.168.1.161)
Host is up (0.00011s latency).
Nmap scan report for RE230 (192.168.1.179)
```

```
Host is up (0.023s latency).
Nmap scan report for 192.168.1.207
Host is up (0.020s latency).
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.44 seconds
```

If you try this at home, you better make sure that you know who all the participants of your subnet are. You can also ping them and if they are other computers with ssh, connect to them remotely.

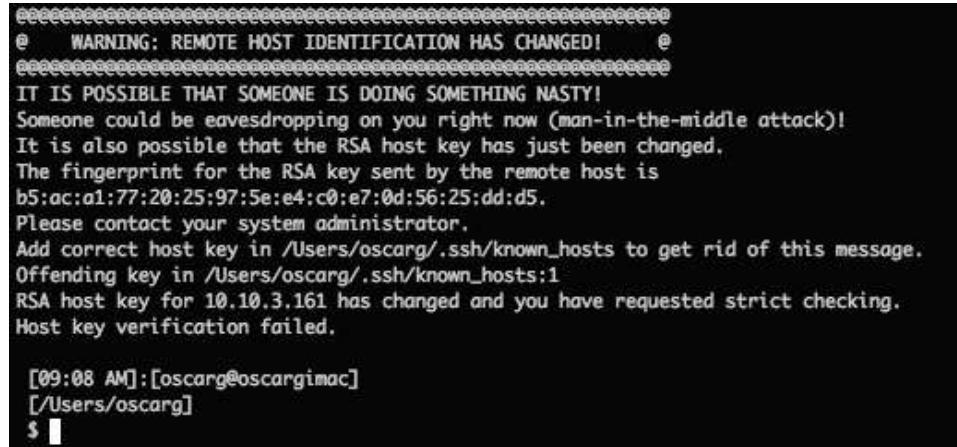
Connect with your neighboring Pi using SSH.

- []

Make sure ssh is alive and running: type `systemctl status ssh` in a terminal or Emacs shell or run the block below and check the file.

```
systemctl status ssh > ssh.status
cat ssh.status
```

- [] In a terminal: enter `ssh pi@hostname` then enter the password, which is `ly0Np1_Numb3r_xx` where `xx` is the number of your Pi. `hostname` is the IP address you just obtained.
- The first time the connection is attempted, a message is displayed that the authenticity of the remote host cannot be established (because the client - your computer - has never seen this remote host before). You need to explicitly OK this (yes)
- If the connection is refused by the host computer, then ssh is not installed or enabled, or not running.
- If the remote host answers but does not successfully authenticate, you get a message like this:



```
=====  
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @  
=====  
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!  
Someone could be eavesdropping on you right now (man-in-the-middle attack)!  
It is also possible that the RSA host key has just been changed.  
The fingerprint for the RSA key sent by the remote host is  
b5:ac:a1:77:20:25:97:5e:e4:c0:e7:0d:56:25:dd:d5.  
Please contact your system administrator.  
Add correct host key in /Users/oscarg/.ssh/known_hosts to get rid of this message.  
Offending key in /Users/oscarg/.ssh/known_hosts:1  
RSA host key for 10.10.3.161 has changed and you have requested strict checking.  
Host key verification failed.  
  
[09:08 AM]:[oscarg@oscargimac]  
[~/Users/oscarg]  
$ █
```

Figure 6: Remote connection not authenticated

- This message can be the result of two situations:
 1. an attacker may be attempting a man-in-the-middle attack. This is unlikely because it is known that ssh signals an alert.
 2. the remote system has been altered since the last connection - the OS or the ssh server may have been re-installed.
- To clean up the **client side**, you need to remove obsolete keys from the `~/.ssh/known_hosts` file:

```
/home/pi/.ssh:
total used in directory 12 available 19 GiB
drwx----- 2 pi pi 4096 Feb 11 07:10 .
drwxr-xr-x 24 pi pi 4096 Apr 25 20:42 ..
-rw-r--r-- 1 pi pi 444 Mar 17 06:45 known_hosts
```

Figure 7: Known hosts file in \$HOME/.ssh/

- [] Take a look at your `known_hosts` file with `cat`.
- [] If your connection is successful, check that you're on the other machine by checking the SSH `systemctl status` and/or the `hostname`. You can even open Emacs here with `emacs -nw` (non-graphical Emacs).
- []

In Emacs: open a `Dired` buffer with `C-x d` and at the prompt, enter

```
/ssh:pi@hostname:~/
```

You should now see the other computer's `/home/pi` directory. Open a shell with `M-x shell` and you'll see that it will open on the other computer.

Get a directory list from the remote system

- []

This is unexciting if you're in Emacs with `Dired` but if you do it on the shell, it has hacker qualities: get a remote directory listing on your computer with

```
ssh [remote-system] 'ls -l' > dirlist
```

Or create a `dirlist` on the host computer with

```
ssh [remote-system] 'ls -l > dirlist'
```

scp and sftp

- The OpenSSH package contains two more service programs that can make use of the encrypted `ssh` tunnel: `scp` for remote file copy, and `sftp` for remote file transfer (without a remote FTP server).
- [] On both client and host, create an empty file with your own `hostname` as the title, e.g. `192.168.1.161`.
- []

Copy the file to each other's computer using the syntax:

```
scp pi@192.168.1.161:/home/pi/192.168.1.161 ./hostfile
```

This will copy `/home/pi/192.168.1.161` from the host into `hostfile` on your client computer.

- []

Run a remote FTP session with:

```
sftp pi@192.168.1.161
```

This will open an `sftp>` shell on which you can execute the command `ls` or `help`. To get out of the session, type `bye`.

- Many of the graphical file managers found in Linux distributions support the `sftp` protocol. You can enter a URI beginning with `sftp://` in the location bar and operate on files stored on a remote system running an SSH server.

Emacs special

- Tramp ("Transparent Remote (file) Access, Multiple Protocol" is a built-in GNU Emacs package that provides remote file editing.
- Tramp works directly with Dired using a command like:

```
C-x d /ssh:pi@192.168.1.160:~/
```

to connect to a user `pi` on a local network machine. You can open a shell on the other machine, too, and work remotely at ease.

- The `rclone` program e.g. uses Tramp as an external method to connect to network servers like GDrive, or to facilitate cloud backup.

Challenge: How to connect your Pi to GDrive

Sources: rclone.org/drive.

1. Install `rclone` on Raspbian

```
$ sudo apt-get install rclone
```

2. Change file permissions of `$HOME/.config/rclone` to `rwx` for owner only

```
$ chmod 0700 ./config/rclone # change permissions
$ ls -la ./config/rclone    # check - you should see drwx-----
```

3. Follow the detailed instructions in rclone.org/drive until you see the `Success!` web page after connecting `rclone` to GDrive

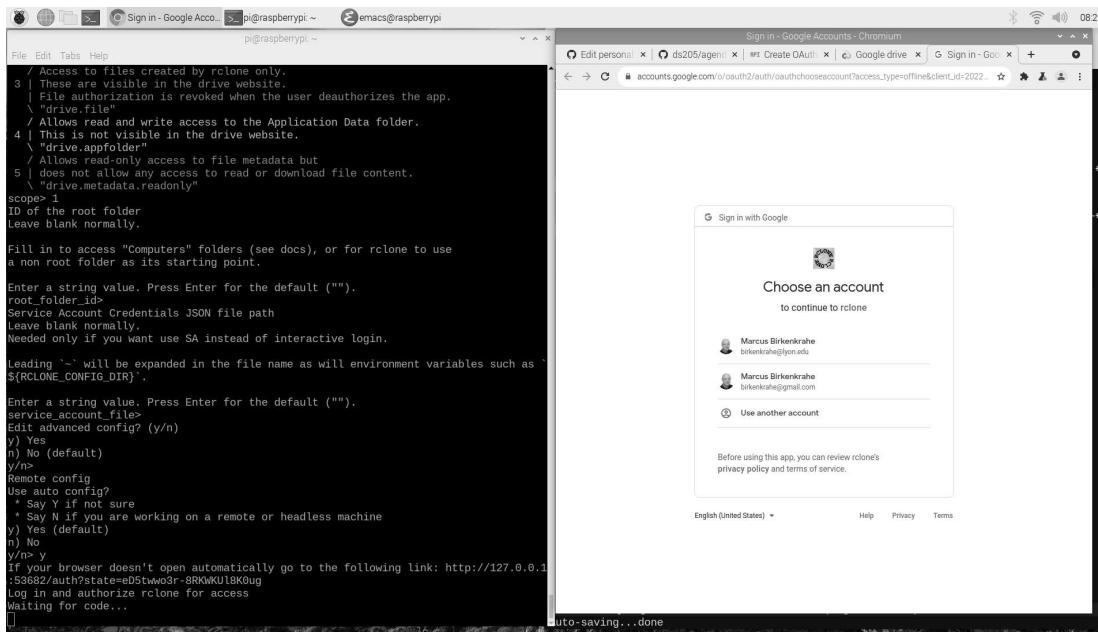


Figure 8: Selecting GDrive account for rclone

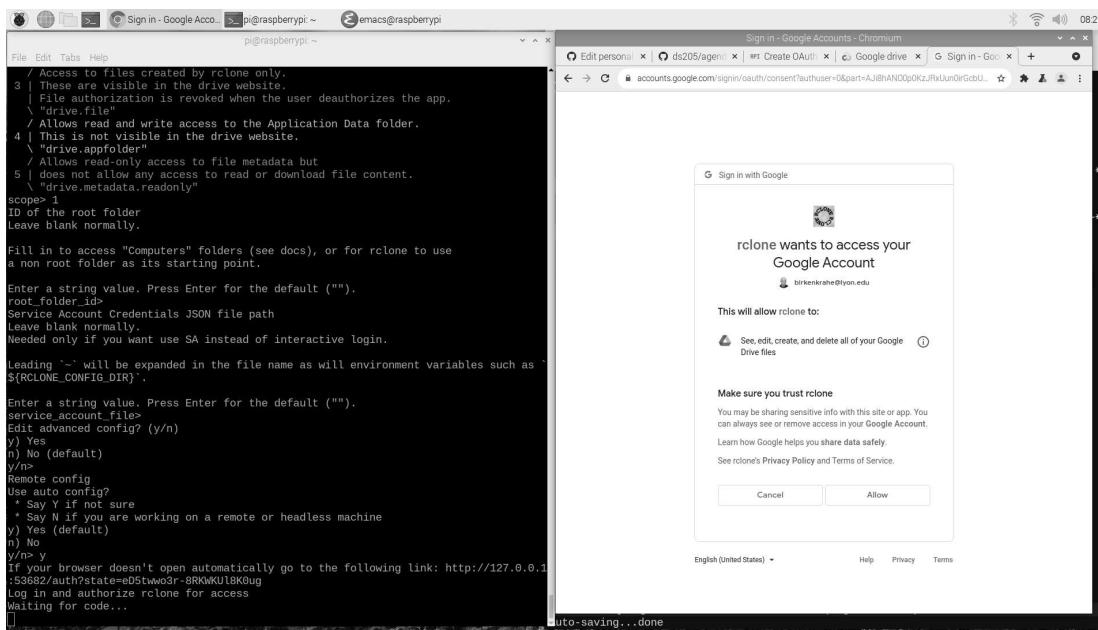
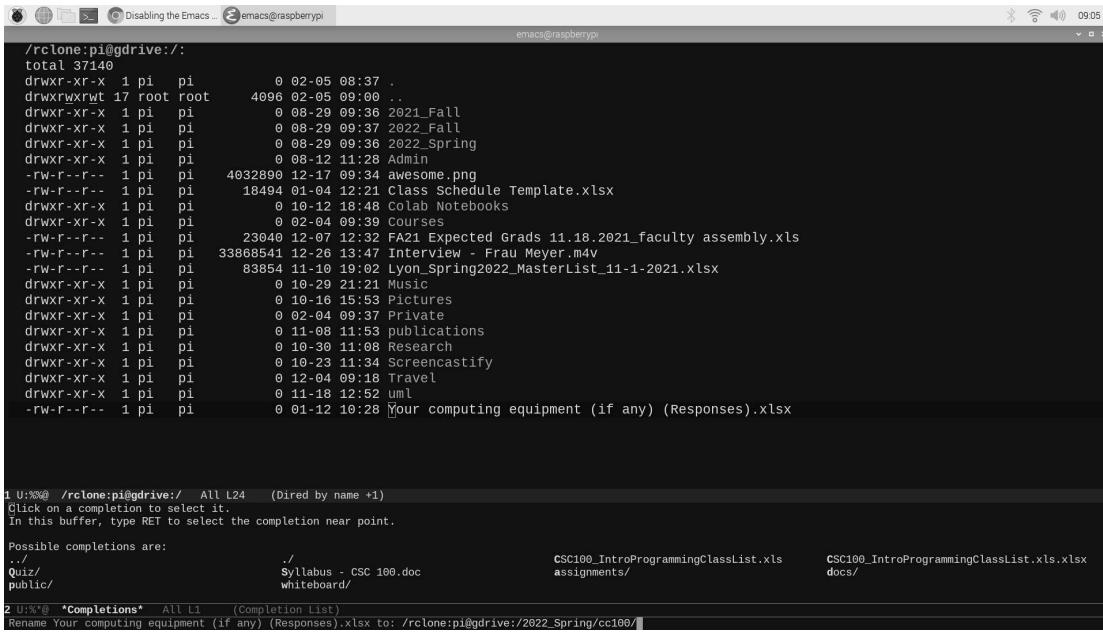


Figure 9: Give rclone permission to access GDrive

"Note that rclone runs a webserver on your local machine to collect the token as returned from Google if you use auto config mode. This only runs from the moment it opens your browser to the moment you get back the verification code. This is on <http://127.0.0.1:53682/> and this it may require you to unblock it temporarily if you are running a host firewall, or use manual mode."

The access information is stored in `$HOME/.config/rclone/rclone.conf`.

You can now access your GDrive from the command line with: `rclone [cmd] gdrive:`, or inside GNU Emacs as an [external method](#):`~C-x d /rclone:gdrive:/`.



The screenshot shows an Emacs window titled "emacs@raspberrypi" displaying a file listing from a directory named "rclone:pi@gdrive:/". The listing includes files like "awesome.png", "Class Schedule Template.xlsx", "Colab Notebooks", "Courses", "Expected Grads 11.18.2021_faculty assembly.xls", "Interview - Frau Meyer.m4v", "Lyon_Spring2022_MasterList_11-1-2021.xlsx", "Music", "Pictures", "Private", "publications", "Research", "Screencastify", "Travel", and "uml". Below the listing, a completion menu is open at the bottom of the screen, showing options such as "Quiz/", "public/", "Syllabus - CSC 100.doc", "whiteboard/", "CSC100_IntroProgrammingClassList.xls", "assignments/", and "docs/". The status bar at the bottom indicates "U:\%@ *Completions* All L1 (Completion List)".

```
/rclone:pi@gdrive:/:
total 37140
drwxr-xr-x 1 pi pi 0 02-05 08:37 .
drwxrwxrwt 17 root root 4096 02-05 09:00 ..
drwxr-xr-x 1 pi pi 0 08-29 09:36 2021_Fall
drwxr-xr-x 1 pi pi 0 08-29 09:37 2022_Fall
drwxr-xr-x 1 pi pi 0 08-29 09:38 2022_Spring
drwxr-xr-x 1 pi pi 0 08-12 11:28 Admin
-rw-r--r-- 1 pi pi 4032890 12-17 09:34 awesome.png
-rw-r--r-- 1 pi pi 18494 01-04 12:21 Class Schedule Template.xlsx
drwxr-xr-x 1 pi pi 0 10-12 18:48 Colab Notebooks
drwxr-xr-x 1 pi pi 0 02-04 09:39 Courses
-rw-r--r-- 1 pi pi 23040 12-07 12:32 FA21 Expected Grads 11.18.2021_faculty assembly.xls
-rw-r--r-- 1 pi pi 33868541 12-26 13:47 Interview - Frau Meyer.m4v
-rw-r--r-- 1 pi pi 83854 11-10 19:02 Lyon_Spring2022_MasterList_11-1-2021.xlsx
drwxr-xr-x 1 pi pi 0 10-29 21:21 Music
drwxr-xr-x 1 pi pi 0 10-16 15:53 Pictures
drwxr-xr-x 1 pi pi 0 02-04 09:37 Private
drwxr-xr-x 1 pi pi 0 11-08 11:53 publications
drwxr-xr-x 1 pi pi 0 10-30 11:08 Research
drwxr-xr-x 1 pi pi 0 10-23 11:34 Screencastify
drwxr-xr-x 1 pi pi 0 12-04 09:18 Travel
drwxr-xr-x 1 pi pi 0 11-18 12:52 uml
-rw-r--r-- 1 pi pi 0 01-12 10:28 Your computing equipment (if any) (Responses).xlsx
```

1 U:\%@ /rclone:pi@gdrive:/ All L24 (Dired by name +1)
Click on a completion to select it.
In this buffer, type RET to select the completion near point.
Possible completions are:
./ Quiz/ public/ CSC100_IntroProgrammingClassList.xls assignments/ CSC100_IntroProgrammingClassList.xls.xls
docs/

2 U:\%@ *Completions* All L1 (Completion List)
Rename Your computing equipment (if any) (Responses).xlsx to: /rclone:pi@gdrive:/2022_Spring/cc100/

Figure 10: GDrive via rclone in Emacs

TODO Summary

Footnotes:

¹ This file specifies rules for web crawlers. If you have your own web server somewhere with an Internet connection, you're likely to have such a file. You can e.g. use it to block sites. It has a simple syntax. [Here is a simple guide](#).

Author: Marcus Birkenkrahe

Created: 2022-04-25 Mon 22:05