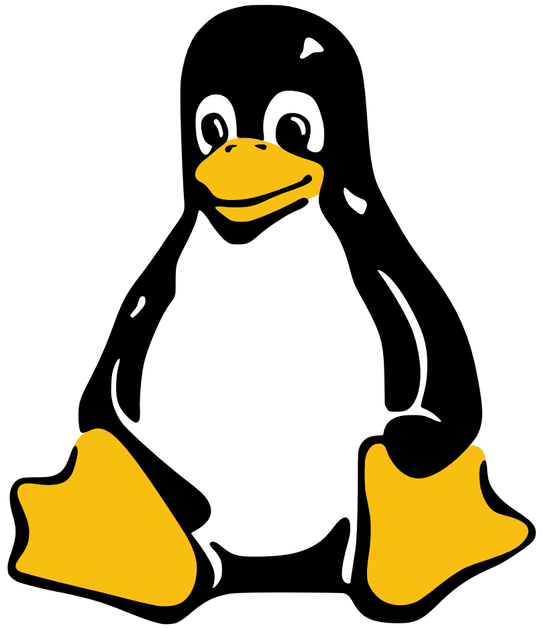
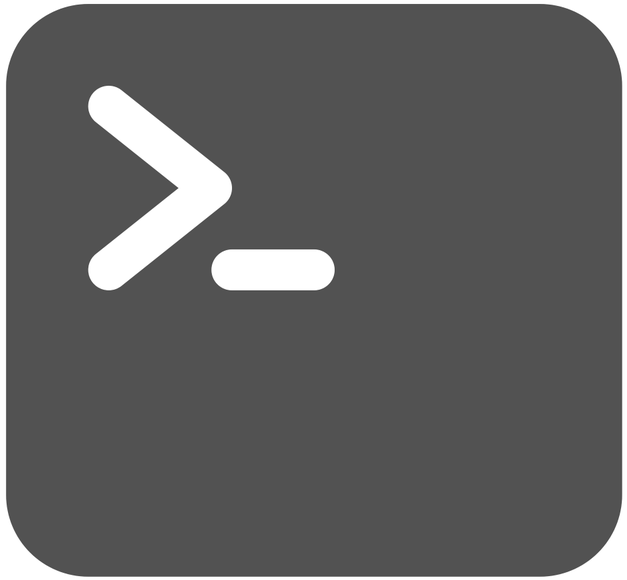




Remote computing

Outline

- command-line (CLI) basics
- ssh
- tmux
- copying files between computers
- examples
  - MATLAB, X11 forwarding
  - outputting to a file
  - Jupyter notebooks
  - neural net, GPU

CLI basics

Navigation

- change directory:

```
cd
```

- list directory contents:

```
ls
```

- make directory:

```
mkdir
```

File operations

- copy files:

```
cp
```

- move files:

```
mv
```

- delete files:

```
rm
```

Misc

- redirect stdout to a file:

```
example-command > out.txt
```

- wildcards:

```
rm *.txt
```

- run in background:

```
some-command &
```

- tab completion: press tab to autocomplete commands, file names, etc.

Connecting to the remote computer Using SSH (secure shell protocol)

usage: `ssh user@hostname`

example:

```
ssh trevor@bz-ece-whitgpu1
```

Note: when connected via the MSU VPN, add `.msu.montana.edu` to the hostname:

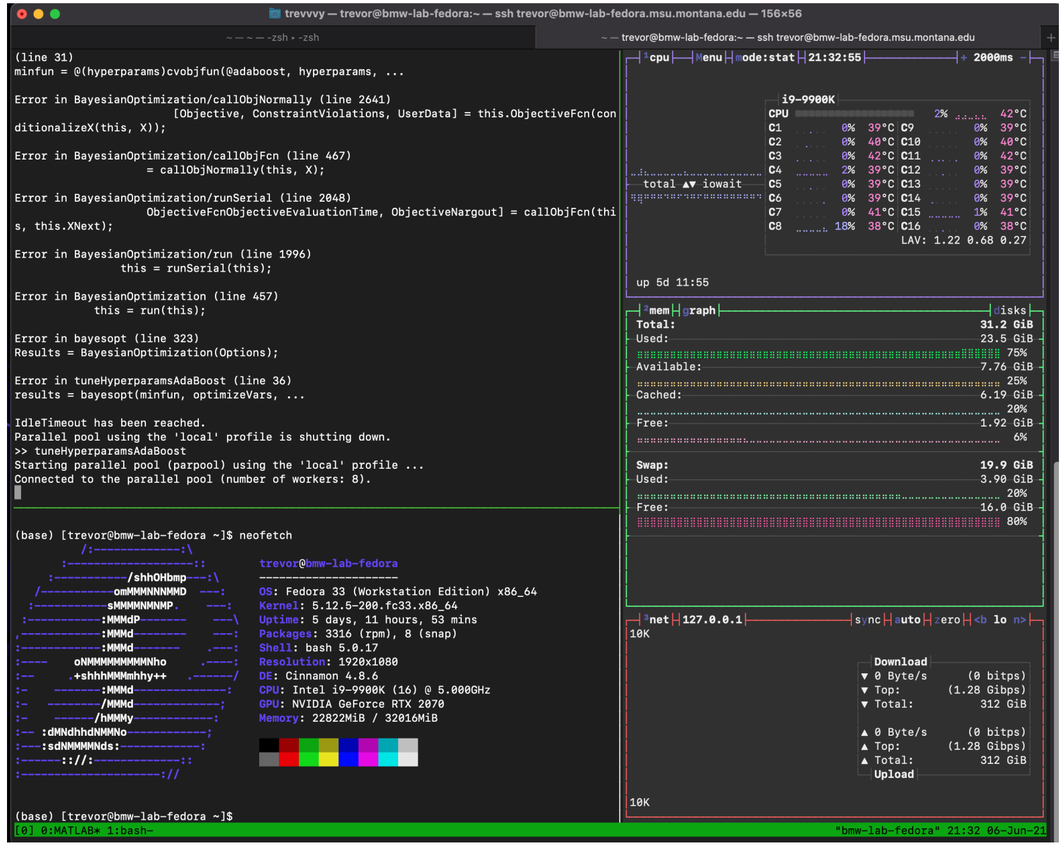
```
ssh bmw@bz-ece-whitlab1.msu.montana.edu
```

Ensuring jobs don’t accidentally die

- By default, exiting the ssh session kills your jobs
- Sometimes ssh likes to exit on its own 😒

Use *tmux* so we can detach our processes from the controlling terminal

- Can have multiple panes (terminal multiplexing)
- Can have multiple windows
- Allows us to run multiple jobs easily



tmux basics

- **create a new session**

```
tmux
```

- **attach to last session**

```
tmux attach
```

- **detach from a session**

```
ctrl+b d
```

- create a new window

```
ctrl+b  c
```

- change window

```
ctrl+b  <window-number>
```

- split pane horizontally

```
ctrl+b "
```

- split pane vertically

```
ctrl+b %
```

- switch between panes

```
ctrl+b <arrow-key>
```

GUIDES / CHEATSHEETS

- https://www.ocf.berkeley.edu/~ckuehl/tmux/
- https://tmuxcheatsheet.com/

Copying files using SCP (secure copy)

From local to remote:

```
scp file user@hostname:<location>

scp test.py trevor@bz-ece-whitgpu1:~/project/
```

From remote to local:

```
scp user@hostname:<file-location> <local-file-location>

scp trevor@bz-ece-whitgpu1:~/project/test.py some-dir/test2.py
```

NEED TO SYNC LOTS OF FILES?

- Globus, OneDrive, Box, etc. ☁️
- use git for code (our lab has a GitHub Organization) 🤓 

Examples

Showing plot windows with X11 forwarding

What if we need plots?

```
x = 0:0.001:2*pi;
y = sum(rand(4,1) .* sin(randi(5, 4, 1) * pi .* x + rand(4,1)));
plot(x,y)
```

- use X11 forwarding if you only need to display a few graphics
- if you need lots of graphics, use VNC, RDP, etc.

X11 FORWARDING

```
ssh -Y user@hostname
```

- On Windows, use MobaXterm

Note: X11 forwarding might be obsolete in the future if Wayland becomes the de-facto display server

Multiple jobs, redirecting output to a file

REDIRECT OUTPUT

```
function silly(n)
disp("n = " + num2str(n))
pause(1)
disp("n^2 = " + num2str(n^2))
end
```

- run from cli and redirect output to a file

```
matlab -batch "silly 2" > 2.txt
```

- if you want to see the output while the program is running

```
matlab -batch "silly 2" | tee 2.txt
```

RUNNING MULTIPLE JOBS

1. run in the background using a trailing &
2. use multiple tmux windows/panes

Jupyter notebooks

- use port forwarding, a.k.a tunneling

```
ssh -L remote-port:host:host-port user@hostname

ssh -L 8888:localhost:8888 trevor@bmw-lab-fedora
```

- Jupyter notebook’s default port is 8888
- launch notebook as so:

```
jupyter notebook --no-browser
```

Neural net with GPU

- take it away, Brad!