Motivation

- A lot of love online for how *editor XYZ* supports multi-user mode
- But ... we have been able to do that with standard Unix tools for two+ decades
- As in Dirk’s *t4 series* it is about *combining* existing (shell) tools for better use
- Here: *tmux*, or its alternate frontend *byobu*, can shine
EMACS: MULTI-USER CONCURRENT EDITING

Setup

• Requirement: We assume multiple accounts on the same machine
  • *e.g.* a departmental server, research group server, AWS/GCS/Azure/DO instance …

• Each participant needs to
  • be able to access the box via `ssh`
  • have access to a shared (*i.e.* “joint”) “named” tmux socket
  • (and we do not cover account creation, `root` use, …)

• We then merely invoke `tmux` (or `byobu`), once creating a named socket
  • *Everything* running inside the tmux/byobu session can then be shared
  • Including Emacs (in terminal mode) or anything else working in terminal mode
EMACS: MULTI-USER CONCURRENT EDITING

Simplest Demo: Launch It

This runs starts the session, i.e. the one granting access to his session.

```
# start tmux: named socket (-S), group (-t) and session (-s) names
$ tmux -S /tmp/joint-session new-session -t joint -s joint
# we are now in a tmux session
> chmod o+rw /tmp/joint-session   # so that other user can access
```

We created a new session under the handle ‘joint’ and gave it rw (== “read-write”) rights. (In real settings you probably want both users under a common ‘group’ for the joint project and make the socket group rw.)
Simplest Demo: Joint It
This runs as one of the users accessing the session. The only requirements are to be one the same machine, and to be part of the group having access to the shared socket.

```
$ tmux -S /tmp/joint-session attach-session -t joint  ## or also
$ byobu -S /tmp/joint-session attach-session -t joint
```

That. Is. All. The two sessions mirror each other.  

So do `emacs -nw` or `R` or ... whatever command session you want to share. 

Best of all: All the `byobu` (or `tmux`) goodness we talked about in t4 works the same way. Multiple concurrent sessions. Persistent sessions i.e. start this, leave it running, re-connect tomorrow. Or next month.
Other Tricks

- Start `emacs --daemon` to have emacs sessions persist
- Log on and off, attach and detach `byobu` / `tmux` and resume
- Emacs sessions remain accessible
- Plotting on tty is an issue but e.g. `httpgd` may help
  - but ideally we would be able to have it send traffic to mulitple devices / users
Lastly ...
  • This is not meant to be normative
  • Other editors can do similar things just fine (code is popular)
  • (Commercial) online solutions also available (replit.io)

But ... we really like Emacs as it
  • is ‘always there’
  • has support for just about anything
  • is extensible
Security

- We do stress that letting other users access
  - your session via the shared socket
  - gives them your session with access to all you can access
  - this may include mail, gpg or ssh keys and more
- That means any files accessible to you are now also accessible to them
- So use on personal servers containing personal data may not be the first choice
- However, on work or cloud settings with no personal files less of an issue
Summary

- An easy and powerful and generic way to share a session
- Works for Emacs, also works for other (console) editors, R, ...
- While keeping possible limitation in mind, it can facilitate collaboration