



MULTI-USER CONCURRENT EDITING WITH EMACS

PART OF THE ESS-INTRO (AND T4) SERIES

Dirk Eddelbuettel, Vincent Arel-Bundock, Grant McDermott

This version is dated 28 May 2021

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 maybe two+ decades
- As in Dirk's *t4 series* it is about *combining* existing (shell) tools for better use
- Here: **tmux**, or, if one wishes to, its **byobu** frontend

Setup

- Requirement is multiple accounts on the same machine
- Back in the day this may have been the departmental or group unix server
- These days it may be a cheap cloud instance, and an AWS/GCS/Azure/DO/... box
- Each participant
 - needs to be on the box and
 - have access to a shared (*i.e.* “joint”) “named” tmux socket
 - we on purpose to not cover account creation, **root** use, ...
- We then “merely” invoke **tmux** (or **byobu**) with the 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

Simplest Demo

This runs as the user starting the session, *i.e.* the one granting access to his session.

```
$ mkdir /tmp/scratch          # new dir, not strictly needed
$ tmux -S /tmp/scratch/session new-session -t joint -s joint
# we are now in a tmux session
> chmod o+rw /tmp/scratch/session # so that other user can access
```

We created a *new session* under the identifiable handle 'joint' and gave rw 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

This runs as one of the users accessing the session. The only requirements are to be on the same machine, *and* to be part of the group having access to the shared socket.

```
$ tmux -S /tmp/scratch/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.

Alternate Demo (as user edd)

- If you prefer `byobu` (as I do) and have it use `tmux` (instead of `screen`)
- Just do `byobu -S /tmp/scratch/session new-session -t joint`
- Followed by `chmod o+rw` or `chgrp` or ...

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. `httplibd` may help
 - but ideally we would be able to have it send traffic to multiple 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

APPENDIX

OPEN QUESTIONS

- Detach (normal) versus exit (once ?)
- Security: When user 2 joins he/she is in your session. Keep that in mind. Sharing maybe best done in 'empty-ish' systems