

remote* pair programming

How to not use teamviewer

(Or, how to use ngrok, screen/tmux, vim/emacs/nano to streamline pairing with your remote colleagues)

(Lighting talk edition)

* some practices also apply for local pairing

Agenda

- Introduction - what is remote pair programming?
- Screen sharing - for and against
- Command line pairing - for and against
- CLI Pairing - tools
 - Ngrok
 - Tmux
 - VIM
- Demo
- Resources
- Conclusion
- Questions



Remote

PAIR PROGRAMMING

U MISSED A SEMICOLON, BRAH

Introduction to Pair Programming

What are the:

- Benefits
- Challenges
- Best practices

Introduction to Pair Programming

Benefits

- ★ Same benefits as in-person pair programming:
 - Higher quality code
 - Fewer defects
 - Focus
 - Intensity
 - Faster problem solving
 - Knowledge transfer
 - Code review
- + Lessen silo effect (developer isolation)
- + It's easy

Introduction

Challenges

- ★ Same challenges as in-person pair programming
 - good communication needed / shyness
 - (seemingly) takes more time
 - mentally tiring
 - personality / some people are jerks
- + no physical whiteboards / notebook
- + different time-zones
- + meetings are harder to organise
- + dependant on technology

Introduction

Best practices

- ★ Pairing etiquette
- ★ Generous support / tooling
- ★ Have fun with it

<https://www.youtube.com/watch?v=f9ALSPIPQhg>

Not sure if I love pair programming



**or
I'm just lonely**

Screen sharing vs Command line pairing

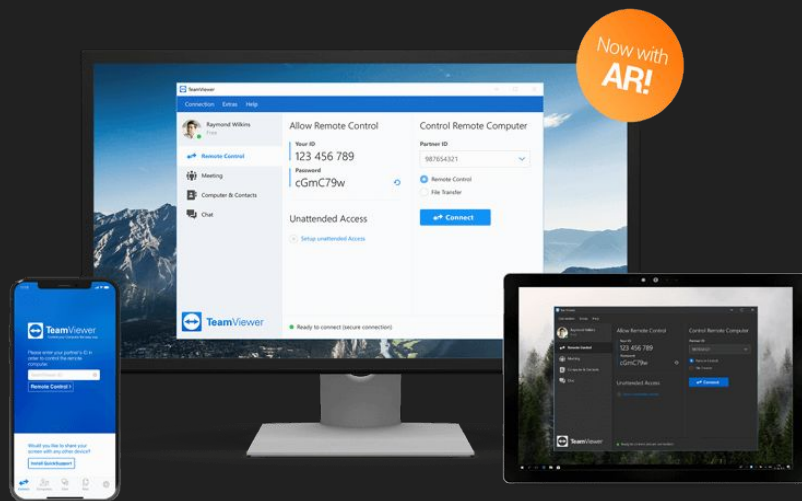
Screen sharing - for and against

Pros

- + Ability to share control to GUI applications
- + Cross platform, binary installer
- + Really good for debugging visual stuff

Cons

- Hard to use w/ unstable network
- Requires one to be fairly familiar with other person's OS / IDE



Command line pairing - for and against

Pros

- + Easier to work under slow network
- + Common CLI tools across all OSes
- + Pair on a remote server
- + Less distractions than screen sharing

Cons

- Need to know at least one CLI editor, and some shell commands
- Nearly impossible to do well in a “native microsoft windows shell”



The process - step by step

A typical remote pairing connection flow

1. ngrok & ssh
2. tmux / screen
3. vim / emacs / nano / etc
4. ...
5. profit

Ngrok

- [Ngrok.com](https://ngrok.com) (sign up for advanced features)
 - Basically does advanced ssh proxy, see <https://dev.to/k4ml/poor-man-ngrok-with-tcp-proxy-and-ssh-reverse-tunnel-1fm>

```
user1$ ssh-keygen -y # send this to user2@mac
```

```
user2$ user1_pub_key >> ~/.ssh/authorized_keys
```

```
user2$ ngrok tcp --region eu 22
```

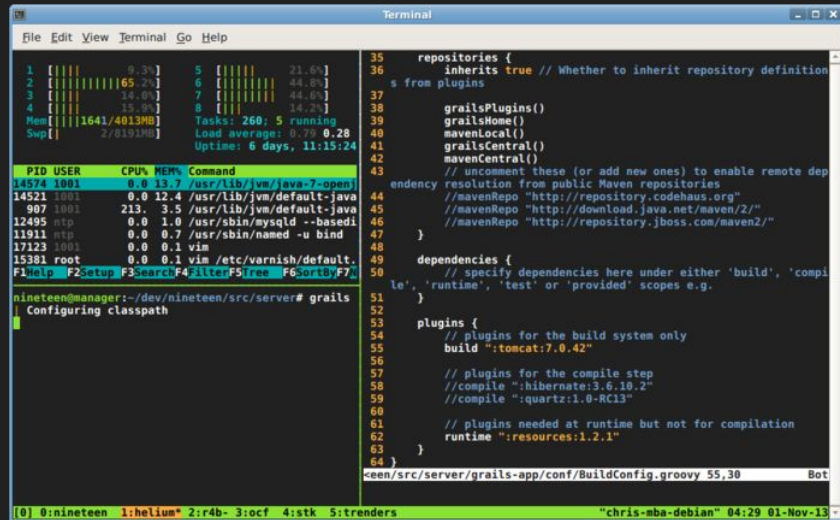
```
user1$ ssh user2@0.tcp.eu.ngrok.io \  
-p 17440
```

```
user1>
```

```
ngrok by @inconshreveable (Ctrl+C to quit)
Session Status      online
Account             Pavel Savchenko (Plan: Free)
Version            2.2.8
Region             Europe (eu)
Web Interface      http://127.0.0.1:4040
Forwarding          tcp://0.tcp.eu.ngrok.io:17440 -> localhost:22
Connections
  ttl    opn    rt1    rt5    p50    p90
   0     0     0.00  0.00  0.00  0.00
```

Tmux

- Tl;dr ALT+TAB for your shell
- Powerful tool when working in CLI
- Available in most repos via `$ brew install tmux`
- Easy to get started:
user1\$ `tmux new`
user2\$ `tmux attach`
- Use [wemux](#) for enhanced pairing features
- [Tmuxinator](#) - configure sessions
- [Xpanes](#) - tmux on steroids!
- Open source software



```
File Edit View Terminal Go Help
1 [||||| 8.3%] 5 [||||| 21.6%]
2 [|||||65.2%] 6 [||||| 44.8%]
3 [||||| 14.0%] 7 [||||| 44.8%]
4 [||||| 15.9%] 8 [||||| 14.2%]
Mem[|||||1641/4013MB] Tasks: 260: 5 running
Swap[| 2/8191MB] Load average: 0.79 0.28
Uptime: 6 days, 11:15:24

PID USER CPU% MEM% Command
14521 root 0.0 12.4 /usr/lib/jvm/default-java
987 root 213. 3.5 /usr/lib/jvm/default-java
12495 ntp 0.0 1.0 /usr/sbin/mysqld --basedir
11911 ntp 0.0 0.7 /usr/sbin/named -u bind
17123 root 0.0 0.1 vim
15381 root 0.0 0.1 via /etc/varnish/default.
F1=Alt F2=Setup F3=Search F4=Filter F5=Inc F6=Sort F7=

nineteen@manager:~/dev/nineteen/src/server# grails
| Configuring classpath

repositories {
    inherits true // Whether to inherit repository definition
    s from plugins
}
grailsPlugins()
grailsHome()
mavenLocal()
grailsCentral()
mavenCentral()
// uncomment these (or add new ones) to enable remote dep
endency resolution from public Maven repositories
//mavenRepo "http://repository.codehaus.org"
//mavenRepo "http://download.java.net/maven/2/"
//mavenRepo "http://repository.jboss.com/maven2/"
}
dependencies {
    // specify dependencies here under either 'build', 'compl
    le', 'runtime', 'test' or 'provided' scopes e.g.
}
plugins {
    // plugins for the build system only
    build "tomcat:7.0.42"
}
// plugins for the compile step
//compile "hibernate:5.6.10.2"
//compile "quartz:1.0-RC13"
// plugins needed at runtime but not for compilation
runtime "resources:1.2.1"
}
}
ken/src/server/grails-app/conf/BuildConfig.groovy 55,30 Bot

[0] @nineteen 1:helium 2:4b- 3:ocf 4:stk 5:trenders "chris-mba-debian" 04:29 01-Nov-13
```

VIM (or Emacs)

- Hugely popular, wide community
- Been around for ages (tons of info, tutorials and books online)
- Available for most OS-es, vim often pre-installed on Linux servers
- Highly configurable (can function as a simple text editor or full-blown IDE)
- Many plugins and customisations exist

Live Demo time!



A meme featuring the character Yoda from Star Wars. He is shown from the chest up, wearing his characteristic brown robes. He has a serious expression and is gesturing with his right hand as if giving a high-five. The background is a blurred purple interior, likely from a spaceship. The text 'HIGH FIVE' is written in large, white, bold, sans-serif capital letters at the top, and 'YOU MUST ME' is written in the same style at the bottom.

HIGH FIVE

YOU MUST ME

Tool Resources

- ngrok
 - <https://github.com/dwyl/learn-ngrok>
 - <https://ngrok.com/docs>
- tmux
 - <https://leanpub.com/the-tao-of-tmux/read>
 - <https://thoughtbot.com/blog/a-tmux-crash-course>
 - <http://www.hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/>
 - <https://hackernoon.com/a-gentle-introduction-to-tmux-8d784c404340>
 - [Freecodecamp: Tmux in practice: local and nested remote tmux sessions](#)
 - <http://hyperpolyglot.org/multiplexers>
- vim
 - vimtutor
 - <https://www.openvim.com/>
 - <https://danielmiessler.com/study/vim/>
 - <https://vim-adventures.com/>
 - <https://chrome.google.com/webstore/detail/vimium/dbepggeogbaibhgnhhndojpepiihcmeb>

Windows Resources

- run Tmux in GIT Bash on Windows - <https://blog.pjsen.eu/?p=440>
- [Running SSH Agent when starting Git Bash on Windows](#)
- [VSCode Live Share](#)
- ...

General Resources

- <http://wiki.c2.com/?PairProgrammingBenefits>
- <https://www.pairprogramwith.me/>
- <https://www.youtube.com/watch?v=f9ALSPIPQhg>
- https://www.youtube.com/watch?v=rG_U12uqRhE (cheesy voiceover)
- <https://www.youtube.com/watch?v=1RYI9FaT9UQ>
- <https://ryanlue.com/posts/2018-01-13-pair-programming-over-ssh> (mac-ish)

Conclusion

- Easy to get started with
- Invaluable for remote teams
- Fun

Questions?

Reach out to me on slack: @Pavel Savchenko

Or telegram: @pavelsavchenko

