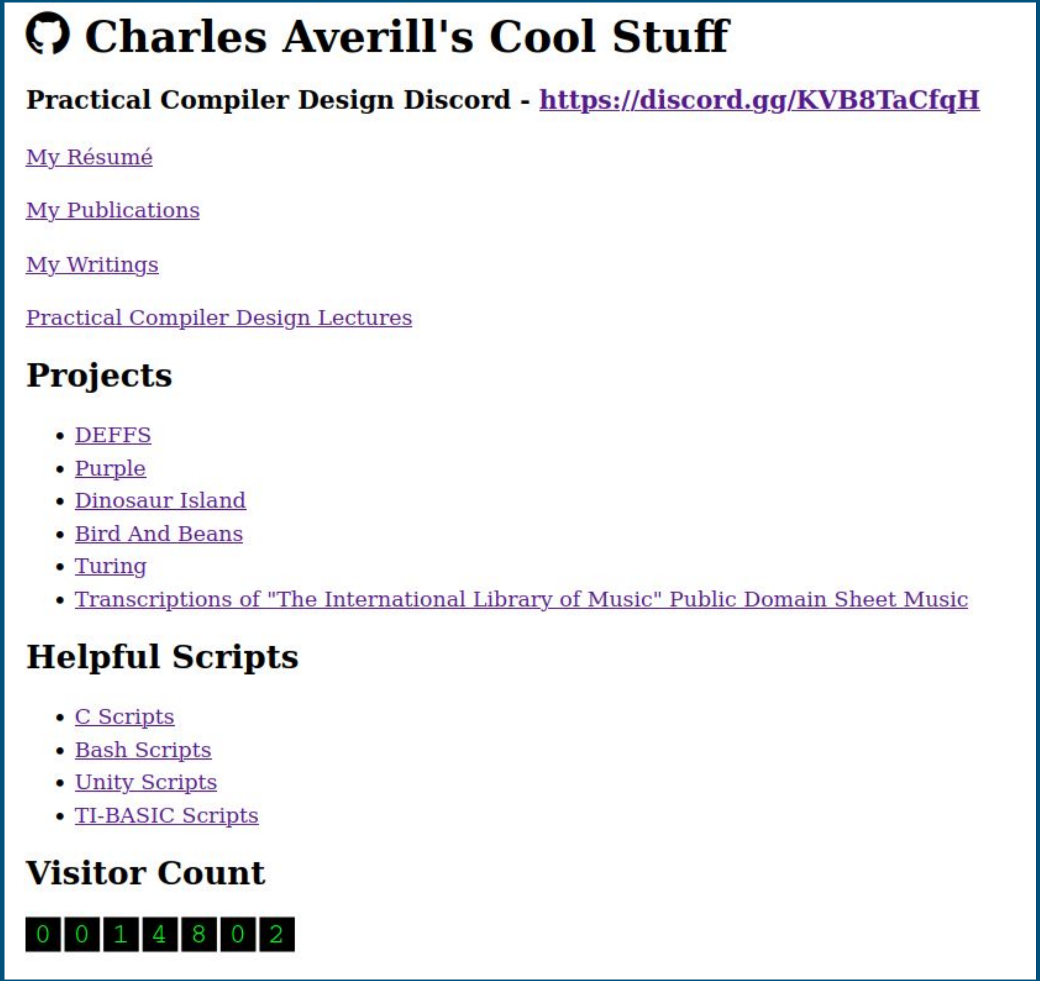
Dr. BrickMachine, or: How I Learned to Stop Worrying and ~~Love~~Hate The Apache HTTP Server Project

Charles Averill

This is my website:

https://www.charles.systems



Charles Averill's Cool Stuff

Practical Compiler Design Discord - https://discord.gg/KVB8TaCfqH

My Résumé

My Publications

My Writings

Practical Compiler Design Lectures

Projects

- DEFFS
- Purple
- Dinosaur Island
- Bird And Beans
- Turing
- Transcriptions of "The International Library of Music" Public Domain Sheet Music

Helpful Scripts

- C Scripts
- Bash Scripts
- Unity Scripts
- TI-BASIC Scripts

Visitor Count

0 0 1 4 8 0 2

It Does What I Need

- Host my Resume
- Show off writeups/code for my favorite projects
- Send static data to the people who want to know more about me

What It Used To Be

- For a long time it was hosted on Github Pages
- GH Pages did exactly what I needed, and I could hook up my cool custom URL
- I could just push to my website repository and have an update within a few minutes

Issues

I like GH Pages. But it has some issues:

- No dynamic content
- Repository size limits
- I got tired of reloading the Action that published the new site

Solution

I will self-host my own web server! Shouldn’t be too hard:

1. Buy a cheap machine
2. Install Ubuntu Server
3. Install Apache
4. Update DNS Records

Attempt #1

- Bought the machine in August
- Half set up the server
- School started
- Attempt dead

Attempt #2

- Reinstalled Ubuntu Server
- Was humbled by my previous attempt, so I was more obedient to tutorials
- It worked! Self-hosted website that sits next to my couch and has secondary function of drink holder
- Still using Github as a remote, centralized server so that I can update my website from anywhere that I’m authenticated at

Where it Went Wrong

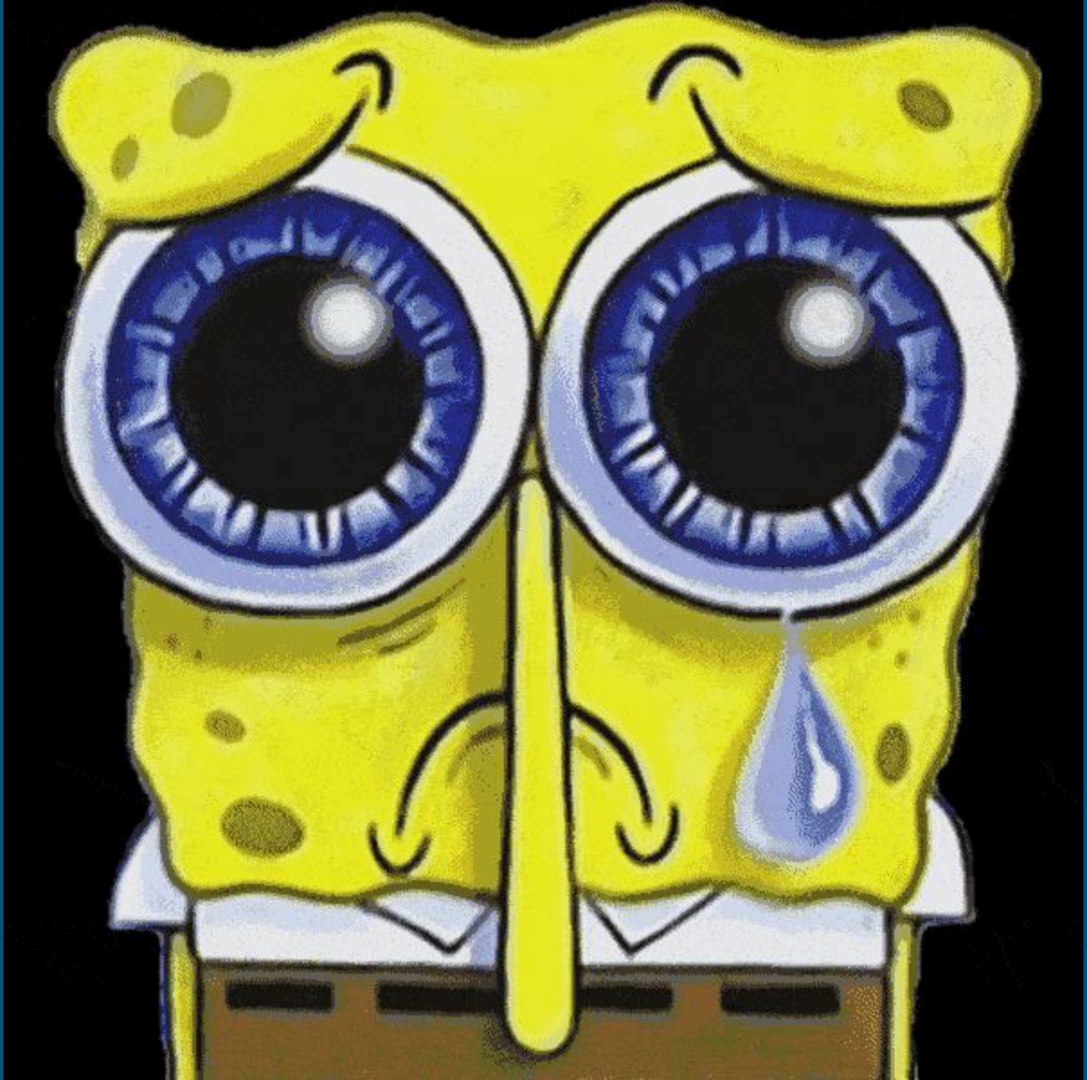
Game Development Project

- Rebuilding a now-defunct mobile game whose servers have been taken offline
- It's really big (200MB)
- It is a good thing I am hosting my own web server!
- https://www.charles.systems/AT_CWD

Game Development Project

- Initially, project source is public but I’m not accepting PRs until it grows a little bit. Therefore, code lives on my desktop with no version control
- Eventually, I’m considering allowing PRs
- All of my professors who have done research set up Gitlab instances for their students to use. OK, I’ll do that

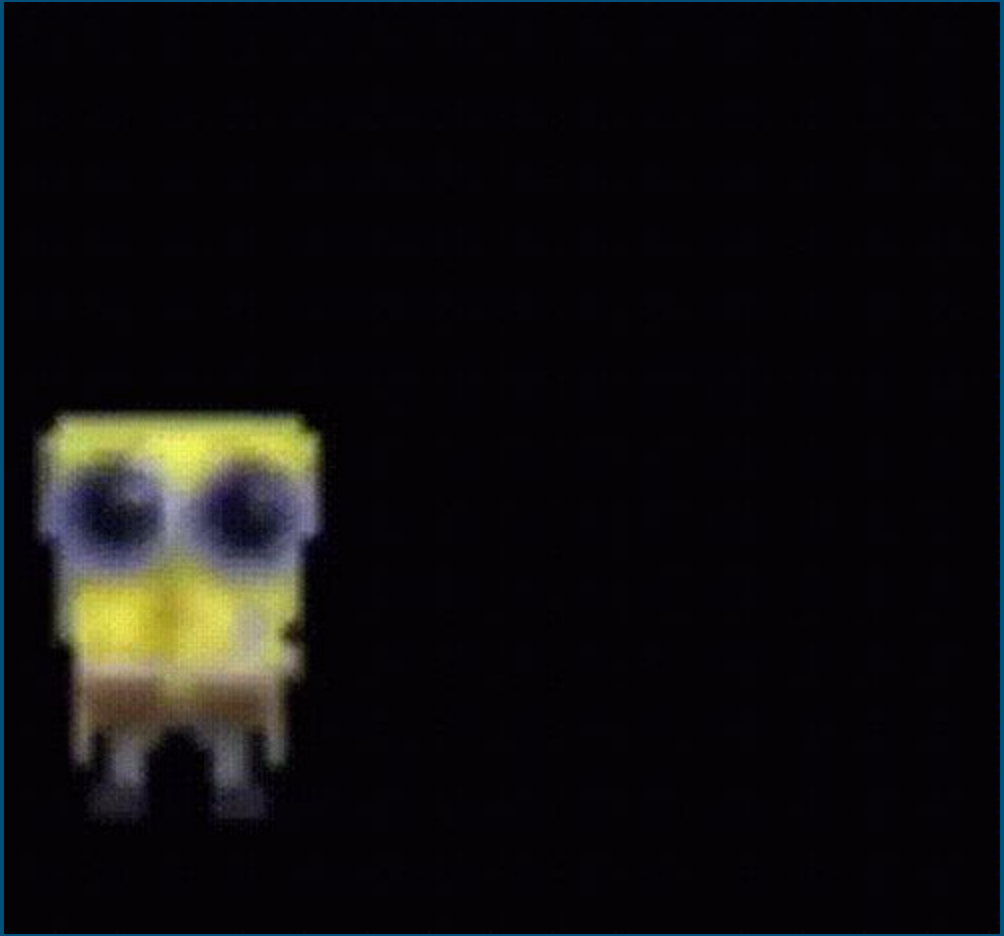
Nope!



Gitlab was mean to me

- Gitlab initially seemed promising: cool UI, PR support, user accounts, public access
- Unfortunately, its git server would not cooperate with Apache!
- Maybe it was because I was using a subdomain instead of a standard URL path (git.charles.systems) for the gitlab instance? Probably not, I tried a standard path as well
- Oh well. Surely one of the other available git frontends will work

Nope!



Gitea and GOGS were mean to me

- Neither would progress past the database setup/installation page
- Normally I would not let this stop me, but I was tired

It's Time to Go Old School

- Who needs a UI?
- People have been sharing code remotely for *dozens* of months
- I will just host a plain-Jane git server, no frou-frou pull requests, just old school badass ssh access

How will people clone the repo?

- I don’t want people to make accounts on my machine, people break things
- I will make a ‘public’ user with readonly access to the files I explicitly specify
- I’m a pro linux user of 6 years, I will do this by memory (surely a good idea)

What I Did

- useradd public
- passwd public # set the password to ‘public’

Okay, this is fine

What I Did

- useradd public
- passwd public # set the password to ‘public’
- sudo

Okay, sure I need sudo sometimes

What I Did

- useradd public
- passwd public # set the password to ‘public’
- sudo chmod

Okay I can see the thought process here, we want to restrict the public user’s access

What I Did

- useradd public
- passwd public # set the password to ‘public’
- sudo chmod -rwx

(Oh no)

What I Did

- `useradd public`
- `passwd public # set the password to ‘public’`
- `sudo chmod -rwx /home`

(Oh no...)

What I Did

- useradd public
- passwd public # set the password to ‘public’
- sudo chmod -rwx /home /var /srv

(Oh no...)

What I Did

- useradd public
- passwd public # set the password to ‘public’
- sudo chmod -rwx /home /var /srv /usr

(OH NO)

My Intention

Make it so that the ‘public’ user has no permissions in those directories (so that I can explicitly add permissions as I desired)

What I Forgot

- ‘chmod’ can’t set permissions for an individual user
- All non-builtin bash commands live in /usr
- Ubuntu Server has root login disabled by default
- I didn’t make any other accounts on the system

This Sucks... But It’s Salvageable

- After recognizing my defeat for what it was, I had some cereal and reflected on my poor decisions
- “Oh well, all that Apache setup down the drain. But at least I know how to do it. I guess I’ll reinstall my operating system and start over. All of my actual content is still backed up to Github, I just have to configure everything again.”
- WAIT

A Better Fix

- I was two clicks from wiping my drive and reinstalling Ubuntu when a friend suggested using ‘chroot’ to replace the original permissions
- New plan:
  a. Create Ubuntu Desktop live boot (UServer doesn’t come with a live boot distro)
  b. ‘mount /dev/sda /mnt/recover && chroot /mnt/recover’ within UDesktop live boot to become root user in UServer’s filesystem
  c. ‘chmod +rwx /home /var /srv /usr’ to replace the broken permissions

That worked!

- Everything back to normal
  - Mostly, ~/.ssh was given full permissions, so my cron job that pulls from Github kept crashing because my ssh key was not protected enough. Quick fix though.
- Properly set permissions with ‘setfacl’ instead of chmod
- My fans can now download the code with:

```
git clone public@71.132.166.91:/srv/git/cwd.git

# password is ‘public’
```

What I Learned

- Don't be ~~stupid~~reckless
- Don't blindly mess with permissions
- Web server administration is not something I want to become proficient at
- Apache Sucks Ass. None of this would've happened if it had let me configure gitlab properly!