My Mini-Blog

MY1 C Compiler

Guess what I have been doing for the last month? I call it my1cc (duh!)... and I have published my initial work at codeberg.org. Obviously it is wayyyyy from being usable, but the parser seems to work... somewhat. In the end, I was hoping to develop a usable C compiler for 8051/8085 platforms, so that I can use it in my microcontroller/microprocessor classes. I do have to pause a bit... hopefully I can resume A.S.A.P.

But, DO NOT hold your breath for that 😬

That is all I have to say about that.

2026/02/08 04:56
programming, technology

SPM-Level Computer Science

It turns out, Computer Science (CS) for SPM (Malaysia Certificate of Education @MCE) deals mainly with database and web interface. I have developed database-related systems before, but I was doing it without having formal class on it. Basically, I simply wing-it based on what I know on how computers store data (i.e. C data types).

I need to help my son with some questions, so I looked up 'database normalization' and found some things. The good thing is the way I implemented my database is mainly the 'correct' way. The not-so-good thing is that I am NOT able to help my son much with the practice questions - simply too many terms that I am not familiar with. Just shows that implementations need not be theoretically developed - experience (and good fundamentals) can also help to produce relatively sufficient results.

Anyways, the main reason I am writing this here is to 'store' the following 'note' (for my personal future reference).

db_normalization.txt

```
Database Normalization

Five 'rules' to make data normal: 1NF,2NF,...,5NF (NF=normal form)
- each rule builds on another, starting from 1NF
- 1NF/2NF/3NF -> Core Basics (normalization usually means 3NF!)
- 4NF/5NF -> Exceptions

*Note: Normalization is about grouping & connecting data the right way!

1NF is about
- atomic values
```

```
- unique identifiers

*Term: Imagine a spreadsheet -> [table] or [entity]

1NF rules
- a cell cannot contain more than 1 value
  = if it does, that column needs to be split into multiple columns
- each row (@record) must be unique
  = look for potential primary key
  = usually, we use system-generated (integers are better!)
- each column name must be unique
- there must be no repeating groups (or cells?)
  = it there are, remove and create new table (1NF!)

2NF rule
- all data/column(s) must depend on the primary key
  = if it does not, must be split into it own table (1NF!)
  = use primary key in new table as column value -> foreign key

3NF rule
- primary key must define all non-key column(s)
  = non-key column(s) must not depend on any other key
  = if this is not met, must create new table and use foreign key to
link
```

2026/02/08 04:54
computing

Why engineering students should not use Arduino

I am one of those tertiary educators in engineering (E&E) who is against the idea of Electronics Engineering students using Arduino for their bachelor degree project. This is a great example why that is the case. I know that is about software, but still the same principle.

When your system depends on other people's work (i.e. you use library/libraries most of the time in Arduino), there is always a possibility it cannot be maintained in the future. And, for engineering students, the most important thing for you is to be competent at creating that 'building block' (@fundamental code/design) on your own… so that even if you do not need to write/develop any, your can still fix it when something no longer works.

That is all I have got to say about that… for now 

2026/02/08 04:52
education, electronics

Broadcom-sta on Slackware 15 for old laptop

I need to use my old laptop that requires broadcom-sta driver for its WiFi module.

Updated20240301: *Updated script*

getbuild_broadcom-sta

```
#!/bin/bash

TOOL="wget"
TCHK=$(which $TOOL 2>/dev/null)
[ ! -x "$TCHK" ] && echo "** Cannot find $TOOL!" && exit 1

OPTS="--recursive --no-parent --no-host-directories"
OPTS="$OPTS --reject index.html*"
OPTS="$OPTS --cut-dirs=3"
OPTS="$OPTS --execute robots=off"

FROM="https://www.slackware.com/~alien/slackbuilds/broadcom-sta/build/"

$TOOL $OPTS $FROM
echo "[DONE] $TOOL $OPTS $FROM"
# https://github.com/winterheart/broadcom-bt-firmware
```

That is all I have to say about that.

2026/02/08 04:57
linux, slackware

Valgrind

Just a self reminder - I actually only just found out how useful `valgrind` can be 😛

Simple run

```
$ valgrind --tool=memcheck --leak-check=full --track-origins=yes -s
<command>
```

Or, with logger

```
valgrind --leak-check=full --track-origins=yes --log-file=valgrind.rpt
<command>
```

Maybe I will elaborate on this some other time.

2026/02/08 04:58

programming

Older entries >>