note - 2

Best programming method C

```
struct x {
        structs vars
    char x;
    int y;
    char* s;
        struct pointer functions

      char (*func1) (param ... );
      int (*func2) (struct* x, para);
           etc
}
```

- encapsolation
- no conflict in naming routines and parameters
- spawn multi instances

```
typedef struct x X;
```

-> Function pointer

```
    func1 = func_1 (...)
    func2 = func_2 (...)
```

- if linux had root abstraction has this, would simplify imense code, and hardware would be better manipulated.

-> Function Definitions

```
struct x    func_1 (*self) { ... }
X           func_2 (*self) { ... }
```

make it return a copy of the struct* x.

-> this way can create an object representation of the hardware, returning its parameters as copy.

∴ X.func_1 (*self...) . parameter

- safe and indirect access to source data.