Embedded C Programming

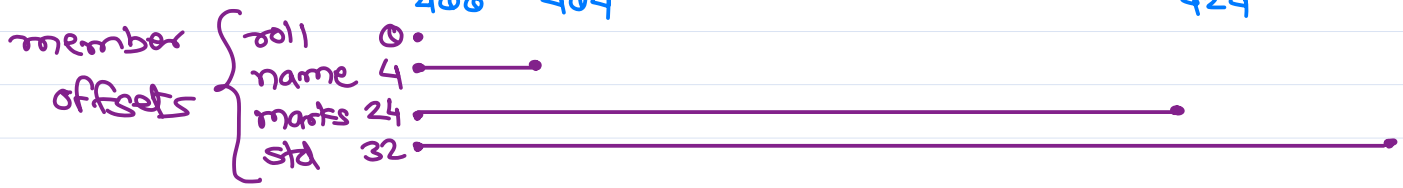
Trainer: Nilesh Ghule

offset Calc

① &s1.std − &s1
(424) − (400) = 32

② p = NULL;
&p→std = 32

③ &((struct student*)0)→std = 32
type, member

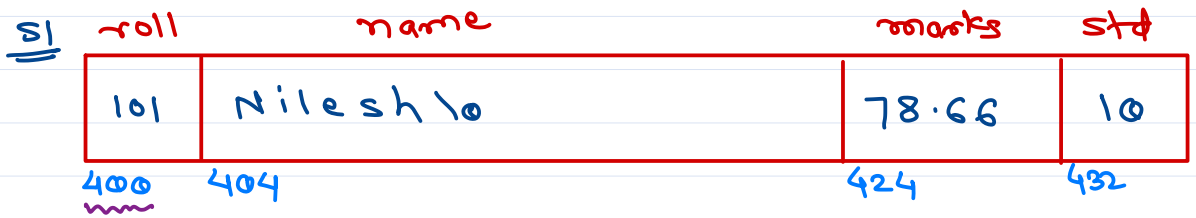


&s1 = 400
&s1.roll = 400 → *(400+0)
&s1.name = 404 → (400+4)
&s1.marks = 424 → *(400+24)
&s1.std = 432 → *(400+32)

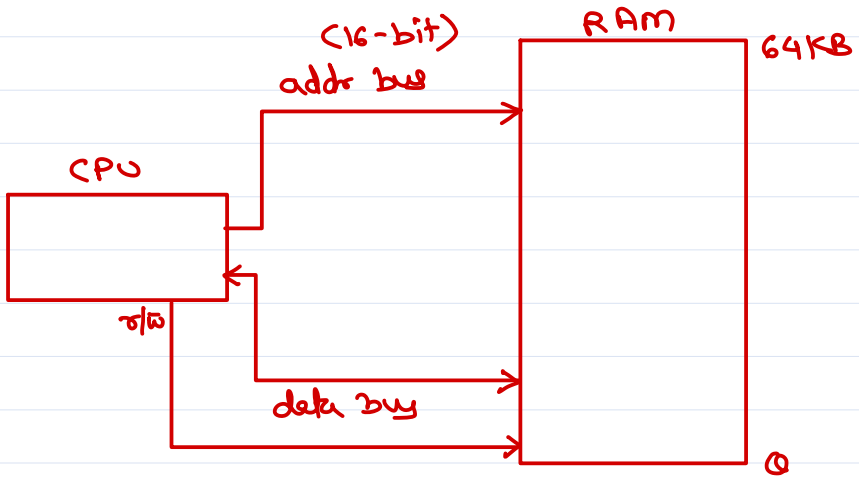
④ #define offsetof(type, member) ((long)(&((type*)0)→member))

pf("%ld", offsetof(struct student, std));

Note: size_t = unsigned long
loff_t = long



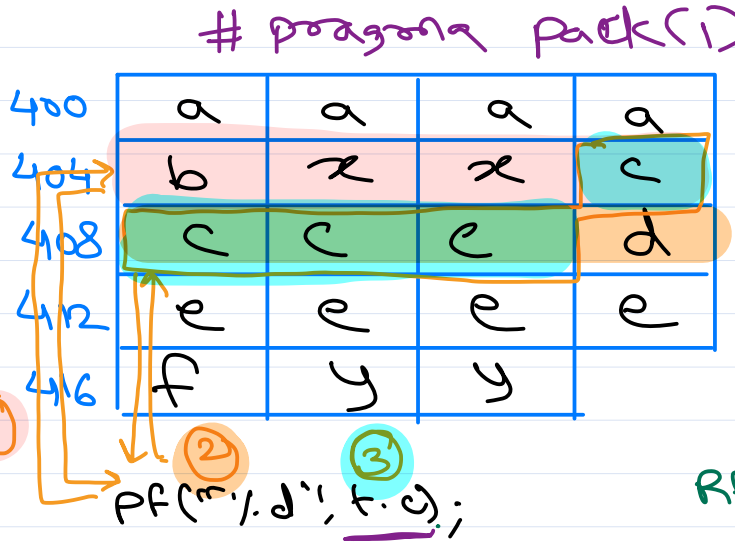

```
printf("s1 marks = %.1f\n",
        *(double*)((char*)&s1 + 24));
```



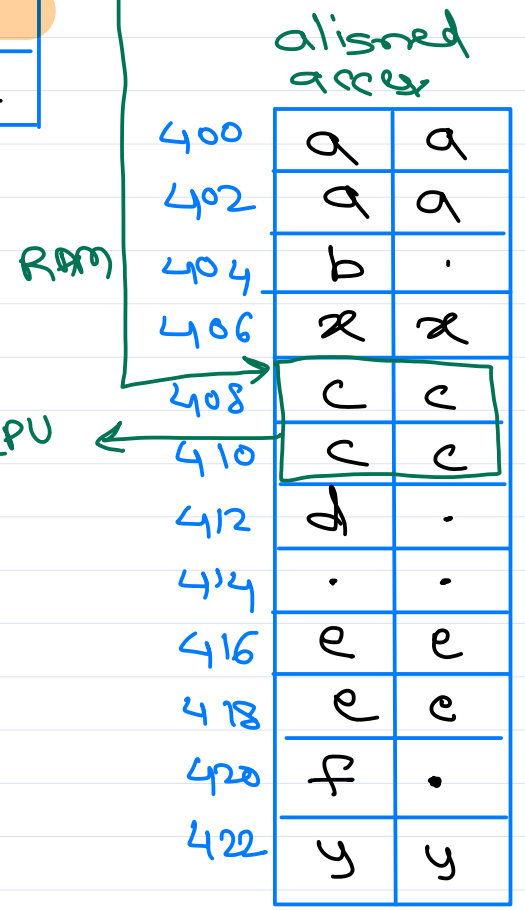
In some CPU arch like ARMv4, unaligned mem access is not possible.

unaligned data access
↳ CPU can read data of any size from any memory addr.

aligned data access - efficient
↳ CPU can read data of a size from an addr that is multiple of size of data.
- int → addr must be multiple of 4
- short → addr must be multiple of 2
- char → addr must be multiple of 1




```
struct test t;
pf("%.d", t.c);
```



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>