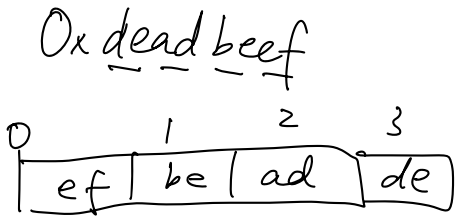
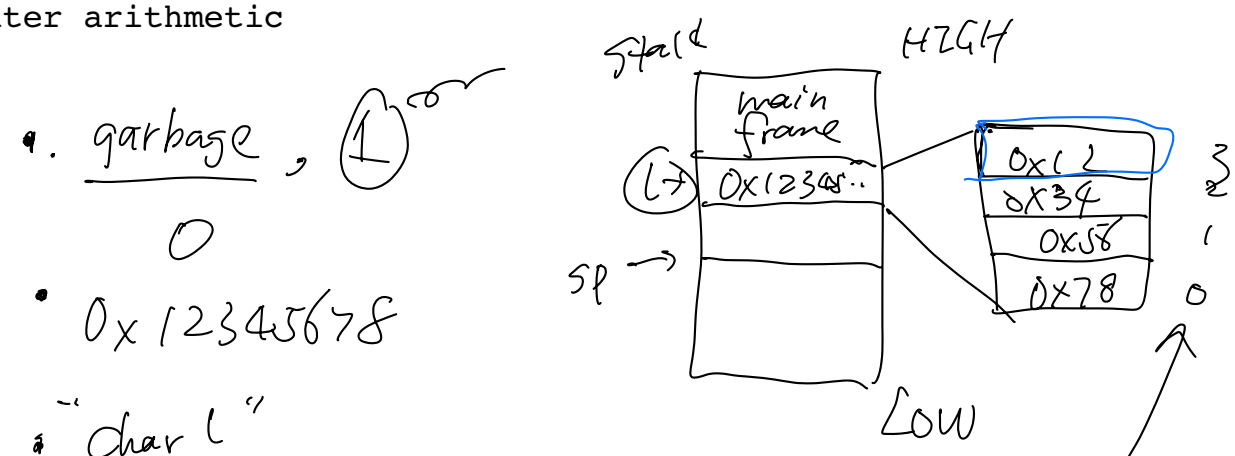
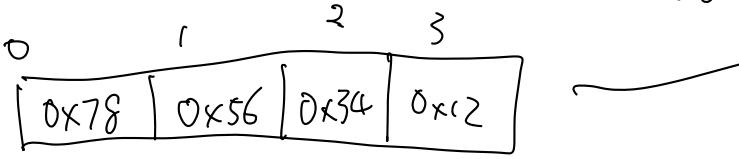
Week 2.b
CS6640
09/14 2023
https://naizhengtan.github.io/23fall/

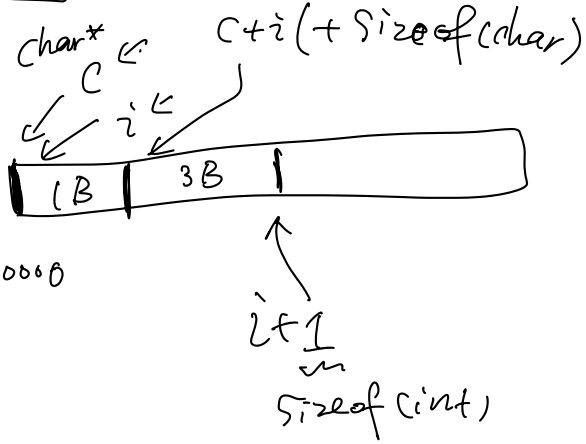
1. C pointers (cont'd)
2. C arrays
3. Some C keywords
4. Bitwise operators
5. labs setup
6. User-level threading (lab2)



- uninitialized local variables on stack
- pointer arithmetic

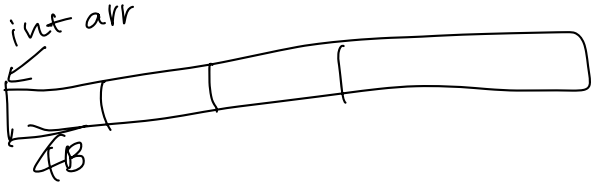


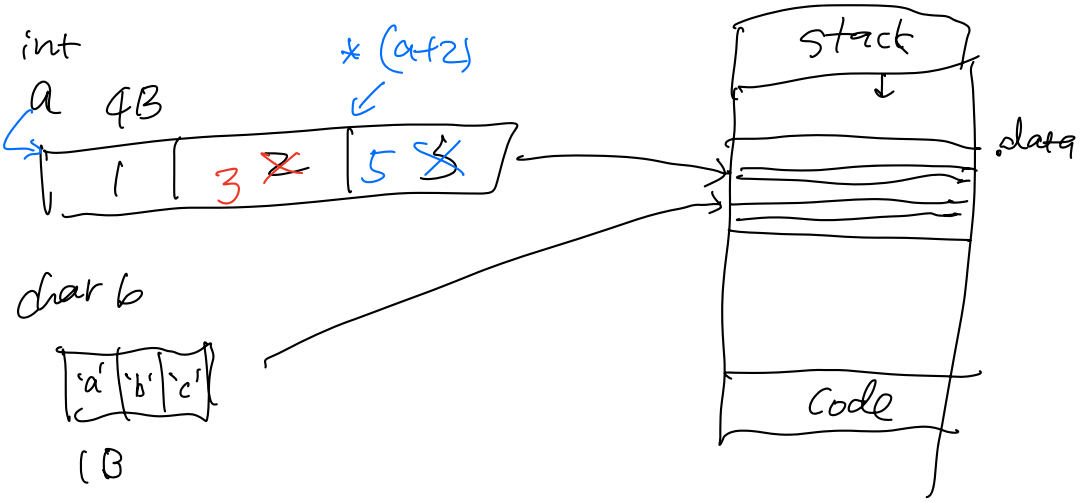


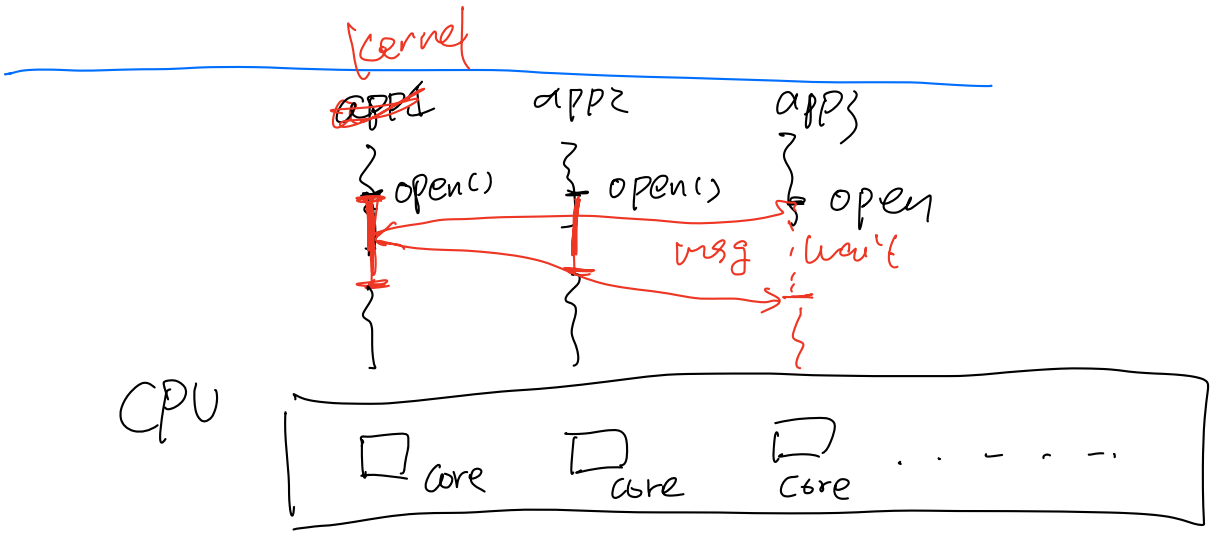


- 0x08200001
  0x08200004

- array




int
a 4B
* (a+2)
1
3
5
stack
data
char b
'a' 'b' 'c'
1B
Code


kernel
app1
app2
app3
open()
open()
open
msg
wait
CPU
core
core
core

Bit fields:

```
struct valid_addr {
  int valid : 1;
  int addr  : 31;
};

struct valid_addr a;
a.valid = 1;
a.addr = 0xdeadbeef;
```

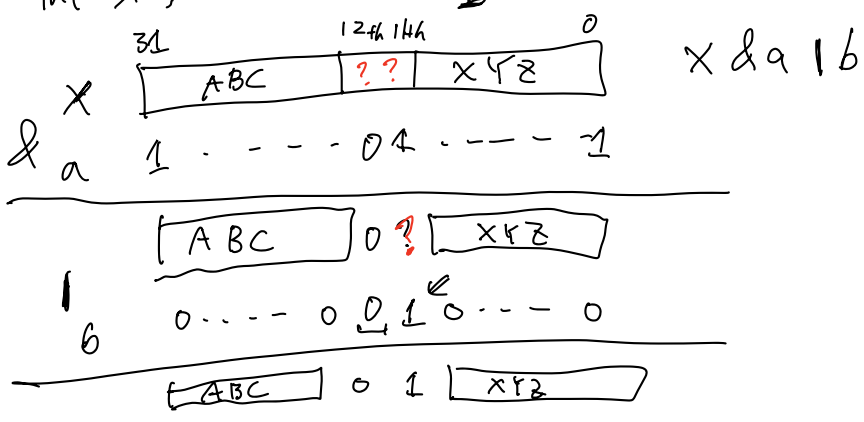
- you can manipulate bits with |, &, ~, ^

```
10001 & 10000 = 10000   // bit-wise and
10001 | 10000 = 10001   // bit-wise or
10001 ^ 10000 = 00001   // bit-wise xor
~1000 = 0111            // bit-wise not
```

0001 << 12 ⇒ 100...0 (12)

100...0 (12) >> 12 ⇒ 000..1

Q: int X, 11th, 12th ⇒ 01



X & a | b

stack

```
int mstatus;
asm("csrr %0, mstatus" : "=r"(mstatus));
asm("csrw mstatus, %0" ::"r"((mstatus & ~(3 << 11))   /* clear MPP */
                             | (1 << 11) ));  /* set MPP to S */
```

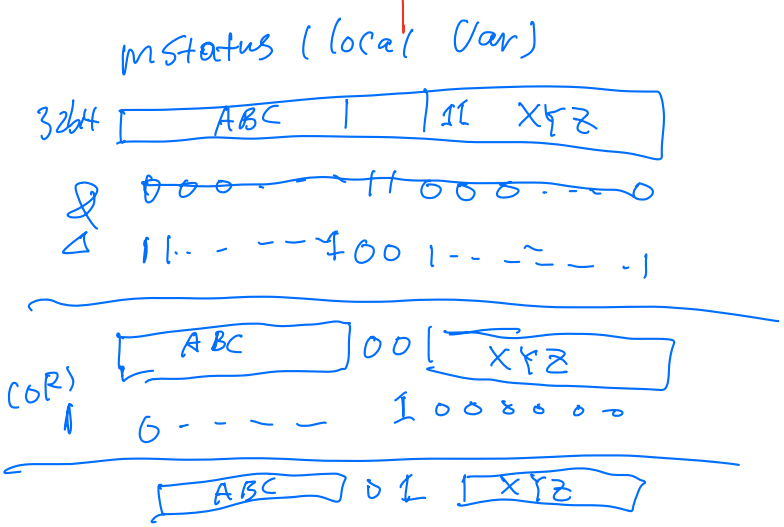
CSR

① asm?

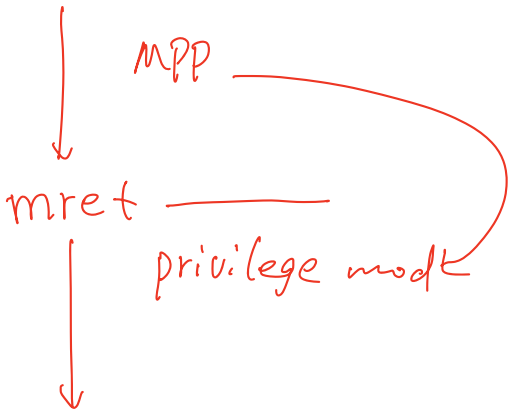
② csrr, csrw

③ mstatus

④ ?



MPP	2bits	
U:	00	(0)
S:	01	(1)
M:	11	(3)



- labs