

Let's solve it

37

- CPU cycle (Read/Write memory)

Minimizing CPU cycle

struct xyz
{
~~int~~

char a; //

~~int~~ b; // 4 int assume

char c; //

Boundary

WS

};

4 byte

1

char

1

2 6 byte

3 (int)

int

Splitted

char

1

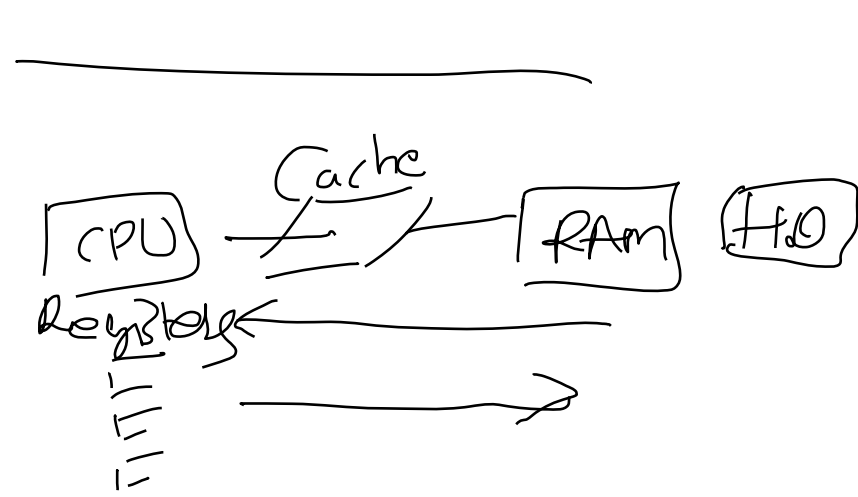
sh

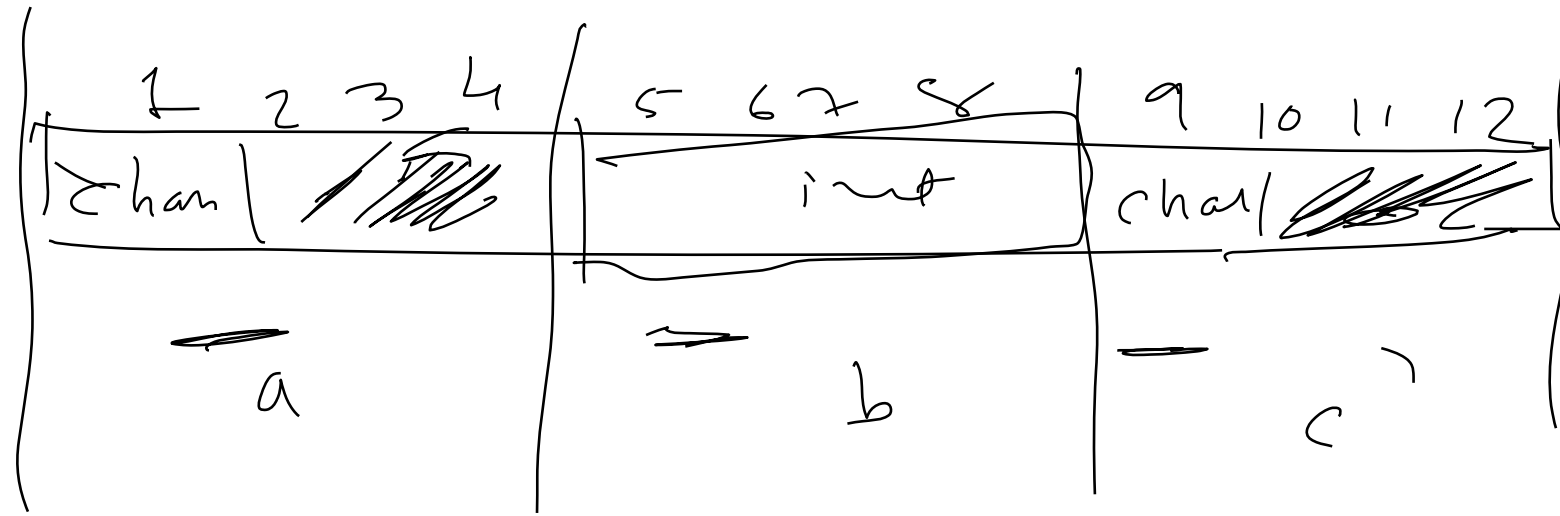
XYZ d1;

d1. b

~~int~~

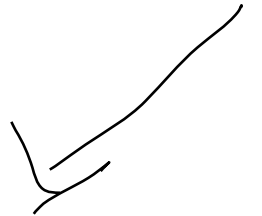
2 CPU cycles





12 bytes

12 bytes allocated. ✓



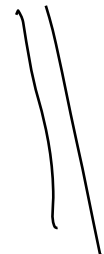
struct A

{

4 // `char` `a`;
 4 // `char` `b`;
 4 // `int` `c`;

};

8 bytes



struct B

{

`int a`; // 4 bytes
`char b`; // 1 byte
`char c`; // 1 byte

};

8 bytes

Bit fields

Concern about memory
~~it~~ in the situation where
members require bit or few bits only for
their purpose.

short
int

flag — 0
 — 1

It only requires single bit < byte
= (8 bits)

Gender

marital status

work of the day
1 - 7

1 1 1
- - -
3 bits

struct person

```
{ int id; ✓  
  char name[25]; ✓  
  int marital_status: 7;  
  int gender: 2;  
  int age: 5;  
};
```

main()

```
{ struct person p;
```

```
  p.age = 27;
```

```
  p.id = 32667;
```

struct Person

```
{
```

// bit field
// reg

// reg

// bit fields

Bits only

```
};
```

reg

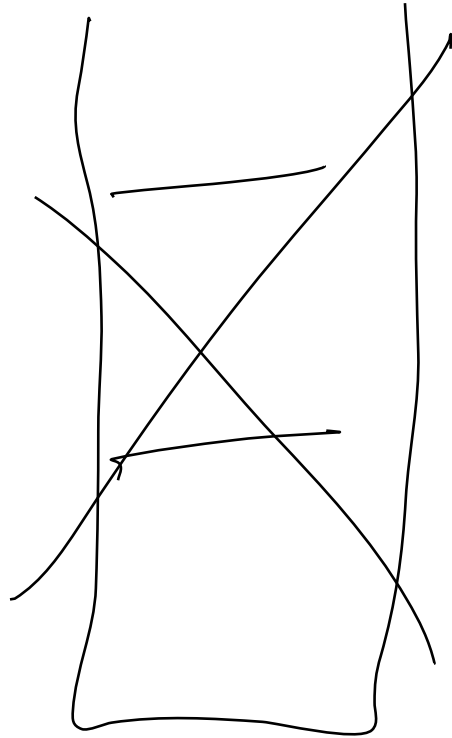
What if
I have allocated
2 bits and
store more.

$a \times b \Rightarrow a + a + \dots + a$ // b times

$\Rightarrow a + (a \times (b-1))$
func(9, 7)

func(3, 3);

9



func(int a, int b)

```
{  
    if (b == 0) ✓  
        return 0;  
    if (b == 1) ✓  
        return a;  
    return (a + func(a, b-1));  
}
```