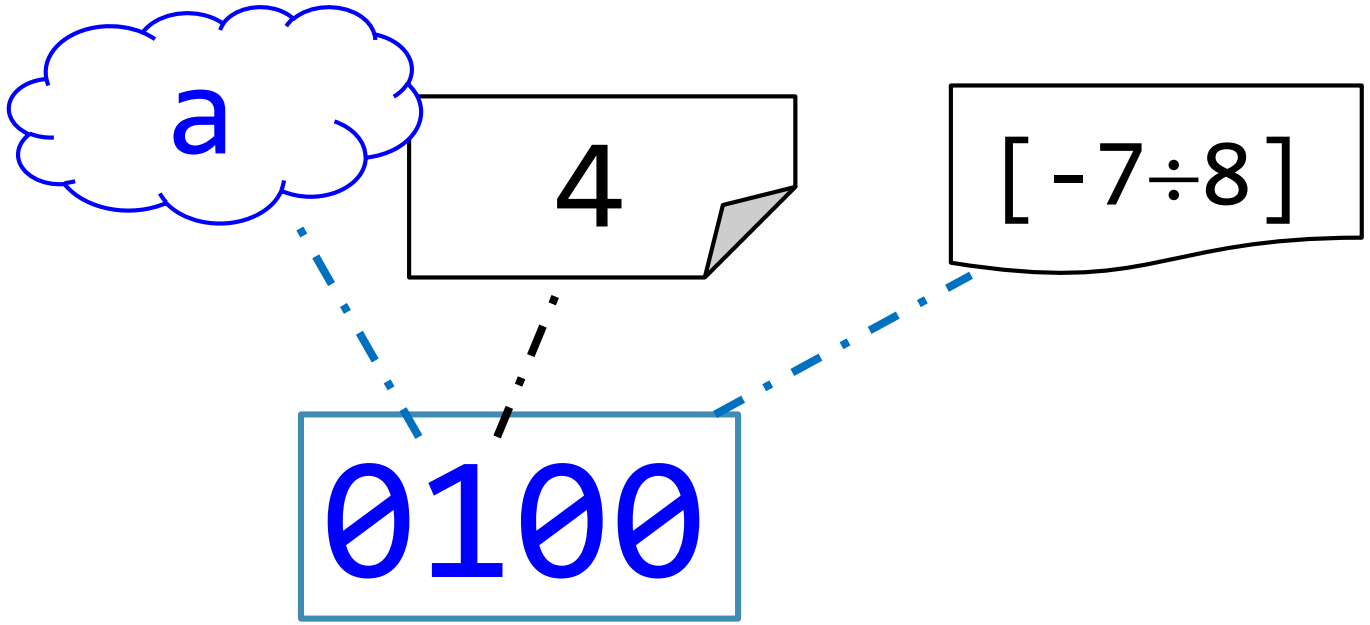
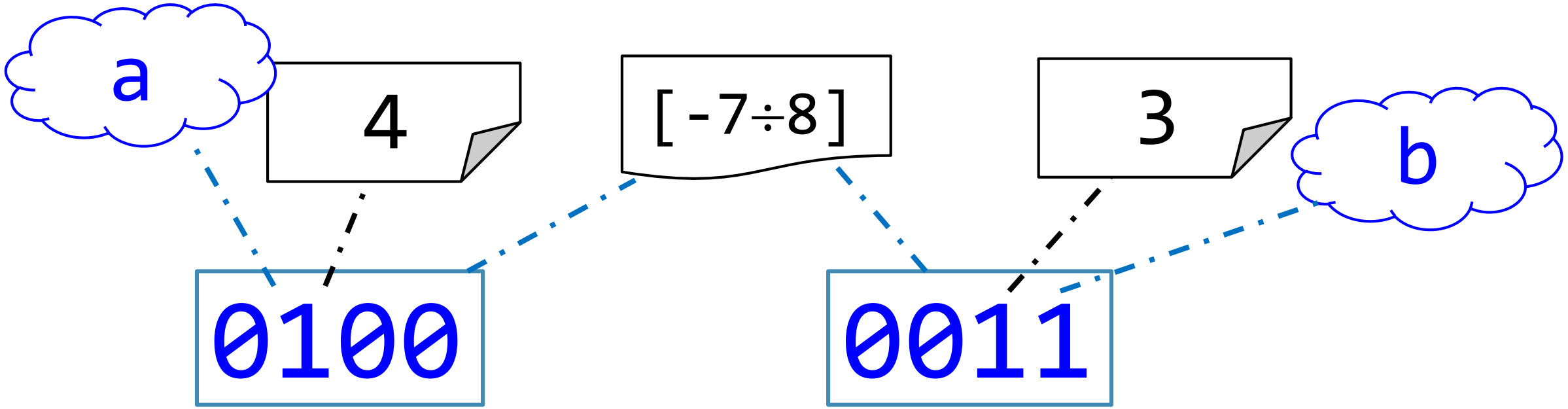
GENESIS

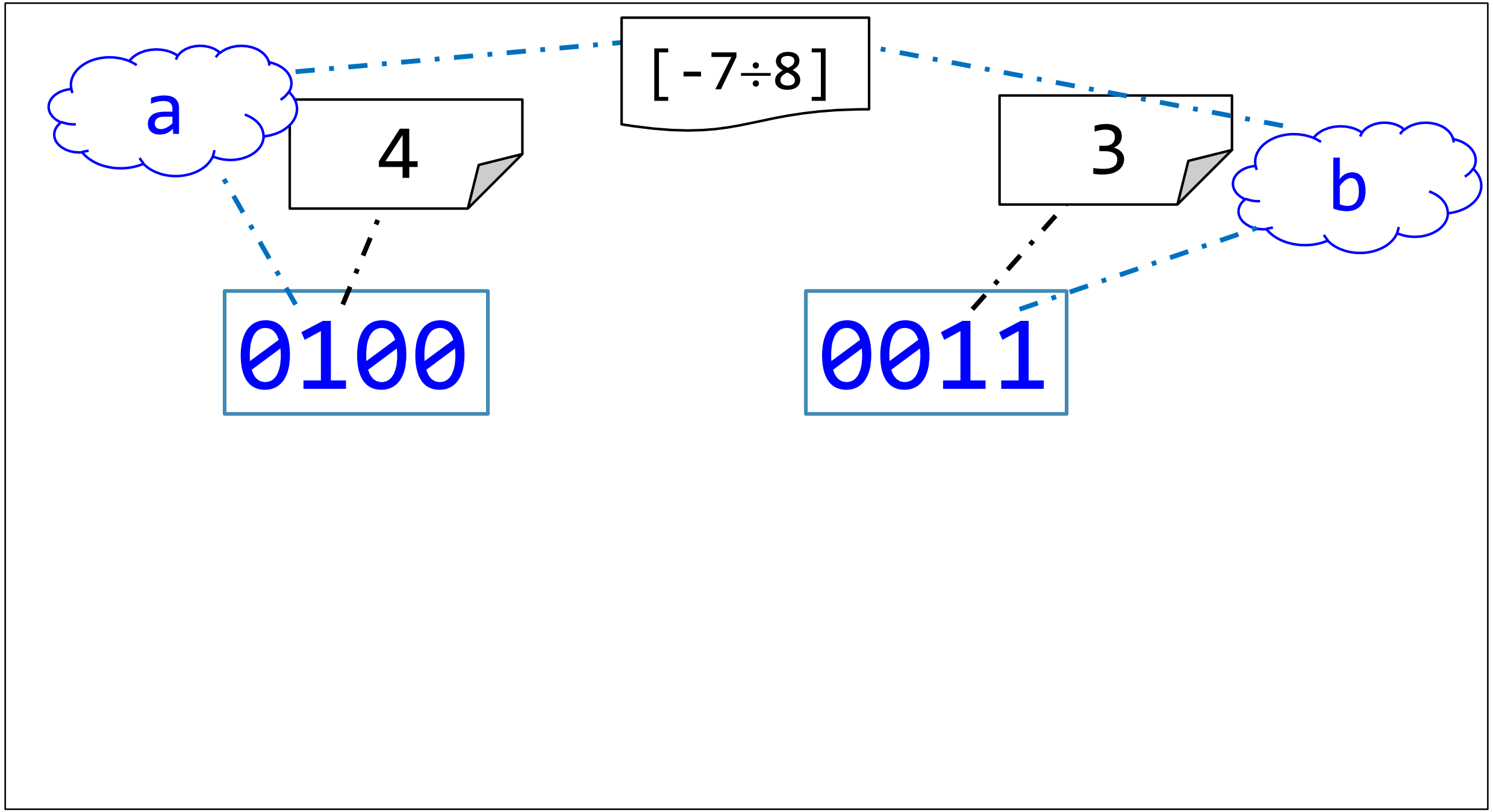
минимум про массивы и указатели

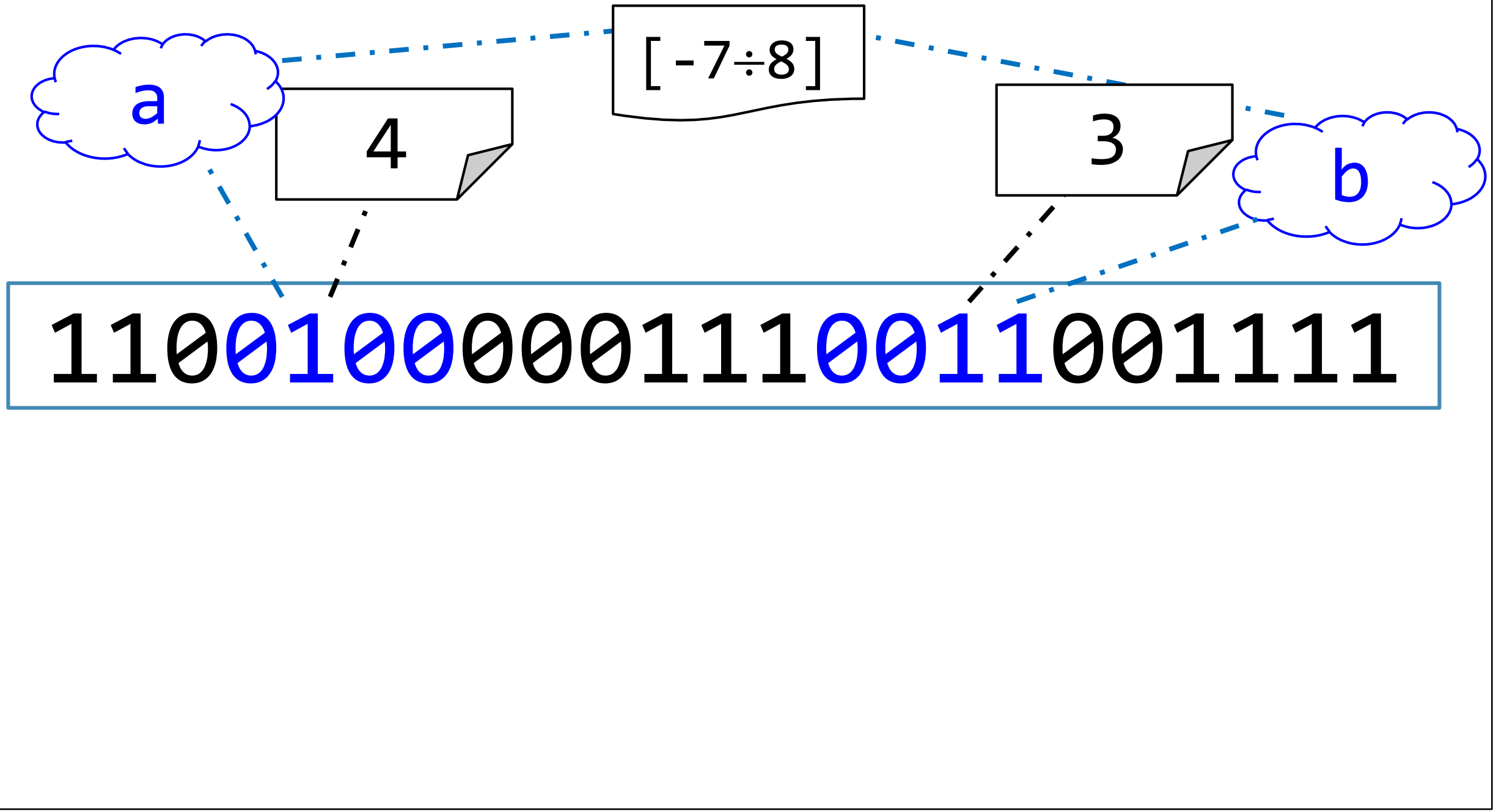
К. Владимиров, Intel, 2018
mail-to: konstantin.vladimirov@gmail.com

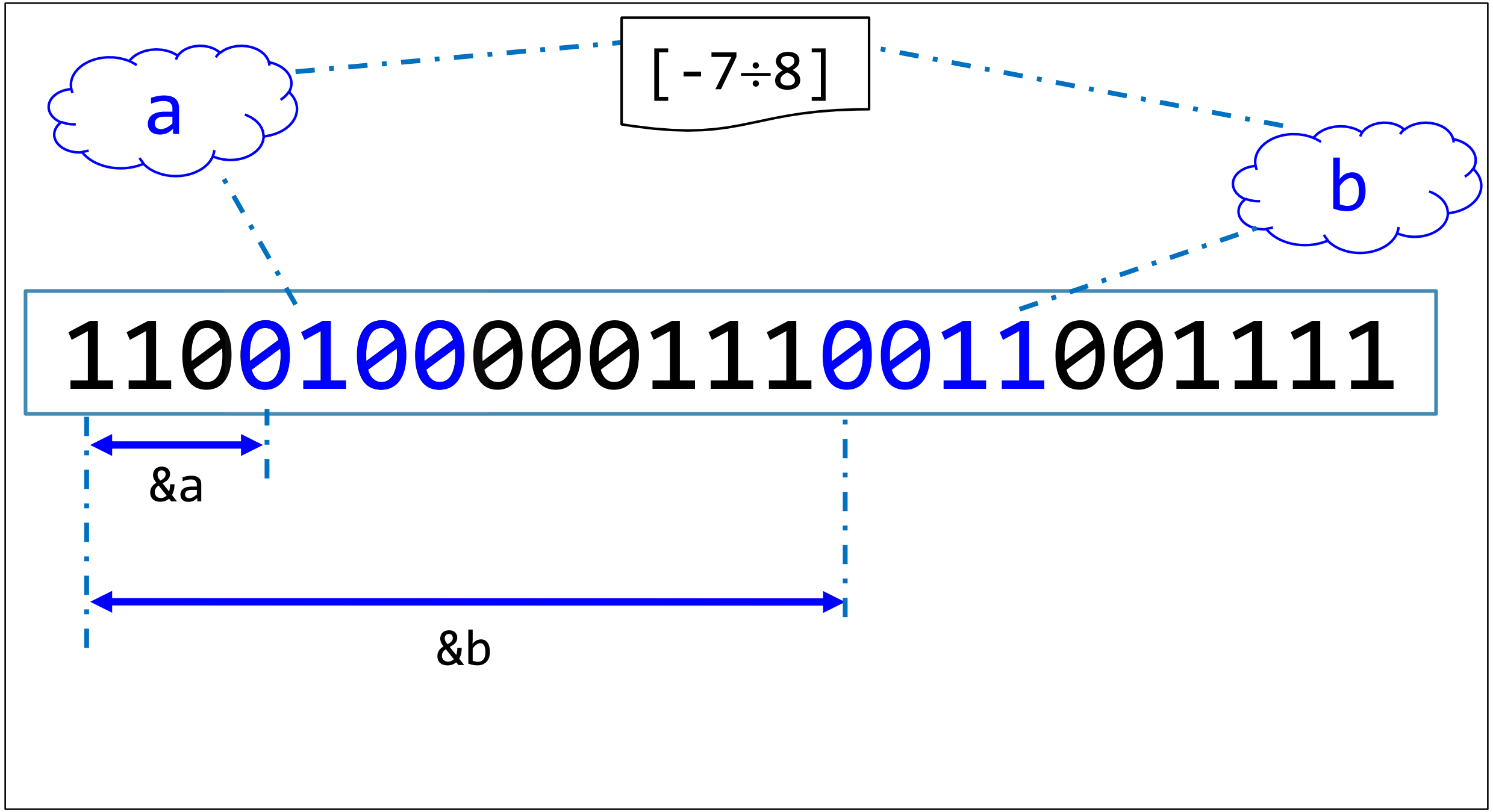
0100

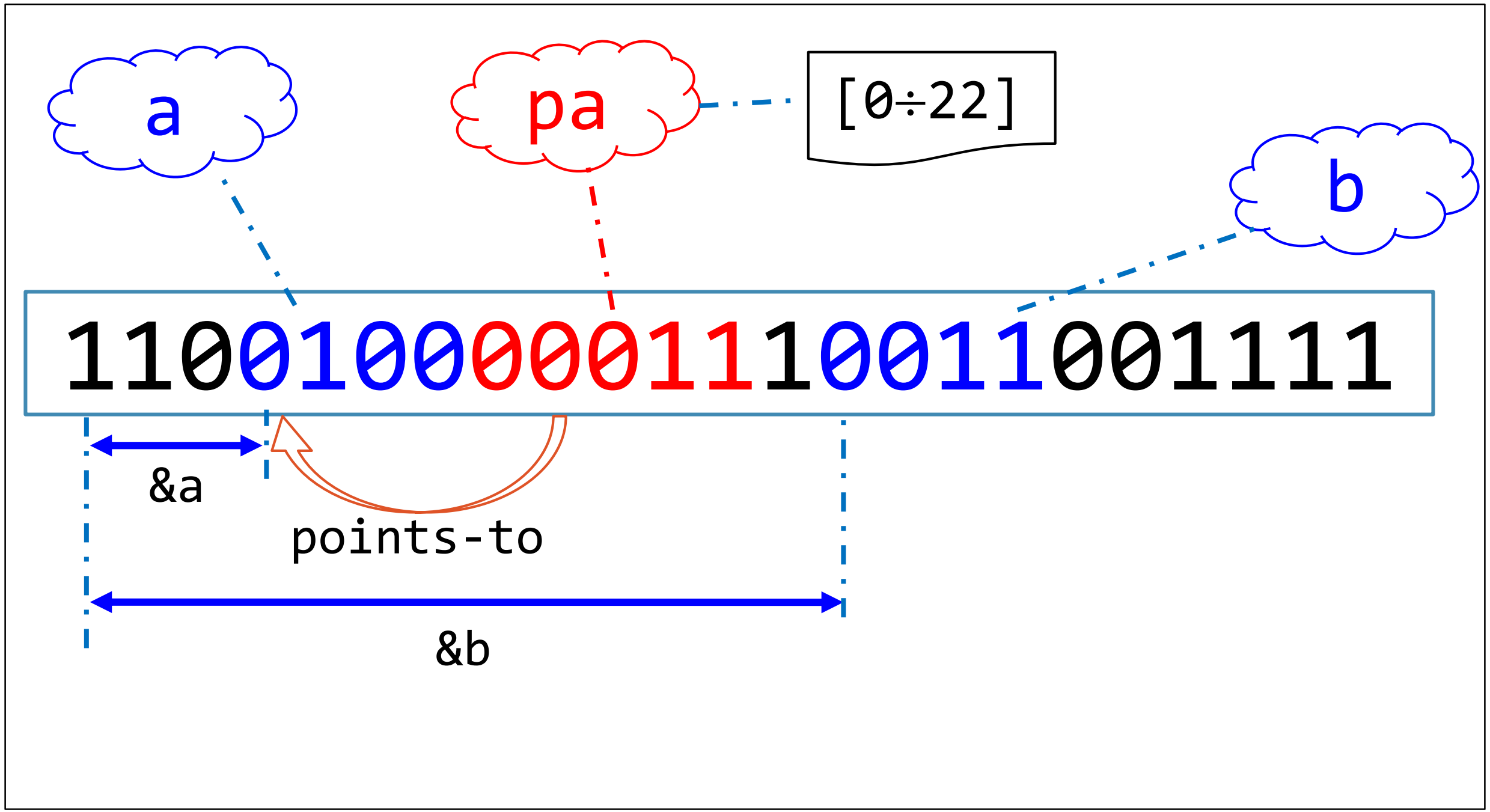

a
4
[-7÷8]
0100

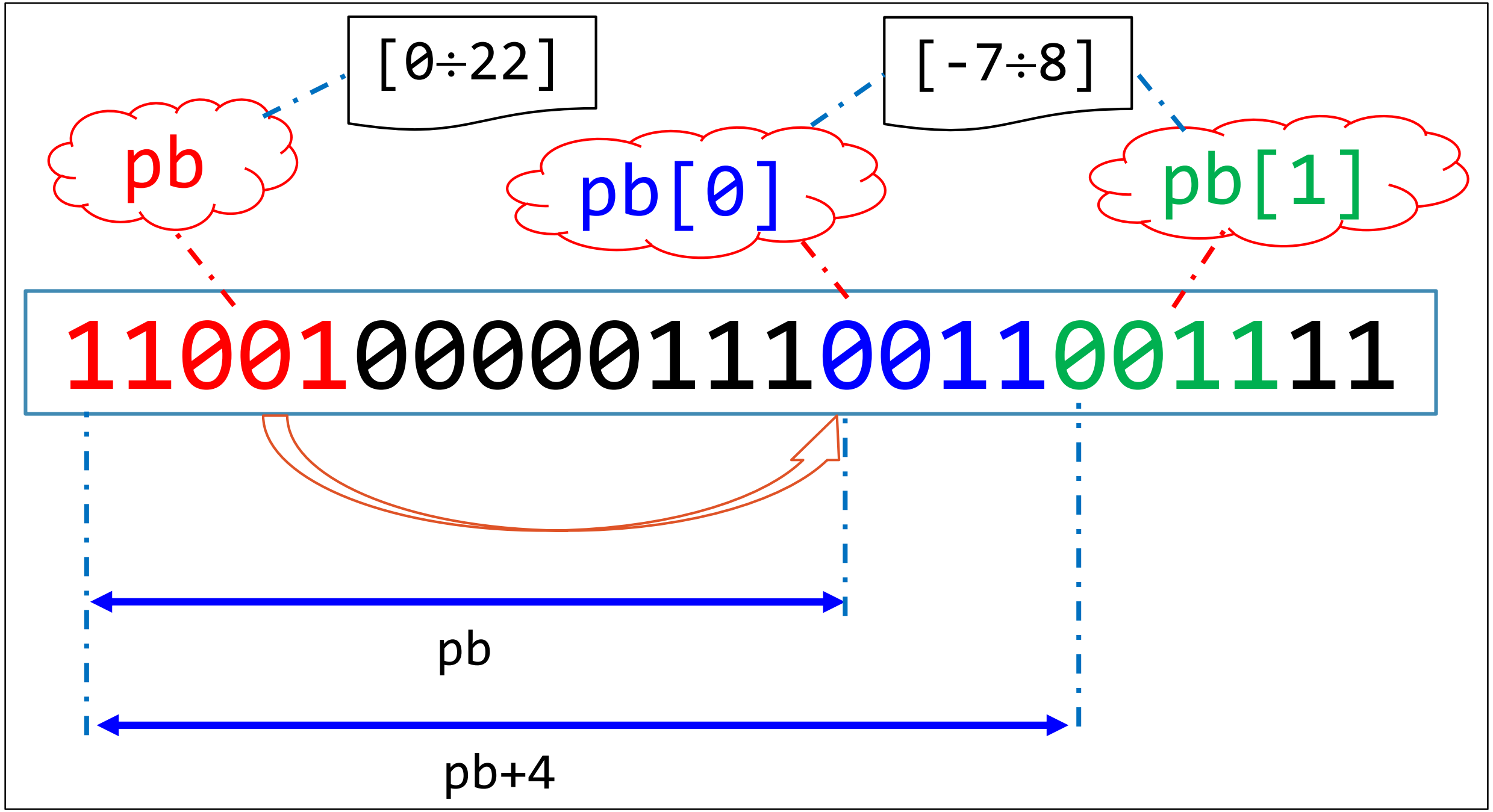

a
4
[-7÷8]
3
b
0100
0011


a
[-7÷8]
4
3
b
0100
0011


a
[-7÷8]
4
3
b
1100100000111001001111


[-7÷8]
a
b
1100100000111100111001111
&a
&b


a
pa
[0÷22]
b
1100100000111001101111
&a
points-to
&b


[0÷22]
[-7÷8]
pb
pb[0]
pb[1]
1100100000011100110001111
pb
pb+4

Минимум про указатели

- На указатель можно взять указатель

```
int *px = &x; int **ppx = &px; int ***pppx = &ppx;
```

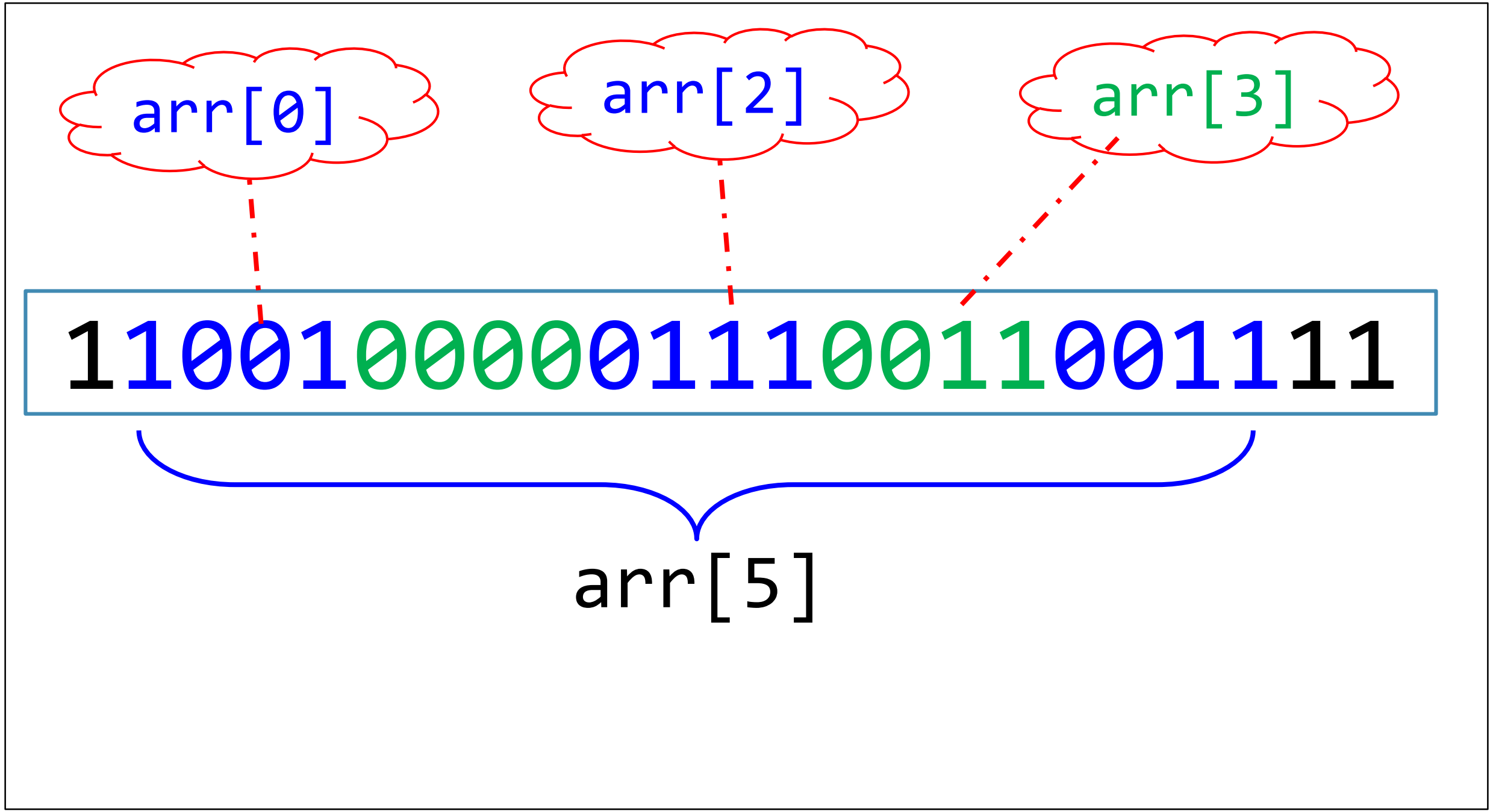
- К указателям можно прибавлять и вычитать целые числа

```
px = py + 2; px -= 3;
```

- Разыменование указателя мало чем отличается от его индексации

```
assert (*x == x[0]); assert(*(x+i) == x[i]);
```

- Поскольку сложение коммутативно, нет разницы между `x[2]`и `2[x]` просто второй способ записи никто не использует
- Минимальная адресуемая ячейка в C это char


arr[0]
arr[2]
arr[3]
1100100000011100110011111
arr[5]

Минимум про массивы

- Название массива можно употреблять как адрес его первого элемента

```
int arr[5]; assert(arr == &arr[0]);
```

- К массиву нельзя ничего прибавить или присвоить, только к элементам

```
arr += 3; // ошибка, но 'arr[1] += 3' ok.
```

- Указатель можно использовать для доступа в массиве

```
int *p = arr; p += 3; p -= 1; assert(*p == arr[2]);
```

- Выход за границы массива это серьёзная ошибка

```
p += 10; printf("%d\n", *p); // печатает что угодно
```