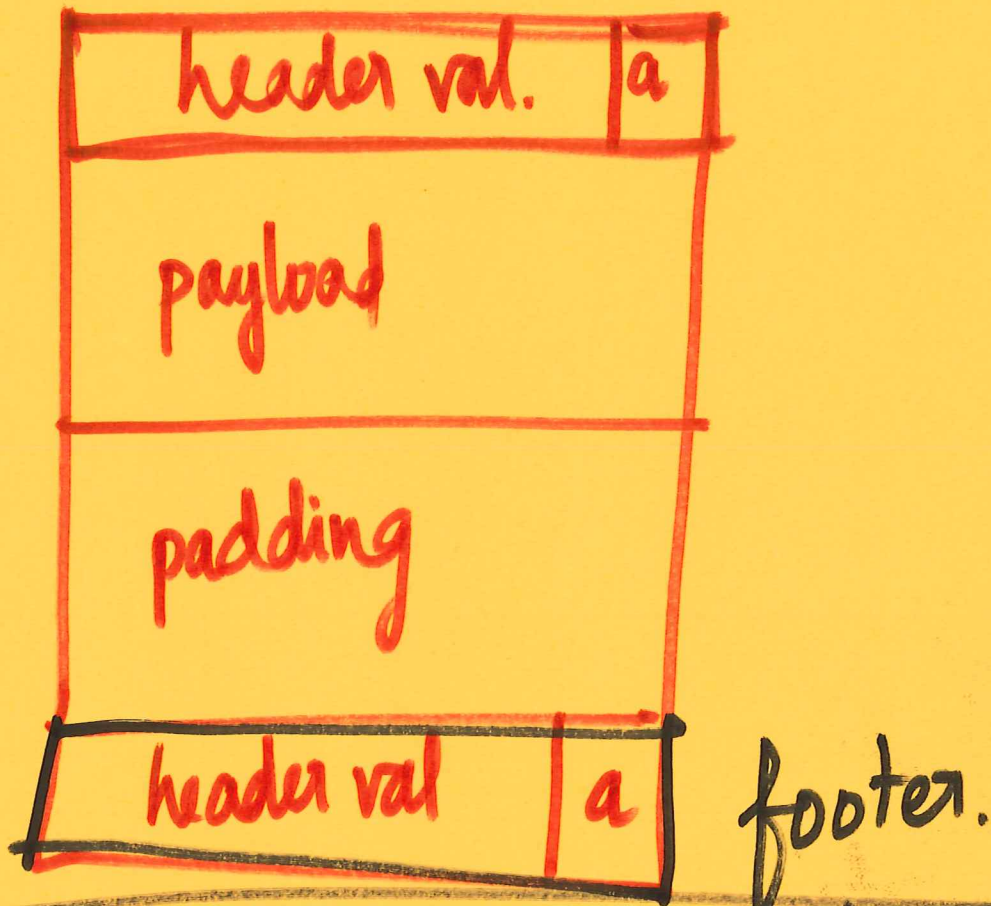


Footer



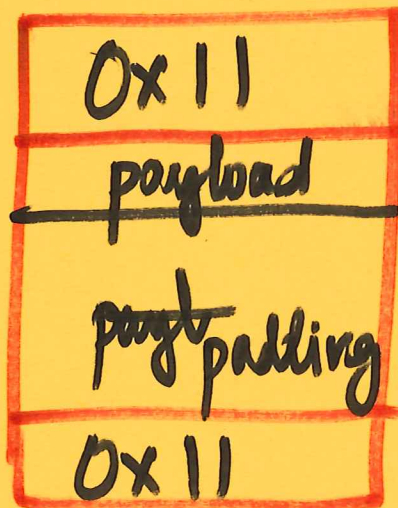
min block size = 16 bytes

malloc (1)

$16_{10} \rightarrow 0x10$

alloc bit =  $\frac{0x01}{0x11}$

bitwise  
OR  
f



4 bytes

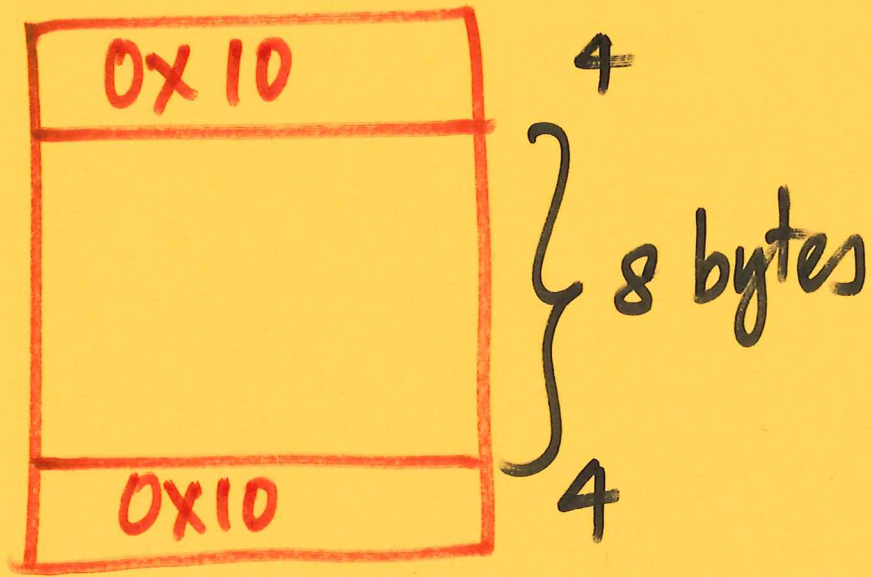
1 bytes

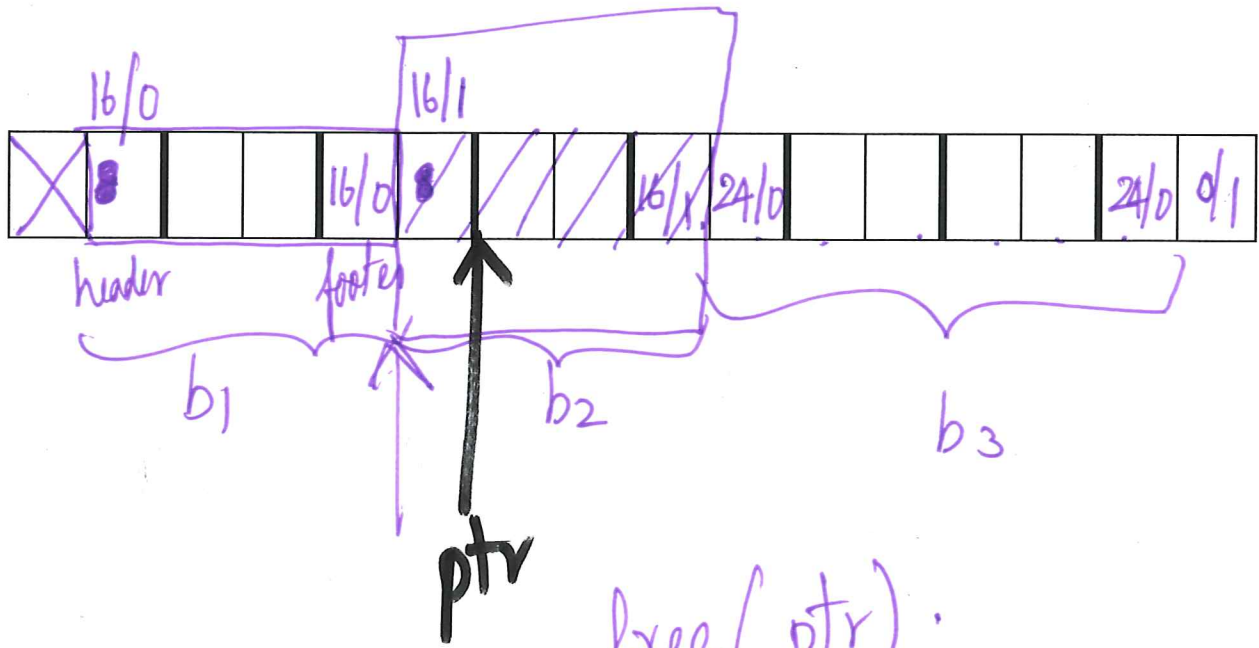
7 bytes

4 bytes

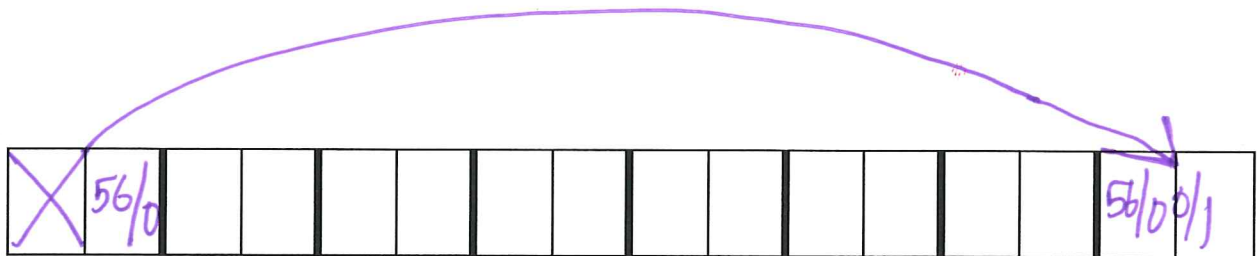
16 bytes

free block of size 16 bytes





free(ptr);

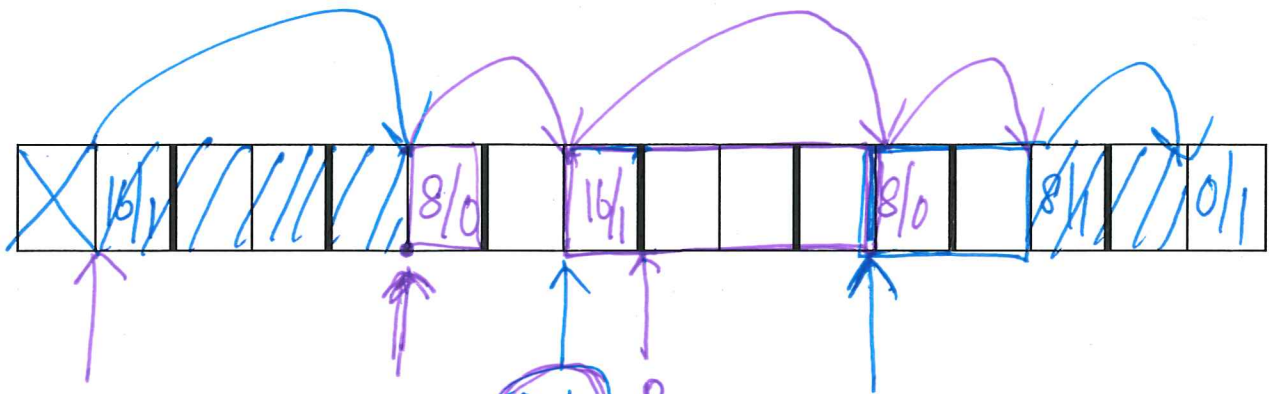


$b_1 \rightarrow 16$

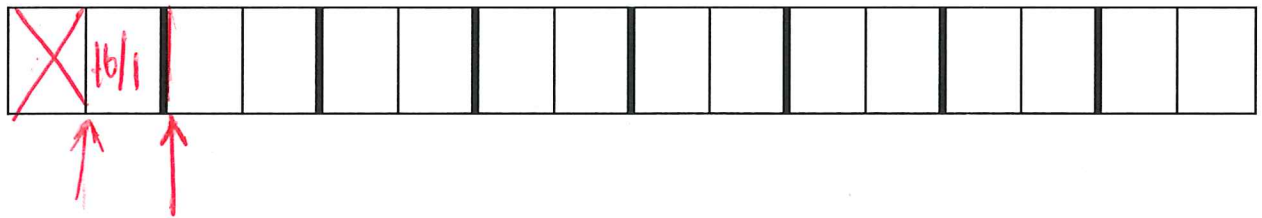
$b_2 \rightarrow 16$

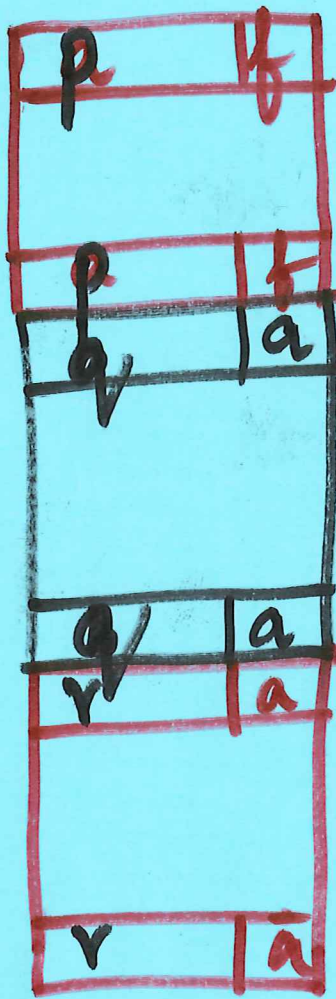
$b_3 \rightarrow 24$

56



$p-4$   $p$   
`free(p);`

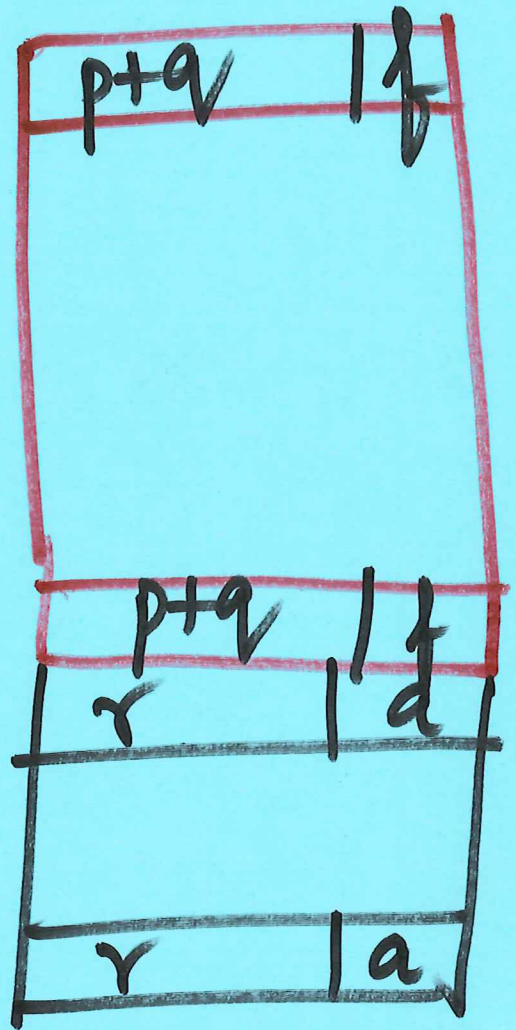




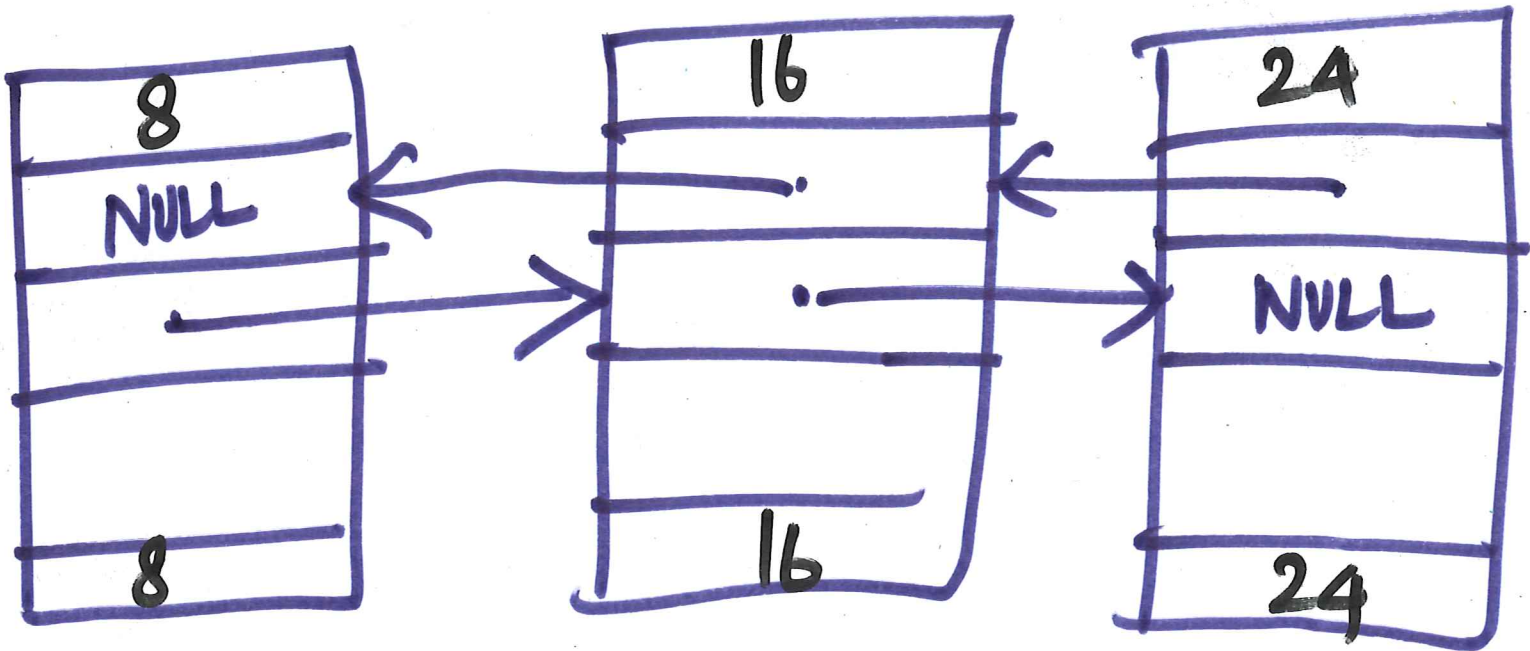
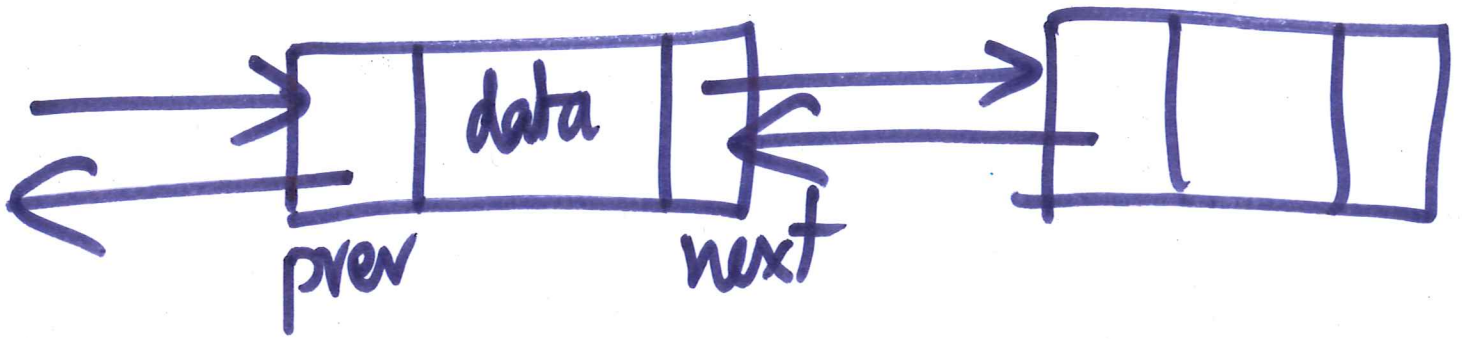
prev block

← p  
free(p);

next block.



# Doubly linked list



3 free blocks

# Allocator design in Ps

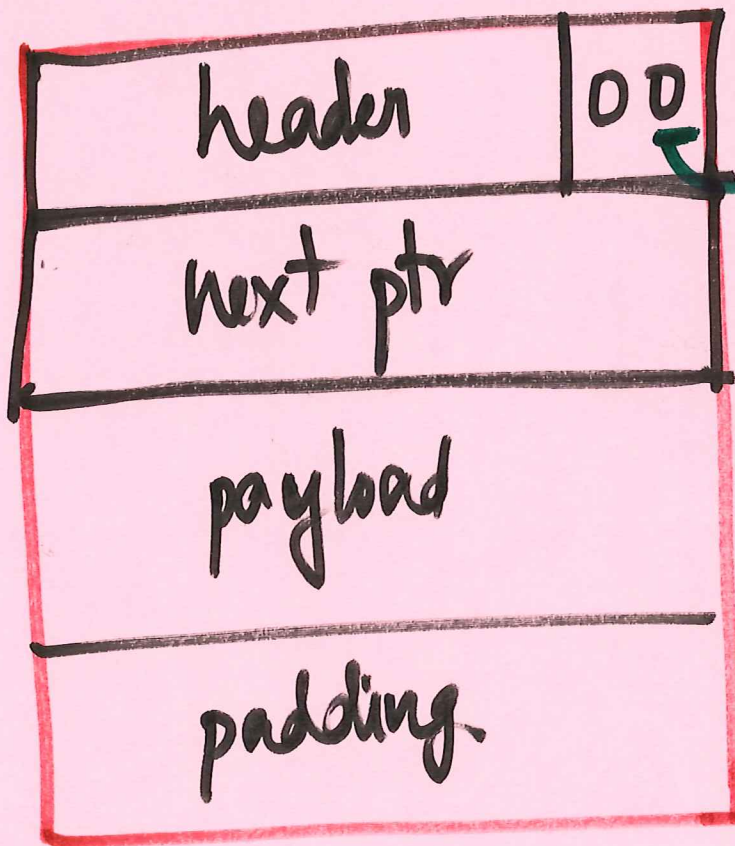
single word aligned (X)

block sizes → 4, 8, 12, ...

0x04    0x08    0x0C

0000 0100    0000 1000    0000 1100

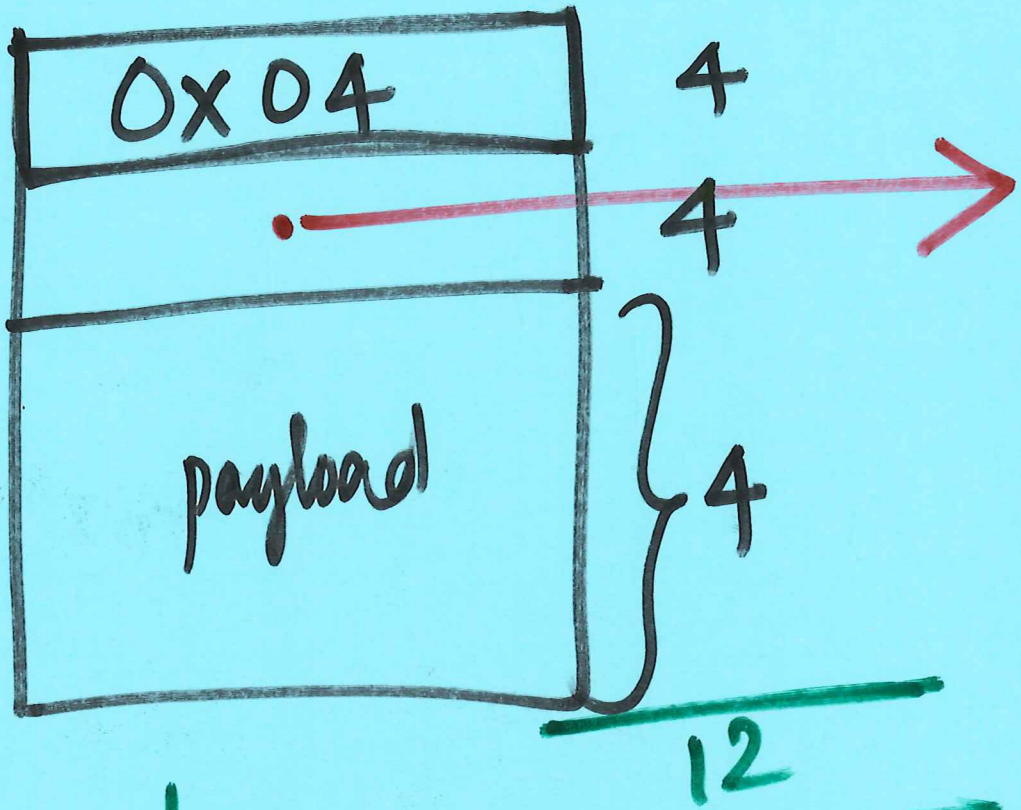
Heap block



→ allocated bit

Min block size = 12 bytes

Free  
block

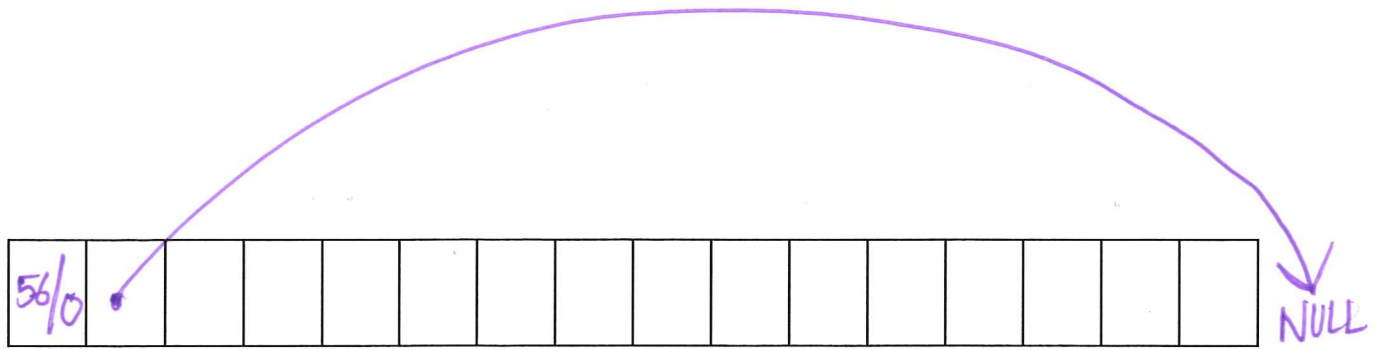


Header value

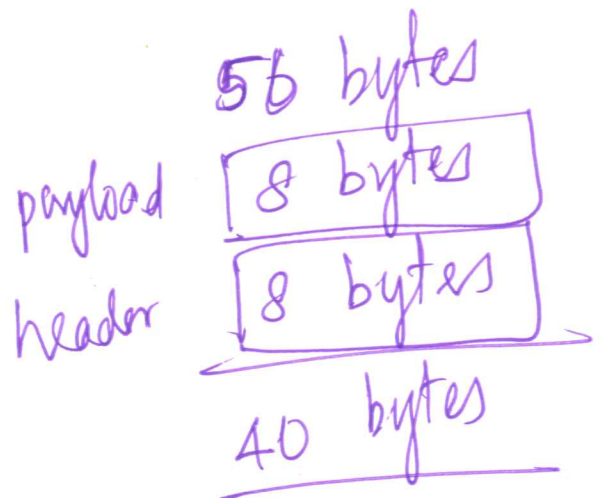
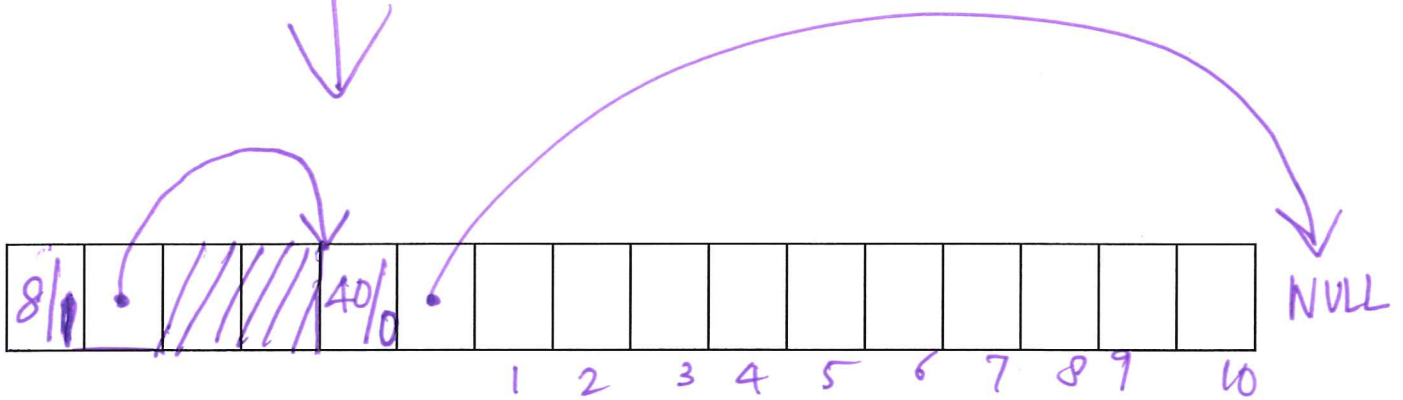
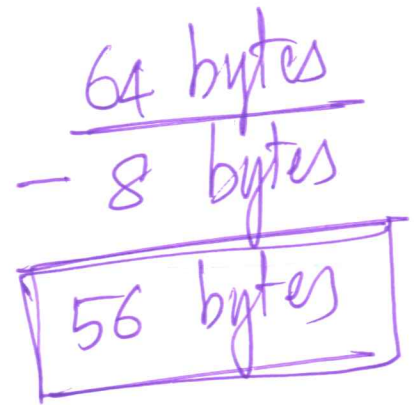
= size of (payload + padding)

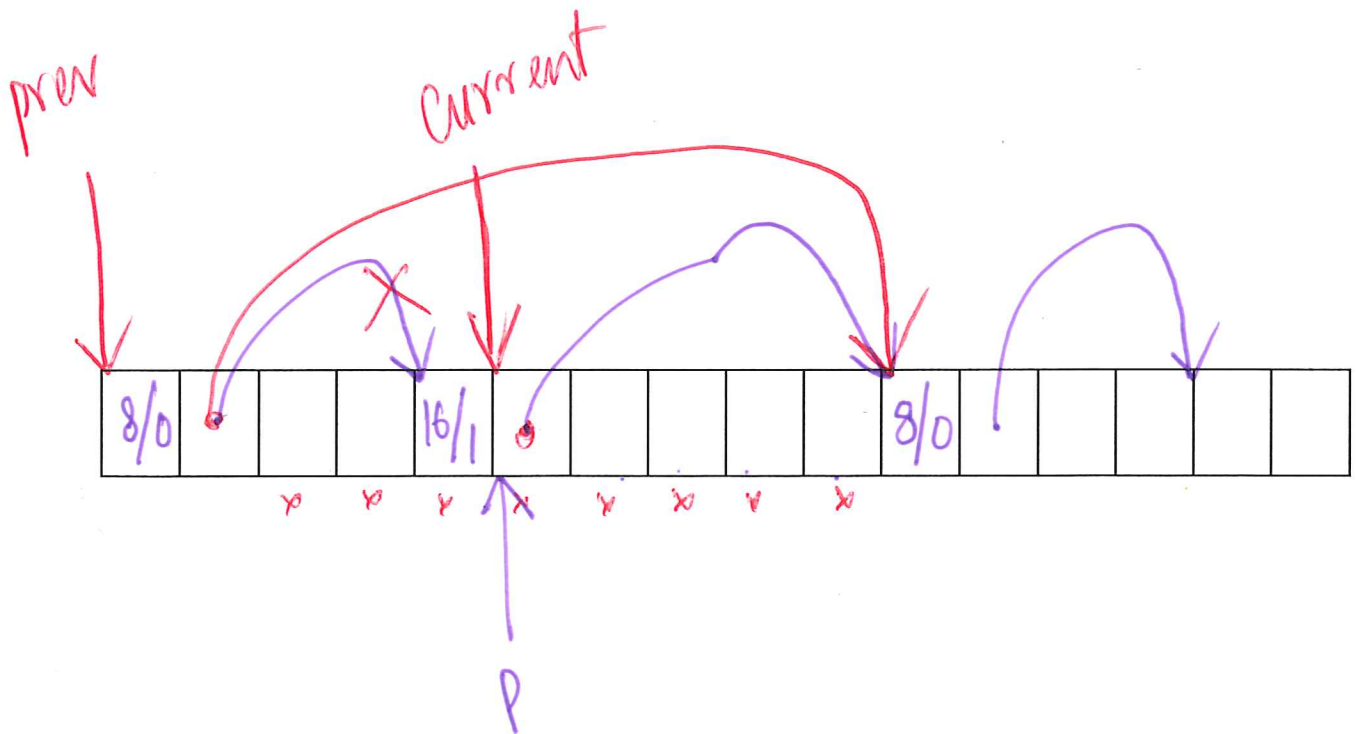
bit-OR  
allocated-bit.





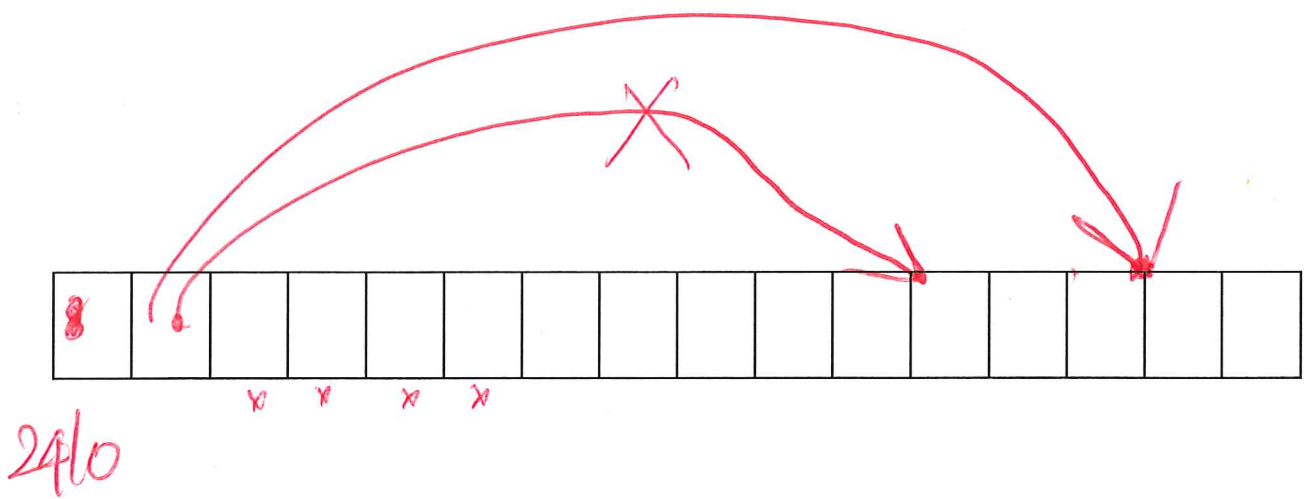
malloc(8)





$$prev \rightarrow next = \text{free}(p)$$

$$current \rightarrow next;$$



8 words  
 $\times 4$   


---

 32  


---