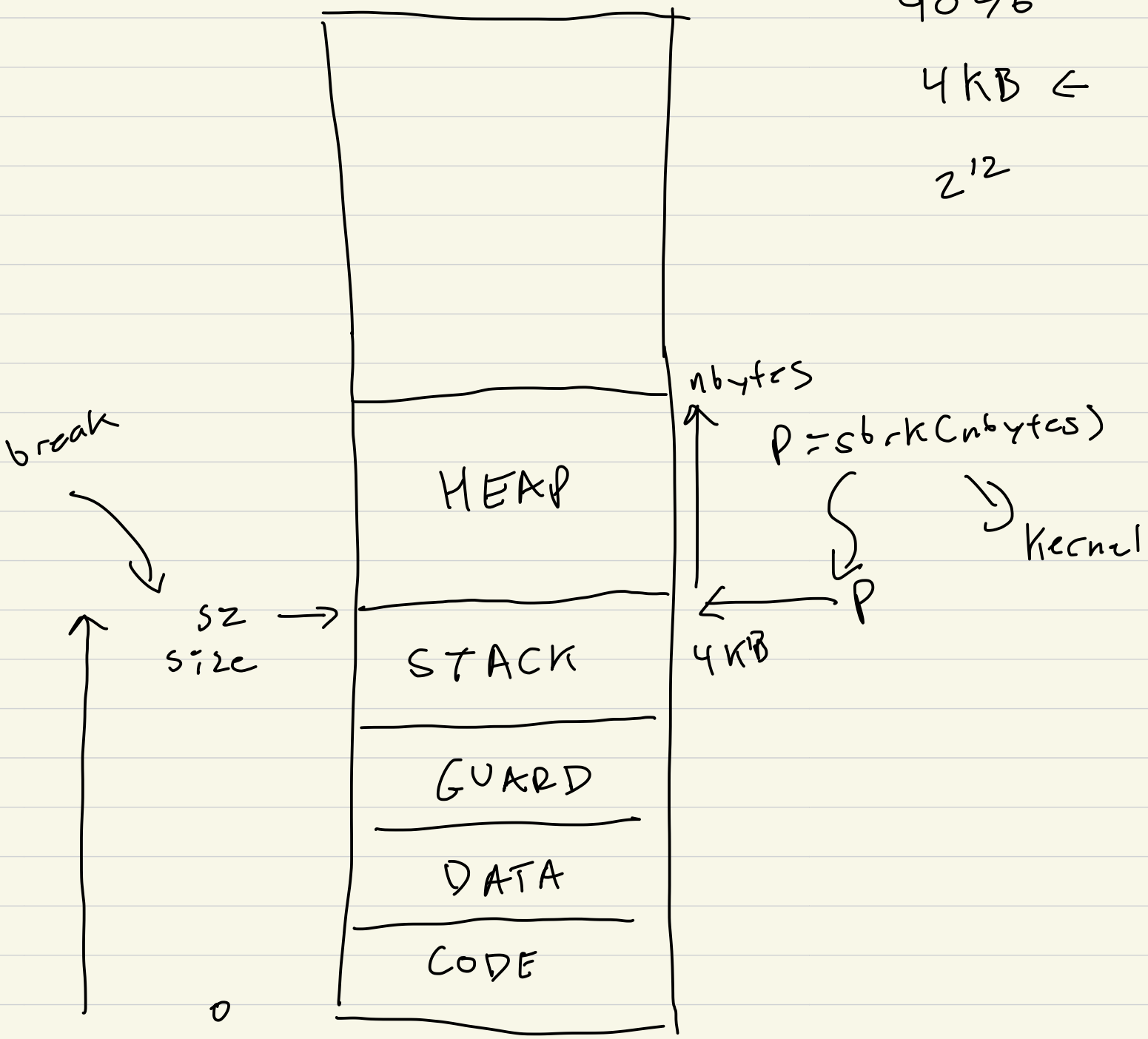
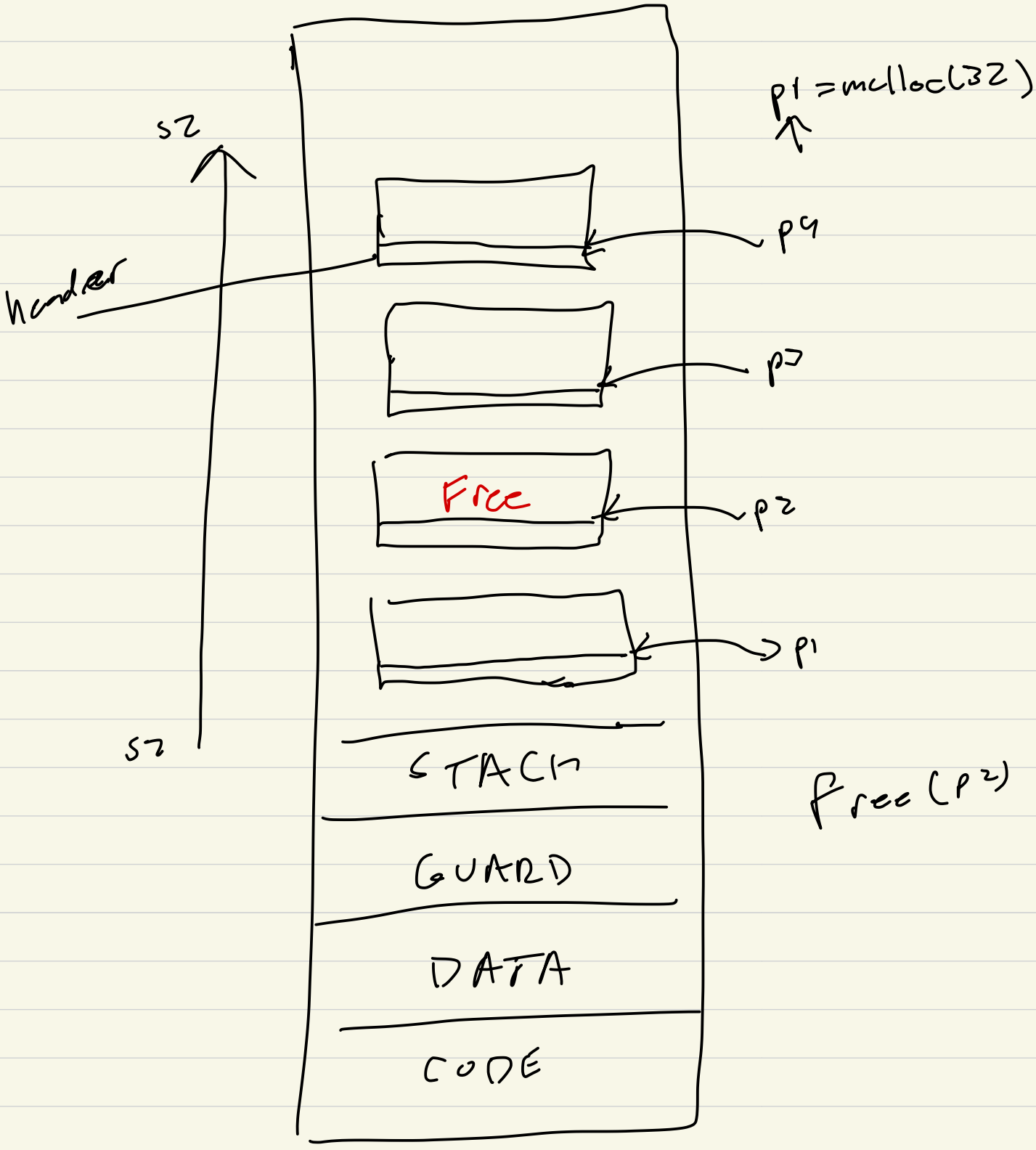
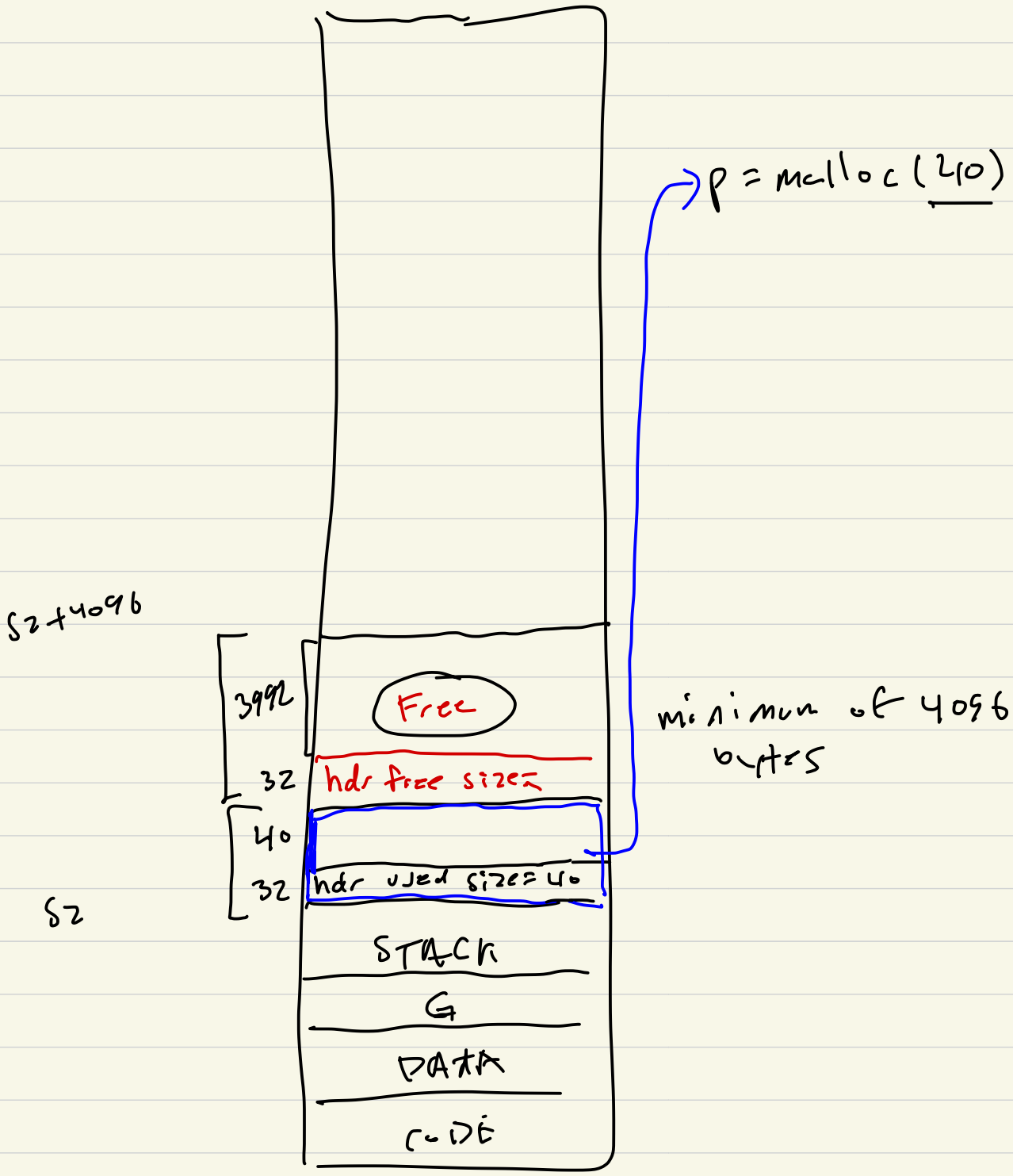
CS 326-01 Heap Allocation

page size
4096
4KB ←
2^{12}




p1 = malloc(32)
sz
header
p4
p3
Free
p2
p1
sz
STACK
GUARD
DATA
CODE
free(p2)



(32+40) + 32 = 104

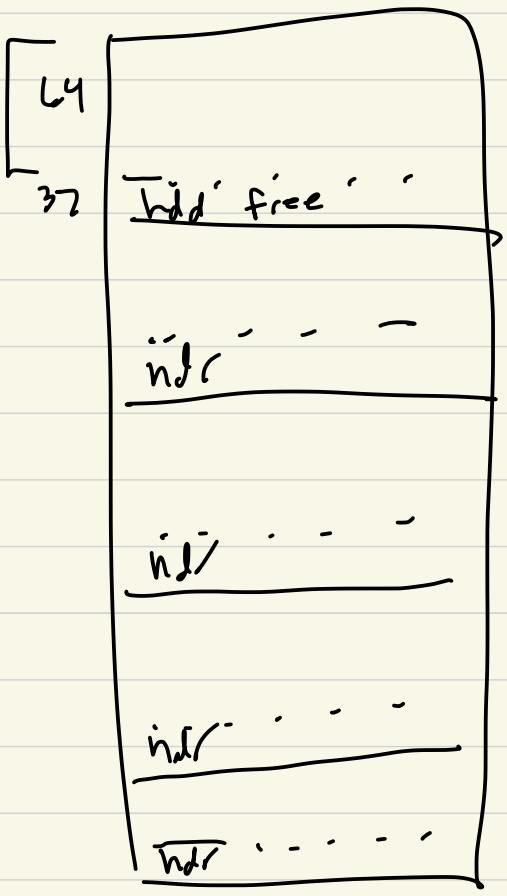
4096 - 104 = 3992

```
struct block_hdr {
(16) struct list_elem elem;
 (8) char name[8];
 (4) int used;
 (4) int size;
 }
```

32

how bis in memory

SZ+4096



SZ

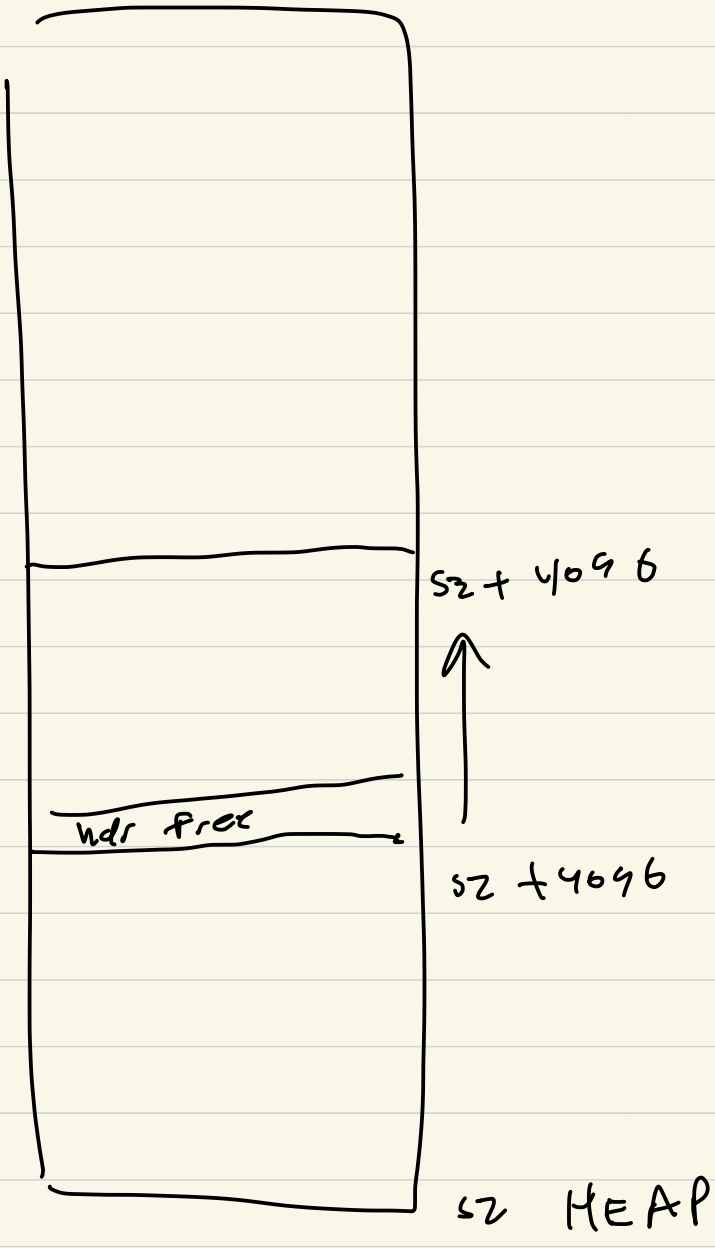
p = malloc(40)

32+40 = 72

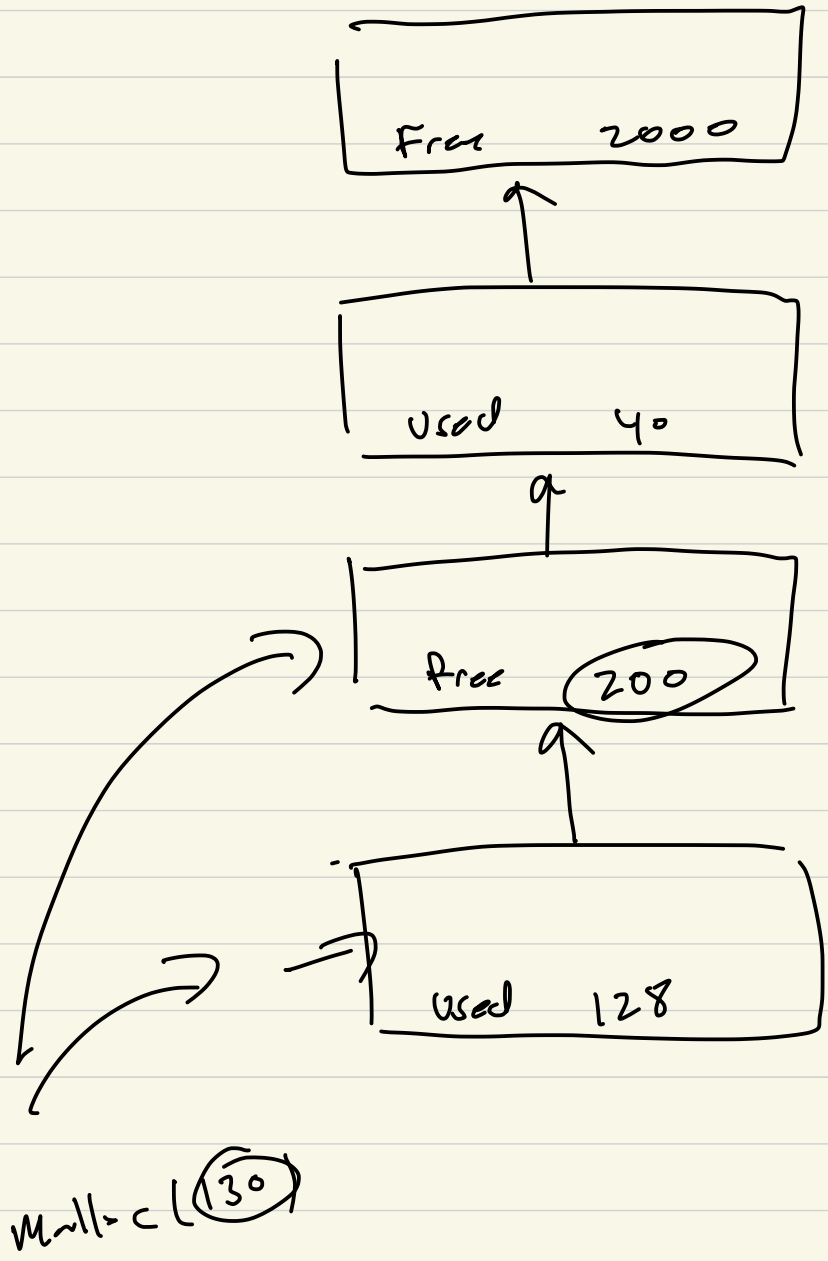
64 - 40 = 24

```
malloc()
{
  if (!heap_initialize)
    need_init()
  }
}
```

4096 - 32 = 4064

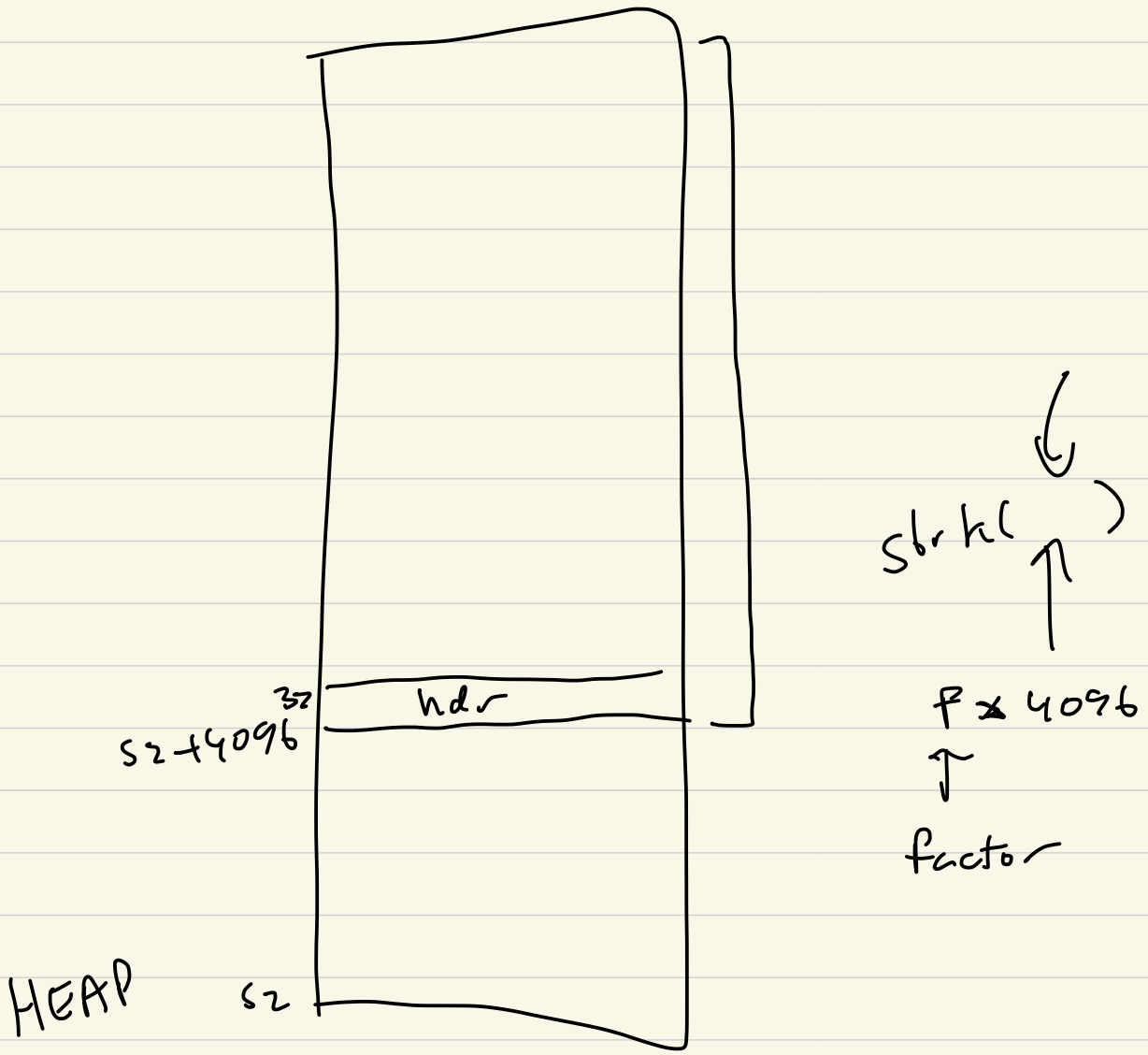

sz + 4096
hdr prev
sz + 4096
sz HEAP

Heap - list



What about

p = malloc(10000)
        n



n = nbytes requested

f = ?

n = nbytes + sizeof(struct block_hdr);

f = n / 4096

= 10,000 / 4096

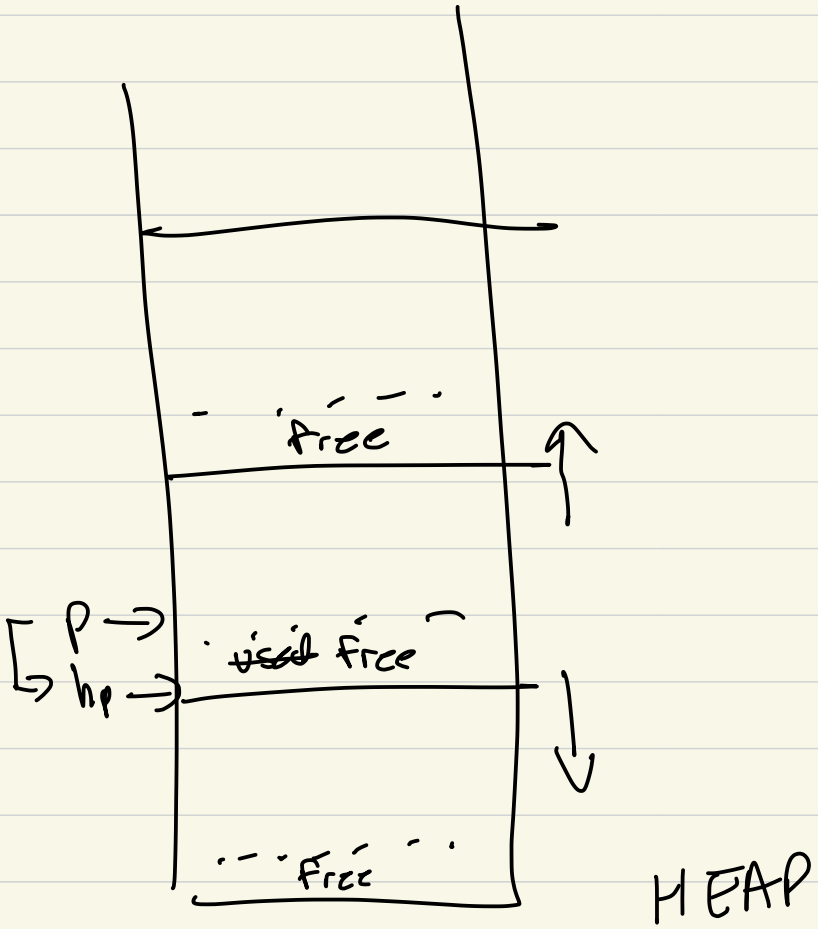
= 2

r = n % 4096

if (r > 0)
    f += 1;

free()

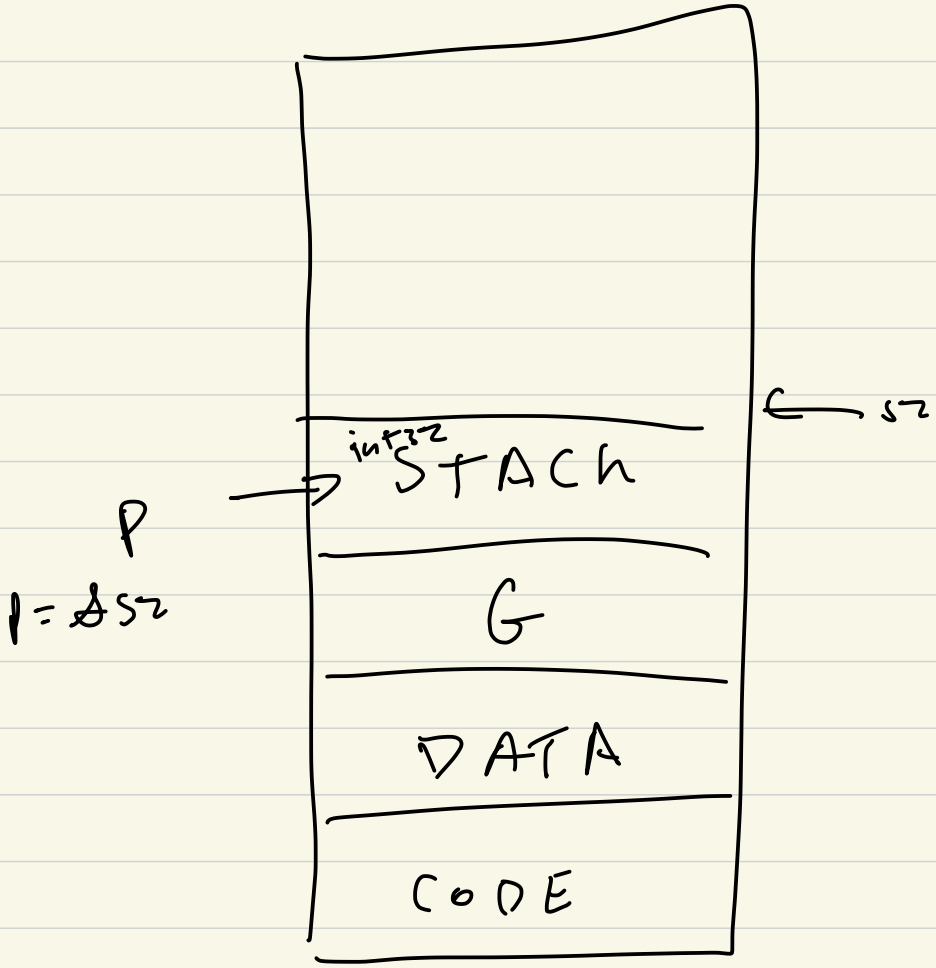
free(p)



struct block_hdr *hp;

hp = (struct block_hdr *) (p - sizeof(struct block_hdr));

hp -> used = 0;


int32
STACK
G
DATA
CODE
sz
P
P = &sz