DMA

Ganesh Kumar . April 8th, 2016

Assume the following,

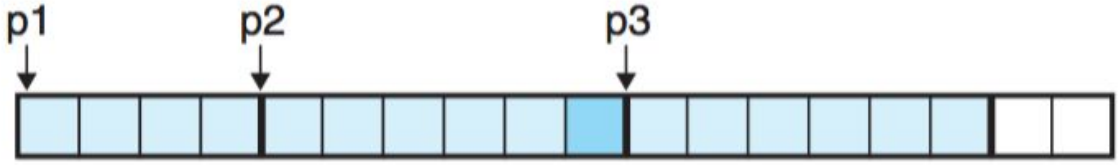
- Initially, the heap has a capacity of 16 words.

0	4	8	12	16	20	24									

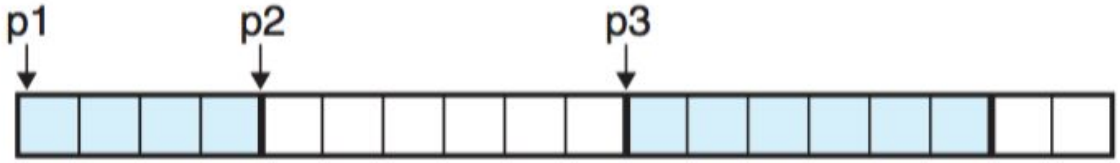
- 1 word = 4 bytes.
- Heap is double-word aligned

(a) `p1 = malloc(4*sizeof(int))`

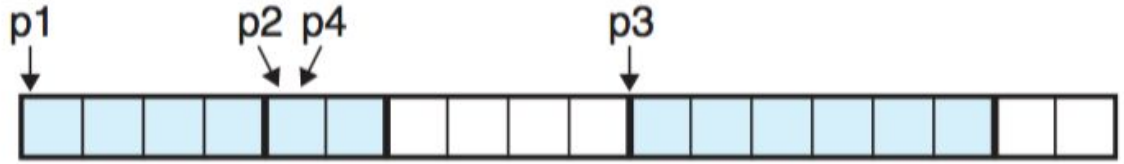
(b) `p2 = malloc(5*sizeof(int))`



(c) `p3 = malloc(6*sizeof(int))`



(d) `free(p2)`



(e) `p4 = malloc(2*sizeof(int))`

4/8.

Last Class

① Virtual Memory / Addressing → Memory Protection

→ More memory that what is physically available.

9.9

② Dynamic Memory Allocation

↳ Heap (Text, Data, Stack).

↳ malloc & free

→ Explicit

→ Implicit.

realloc

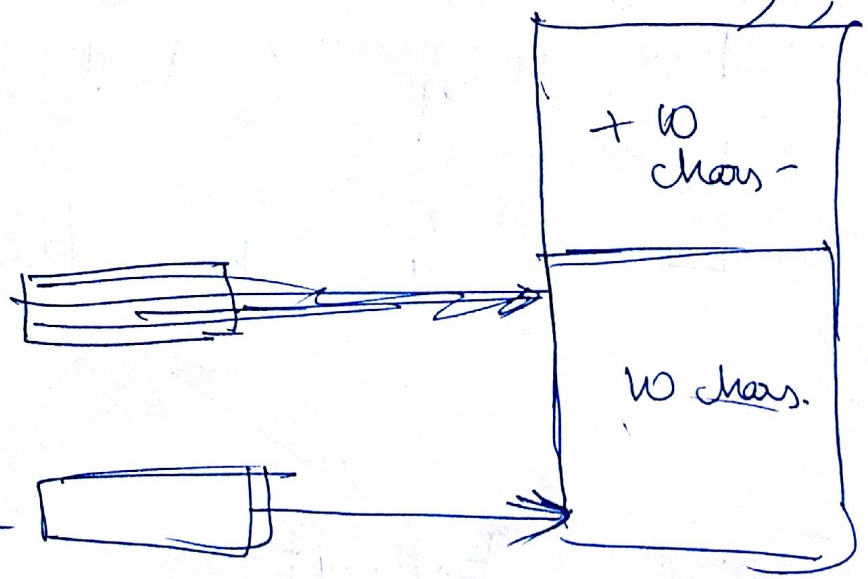
void * realloc (void* ptr, size_t size)

↓

unsigned int.

char * a = (char*) malloc (sizeof(char) * 10);

a = (char *) realloc (a, sizeof(char)* 20);



return null (if there is not enough space)

---

malloc → we do not know what values will be there in the newly allocated block.

calloc.

(void *) calloc (size_t num, size_t size);

returns a pointer to zero-initialized memory block. (size of block = num * size) bytes.

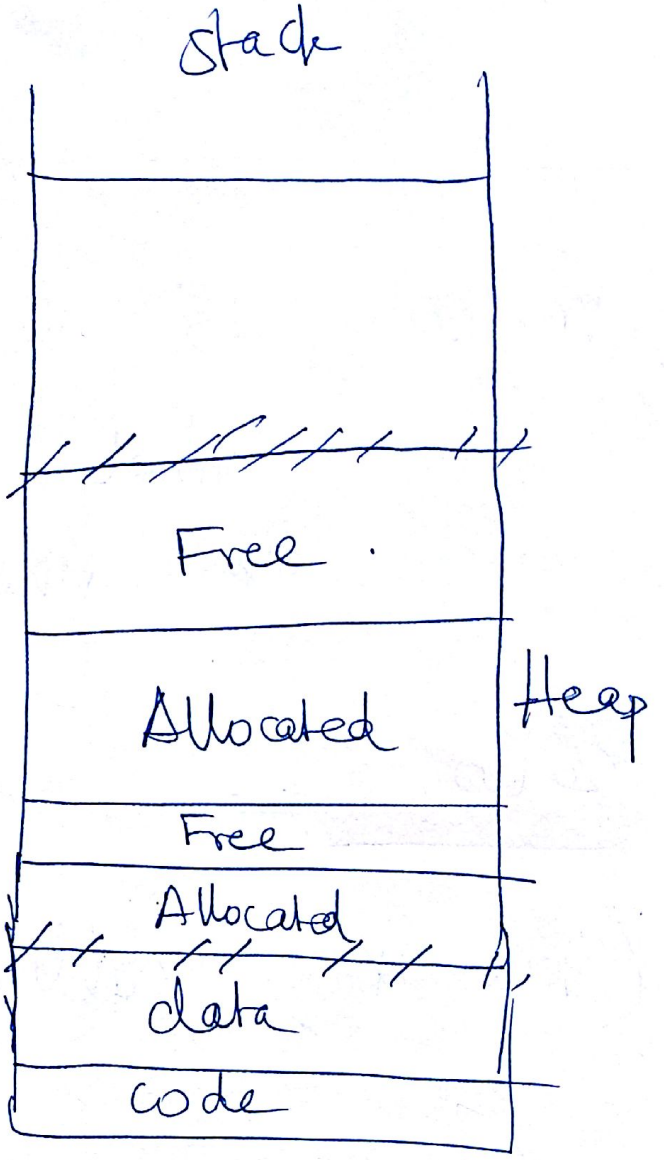
Malloc and Free

malloc.

What if there is no space on the heap?

malloc → call the sbrk() system call to get more memory for the heap

→ a request to the OS.

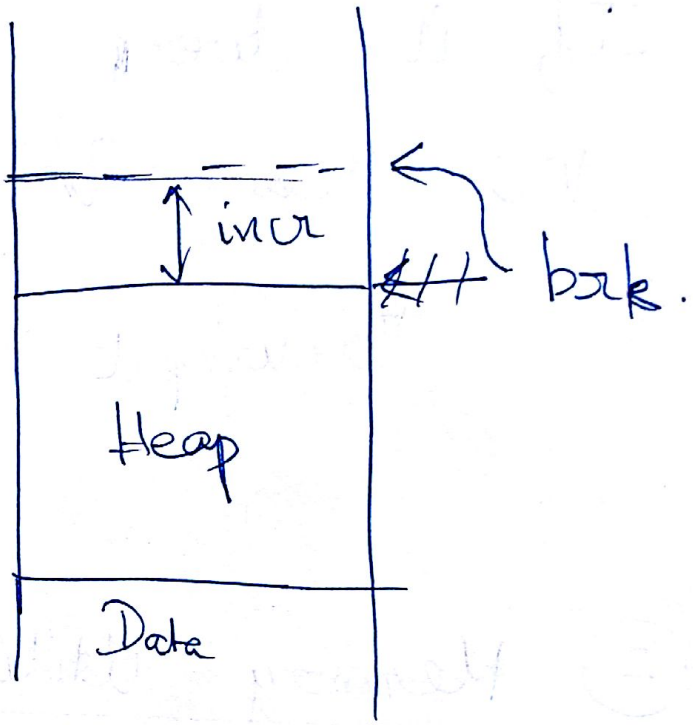


sbrk()

```
void * sbrk (int incr);
```

If everything goes right, sbrk

→ will return the old brk value.



Something went wrong

→ return −1

print → strerror(errno);

sbrk would set this depending on what went wrong.

Design Goals (Allocator)

① Throughput (↑)

If it does 100 mallocs and 100 frees in a second,

Throughput = 200 operations per second.

② Memory Utilization.

VM is limited. ↘ Allocated Memory / Total Heap Size.

Tradeoff → Throughput & Memory Utilization //-

Fragmentation

Poor heap utilization → because of → Fragmentation.

Internal Fragmentation

↳ When an allocated block is larger than the payload.

When could this happen ?

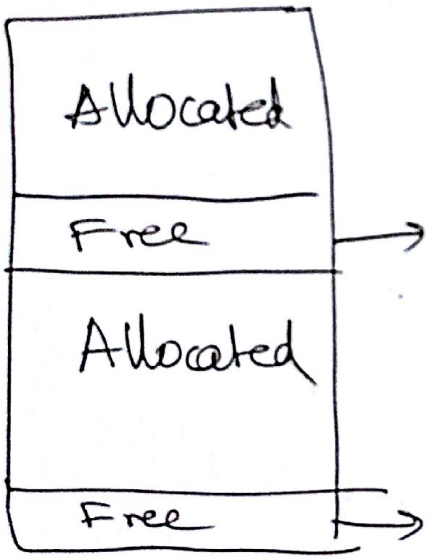
→ Larger block size due to some alignment restriction ↘ (payload).

→ Minimum block size restriction

How much space is wasted by Internal Fragmentation.

SUM (all allocated blocks) − SUM (size of all payloads)

External Fragmentation



(1) Requesting 3 words.

(2) Even though the heap has 4 words of free space, it cannot satisfy the request.

Typically → to avoid external fragmentation.

Small # of large free blocks (over) Large # of small free blocks.

Design Considerations

(1) Free block organization
   - Lists ?
   - Pointers ?

(2) Placement.

How do we choose a block ?

③ Splitting.

After we allocate a part of a free block, what do we do with the rest?

④ Coalescing

What do we do with the block that was just freed?