CSE 29 Week 9 Discussion

Exam 3 Review

Types of Memory Allocators

- First Fit (fast but inefficient)
- Best Fit (slow but efficient)
- Bump Allocator (very fast but not able to free individual allocs)
- Slot Allocator (fast but can potentially have more fragmentation)

- There are many others!

Exercise 1

Suppose you have a heap that has the following rules:

- Uses a first fit allocator for malloc()
- Header is 2 bytes
  - LSB represents if the next block is free (1) or not (0)
  - Bits[15:1] represent the size of the block (including header) as an unsigned int
- Assume blocks are 2-byte aligned



Exercise 1

Suppose you have a heap that has the following rules:

- Uses a first fit allocator for malloc()
- Header is 2 bytes
  - LSB represents if the next block is free (1) or not (0)
  - Bits[15:1] represent the size of the block (including header) as an unsigned int
- Assume blocks are 2-byte aligned

What is the state of the heap after the following operations?

```
x=malloc(20)
y=malloc(40)
z=malloc(30)
free(y)
a=malloc(20)
```

Will the state be different for a best fit allocator? Why or why not?



Solution

Exercise 2

Suppose you have a heap that has the following rules:

- Uses a slot allocator
- Heap is divided into 32KB slots. Total heap size = 256KB
- We have a free list that keeps track of the indices (zero-based index) of all free slots
- Blocks are allocated to the first available contiguous set of free slots

Exercise 2

Suppose you have a heap that has the following rules:

- Uses a slot allocator
- Heap is divided into 32KB slots. Total heap size = 256KB
- We have a free list that keeps track of the indices of all free slots
- Blocks are allocated to the first available contiguous set of free slots

After the following operations, the free list contains the slot indices [1, 5, 6, 7]. Is this correct? Why or why not?

```
a=malloc(31KB)
b=malloc(5KB)
c=malloc(27KB)
free(b)
d=malloc(34KB)
```

Solution

Exercise 3

Suppose you have a heap that has the following rules:

- Uses a first-fit allocator with immediate coalescing
- Has a 4-byte header and 4-byte footer
- Blocks are 8-byte aligned

What is the state of the heap after the following operations?

```
a = malloc(16)
b = malloc(20)
c = malloc(16)
d = malloc(8)
free(a)
free(c)
free(b)
e = malloc(48)
```

Exercise 3

Suppose you have a heap that has the following rules:

- Uses a first-fit allocator with immediate coalescing
- Has a 4-byte header and 4-byte footer
- Blocks are 8-byte aligned

What is the state of the heap after the following operations?

```
a = malloc(16)
b = malloc(20)
c = malloc(16)
d = malloc(8)
free(a)
free(c)
free(b)
e = malloc(48)
```

Would `e = malloc(48)` have succeeded *without* coalescing?

Why does this allocator require a footer in addition to a header to coalesce `free(b)` in O(1) time?

Solution

Exercise 4

Suppose you have a heap that has the following rules:

- Uses a bump-allocator with a single `cursor` pointer initialized to 0x1000
- The heap is 64 bytes long (i.e. max valid address is 0x103F)
- free() releases all allocated blocks.
- No byte-alignment requirements

The following operations are performed on the heap:

```
a = malloc(5)
b = malloc(16)
free()
c = malloc(12)
d = malloc(24)
e = malloc(20)
```

What address is returned for each allocation (a, b, c, d, e)?

Solution

Questions

- **Exam 3**
- Project 5
- Lecture content: pointers, stack, memory allocation, ...
- Labs: unix commands, git, vim, ...
- etc

Reminders

Please complete the SET course and IA evaluations!

Happy Thanksgiving!