CS 240
Computer
Systems

#7/8: Heap Memory and Heap Management

CS 240 - September 15, 2020
Wade Fagen-Ulmschneider

Heap Memory

There are two primary forms of memory storage for a process:

1.

...how is it allocated in C code?

...how is it stored in memory?

2.

...how is it allocated in C code?

...how is it stored in memory?

Sample Program #1:

Can we see the use of the heap and the stack in a real program?

07-memory2/heapAndStack.c

```
int val;
printf("&val: %p\n", &val);

int *ptr = malloc(sizeof(int));
printf("&ptr: %p\n", &ptr);
printf(" ptr: %p\n", ptr);

int *ptr2 = malloc(sizeof(int));
printf("&ptr2: %p\n", &ptr2);
printf(" ptr2: %p\n", ptr2);

return 0;
```

Page Table:

....

Efficient Use of Heap Memory

During the lifetime of a single process, we will allocate and free memory many times. Consider a simple program:

07-memory2/free.c

```
int *a = malloc(4096);
printf("a = %p\n", a);
free(a);

int *b = malloc(4096);
printf("b = %p\n", b);

int *c = malloc(4096);
printf("c = %p\n", c);

int *d = malloc(4096);
printf("d = %p\n", d);

free(b);
free(c);

int *e = malloc(5000);
printf("e = %p\n", e);

int *g = malloc(10);
printf("g = %p\n", g);

int *g = malloc(10);
printf("g = %p\n", g);
```

Heap:

Heap:

How much memory is used if we **do not** reuse memory?

How much memory is used with **optimal** reuse of memory?

- What happens to our memory over time?
- When we have “holes” in our heap, how do we decide what hole to use?

Heap Management Strategies

There are many strategies on the best way to allocate memory to the heap:

#1: [No Reuse]:

#2: [Free Lists]:

Free List Allocation Strategies:

1.

2.

3.

07-memory2/freeList.c

```
int *ptr[10];
for (int i = 0; i < 10; i++) {
  ptr[i] = malloc(100 * (10 - i));
}

for (int i = 0; i < 10; i += 2) {
  free(ptr[i]);
}

int *a = malloc(300);
int *b = malloc(100);
int *c = malloc(800);
int *d = malloc(800);
```

Allocation with No Reuse:

Allocation with Best Fit:

Idea: Segregated Lists:

Allocation with First Fit:

Idea: Block Splitting

Idea: Coalescing