

Process Layout - Heap

Karthik Dantu

Ethan Blanton

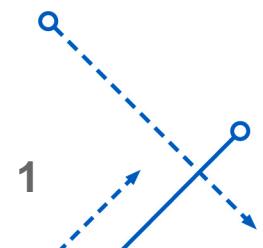
Computer Science and Engineering

University at Buffalo

kdantu@buffalo.edu

Portions of this lecture are from the Princeton COS 217 course slides

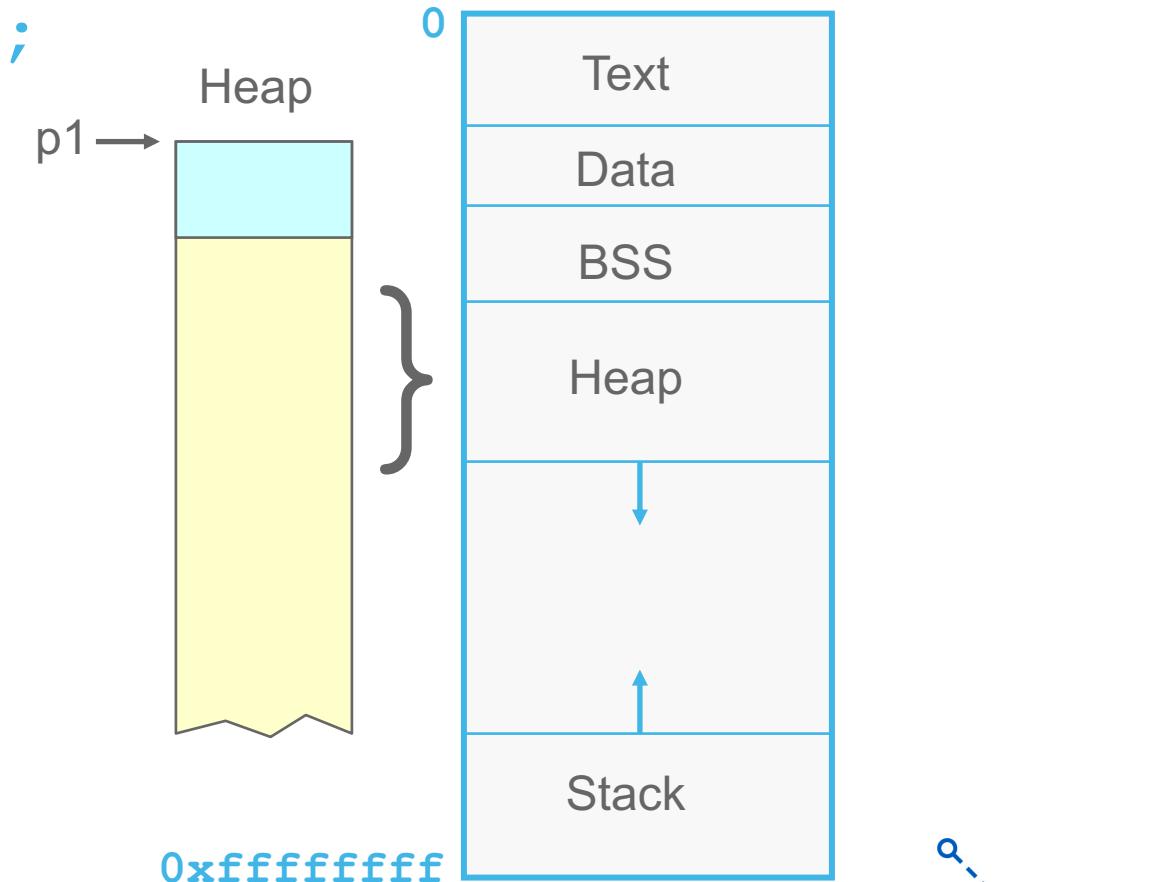
Karthik Dantu



Heap: Dynamic Memory

- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

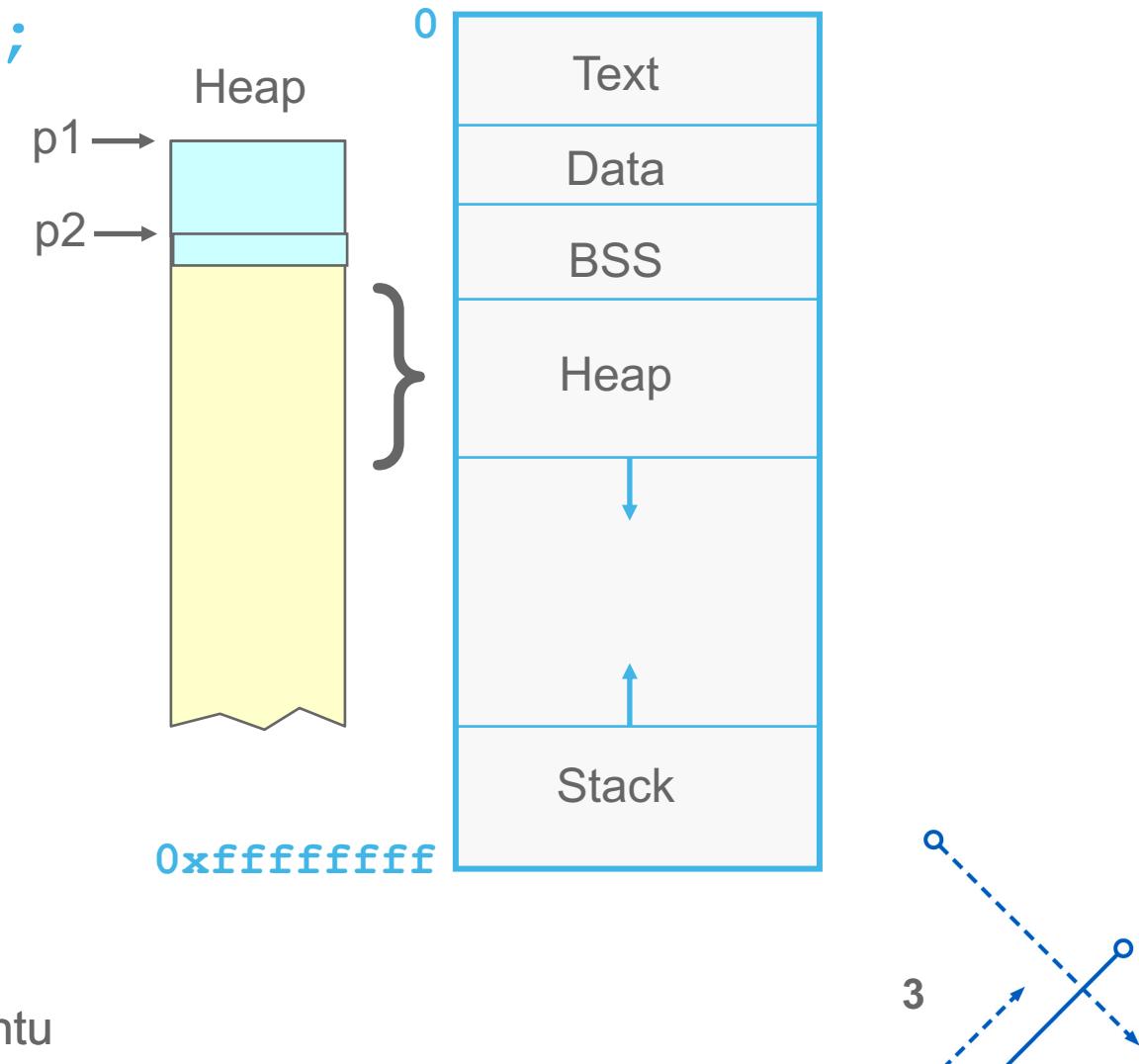
 ➔ char *p1 = malloc(3);
 char *p2 = malloc(1);
 char *p3 = malloc(4);
 free(p2);
 char *p4 = malloc(6);
 free(p3);
 char *p5 = malloc(2);
 free(p1);
 free(p4);
 free(p5);
```



# Heap: Dynamic Memory

- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

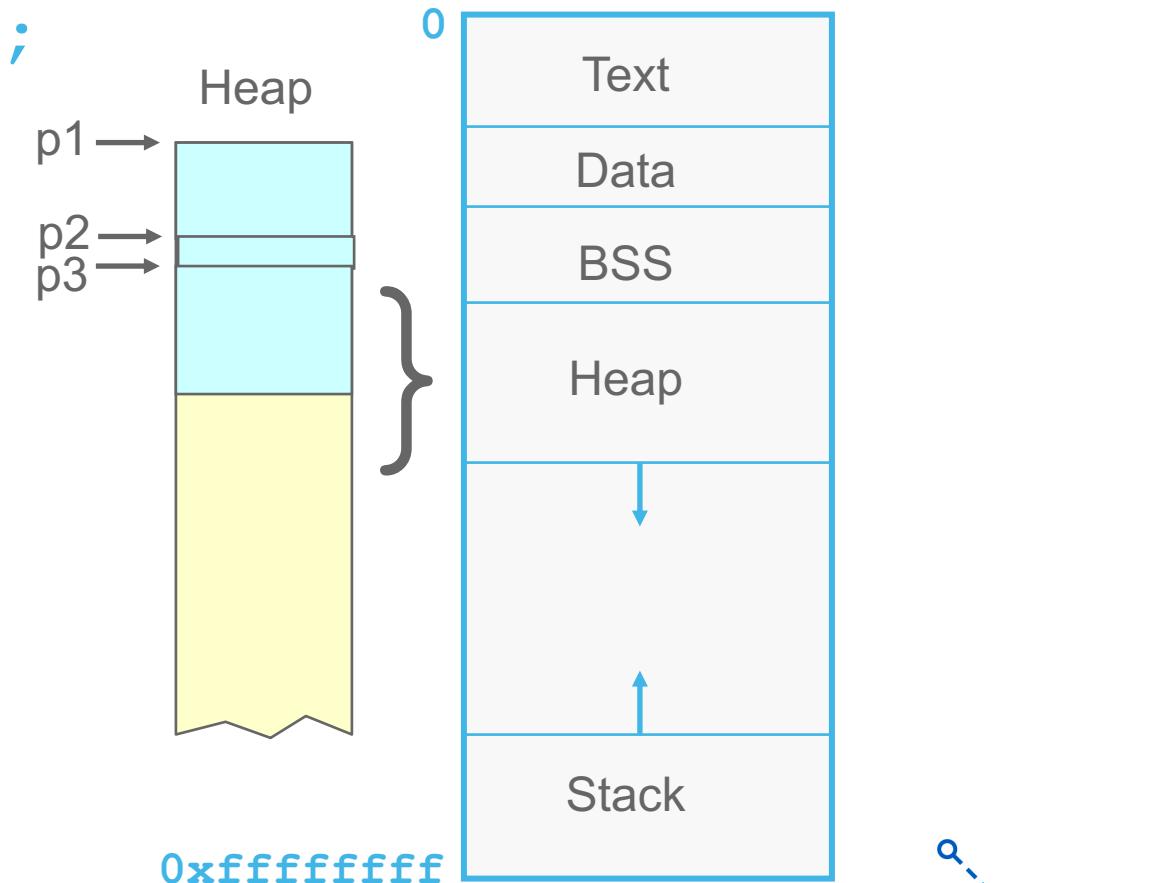
    ➔ char *p1 = malloc(3);
    char *p2 = malloc(1);
    char *p3 = malloc(4);
    free(p2);
    char *p4 = malloc(6);
    free(p3);
    char *p5 = malloc(2);
    free(p1);
    free(p4);
    free(p5);
```



Heap: Dynamic Memory

- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

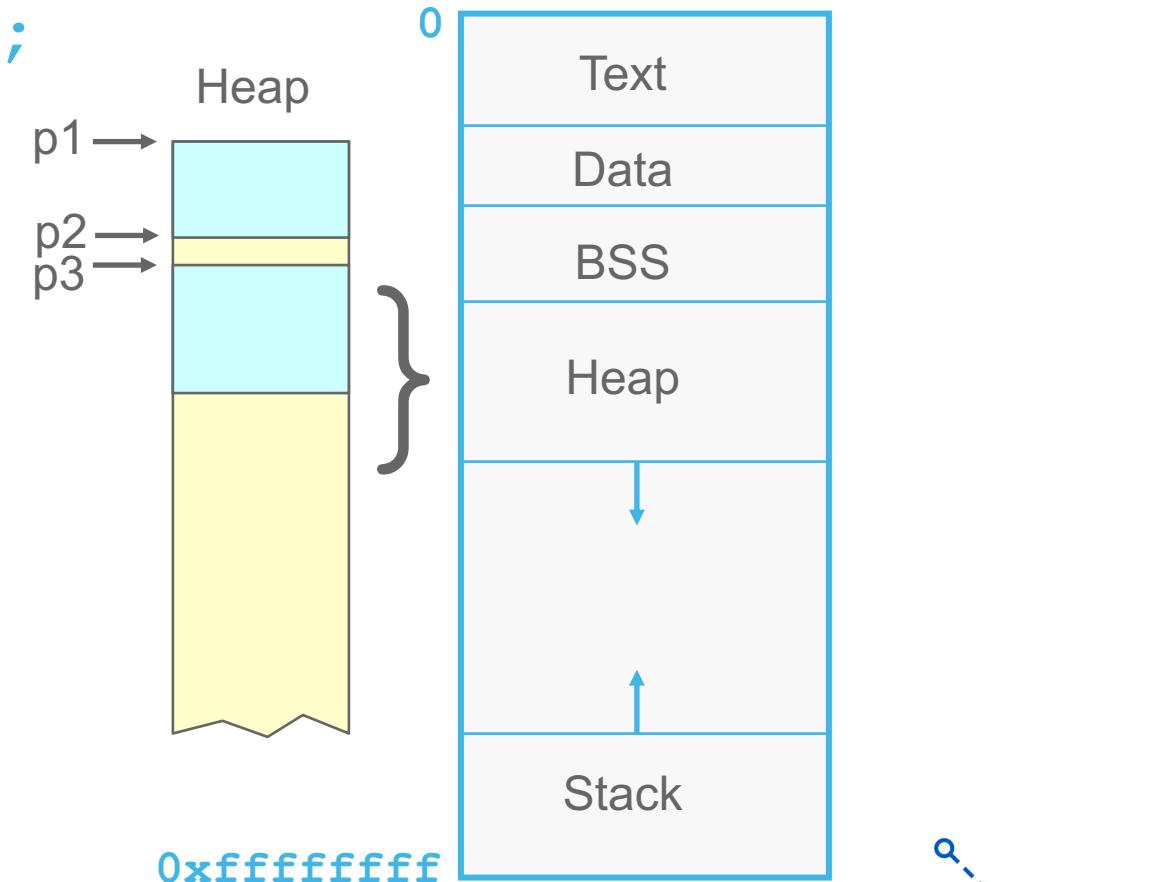
char *p1 = malloc(3);
char *p2 = malloc(1);
char *p3 = malloc(4);
free(p2);
char *p4 = malloc(6);
free(p3);
char *p5 = malloc(2);
free(p1);
free(p4);
free(p5);
```



# Heap: Dynamic Memory

- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

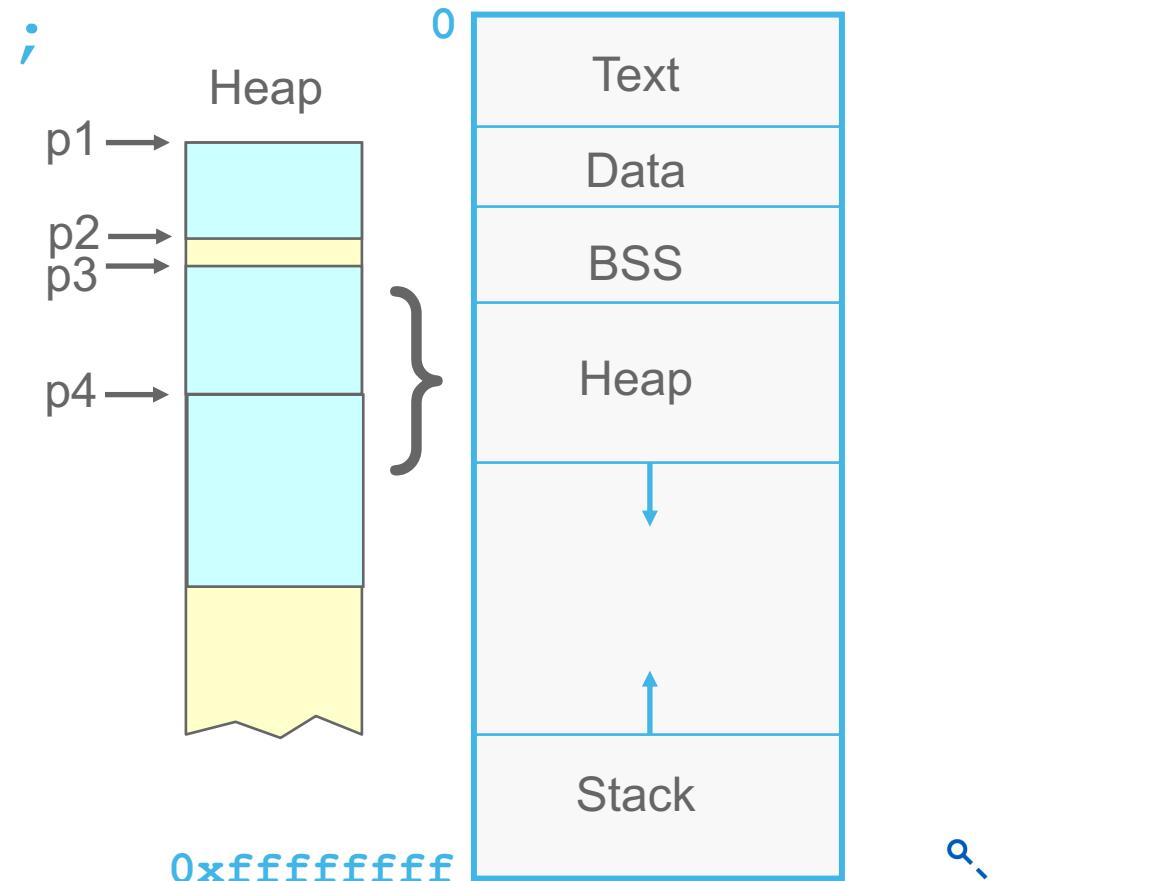
char *p1 = malloc(3);
char *p2 = malloc(1);
char *p3 = malloc(4);
free(p2);
char *p4 = malloc(6);
free(p3);
char *p5 = malloc(2);
free(p1);
free(p4);
free(p5);
```



Heap: Dynamic Memory

- `#include <stdlib.h>`
`void *malloc(size_t size);`
`void free(void *ptr);`

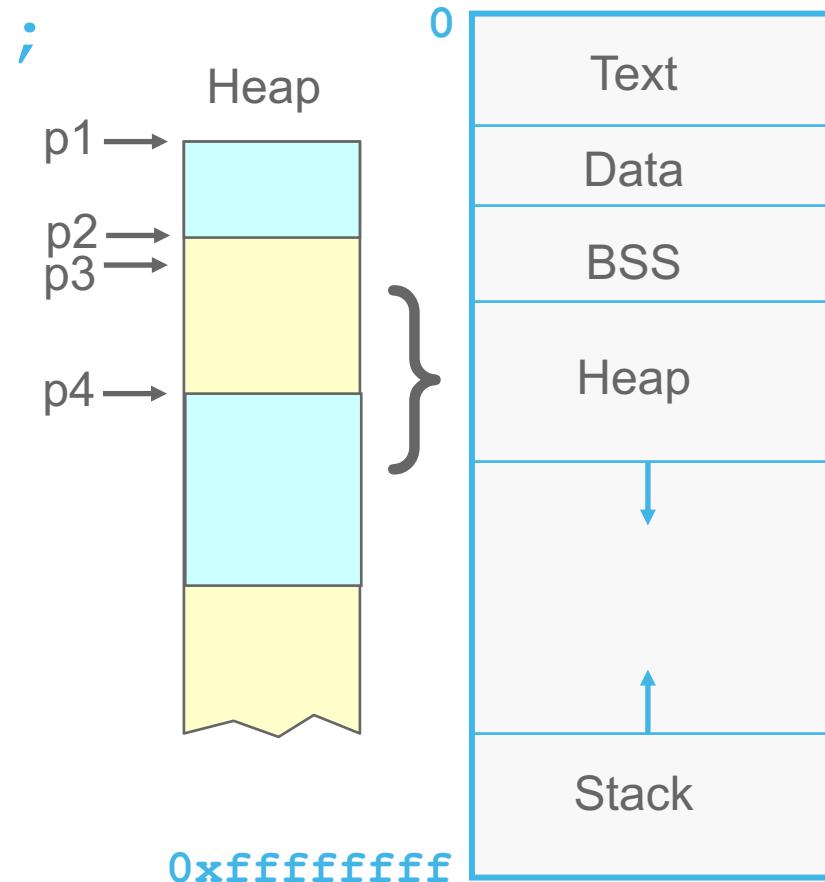
```
char *p1 = malloc(3);
char *p2 = malloc(1);
char *p3 = malloc(4);
free(p2);
→ char *p4 = malloc(6);
free(p3);
char *p5 = malloc(2);
free(p1);
free(p4);
free(p5);
```



Heap: Dynamic Memory

- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

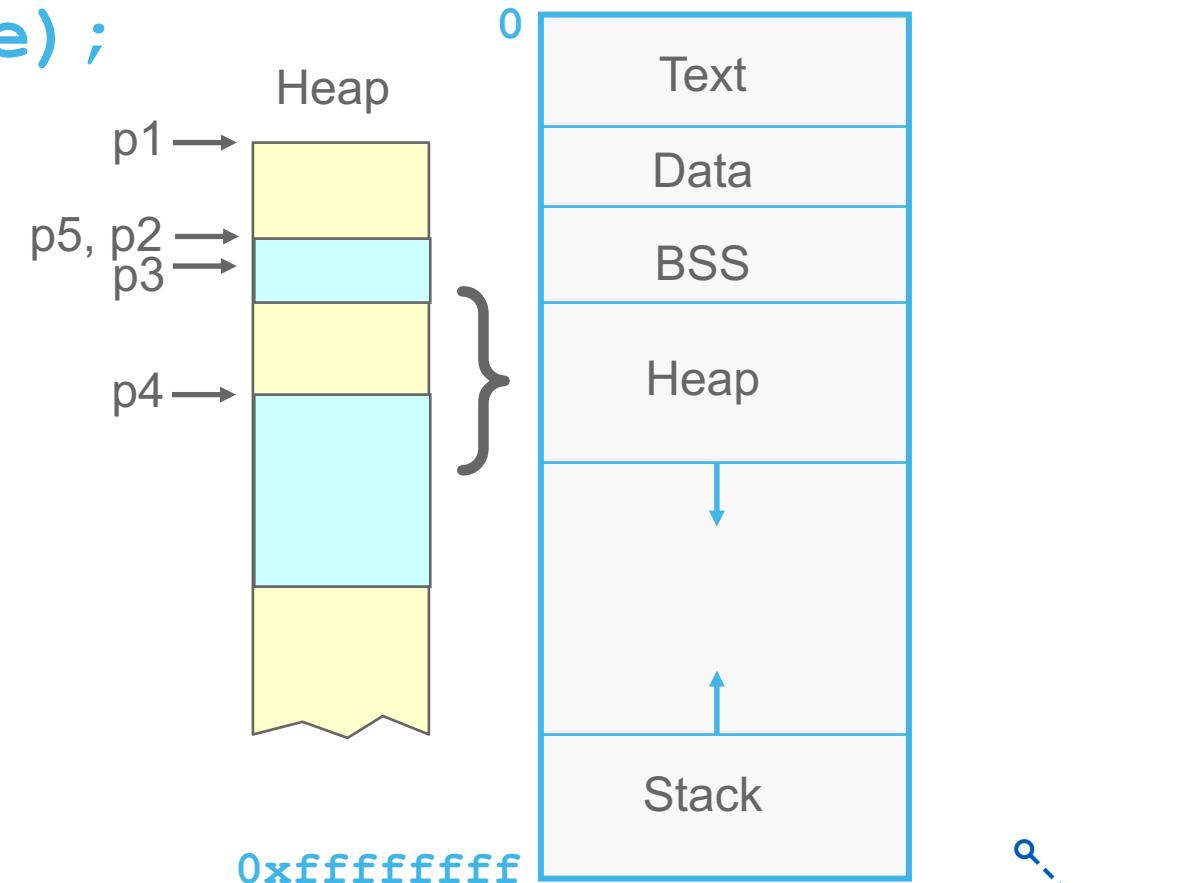
char *p1 = malloc(3);
char *p2 = malloc(1);
char *p3 = malloc(4);
free(p2);
char *p4 = malloc(6);
free(p3);
char *p5 = malloc(2);
free(p1);
free(p4);
free(p5);
```



# Heap: Dynamic Memory

- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

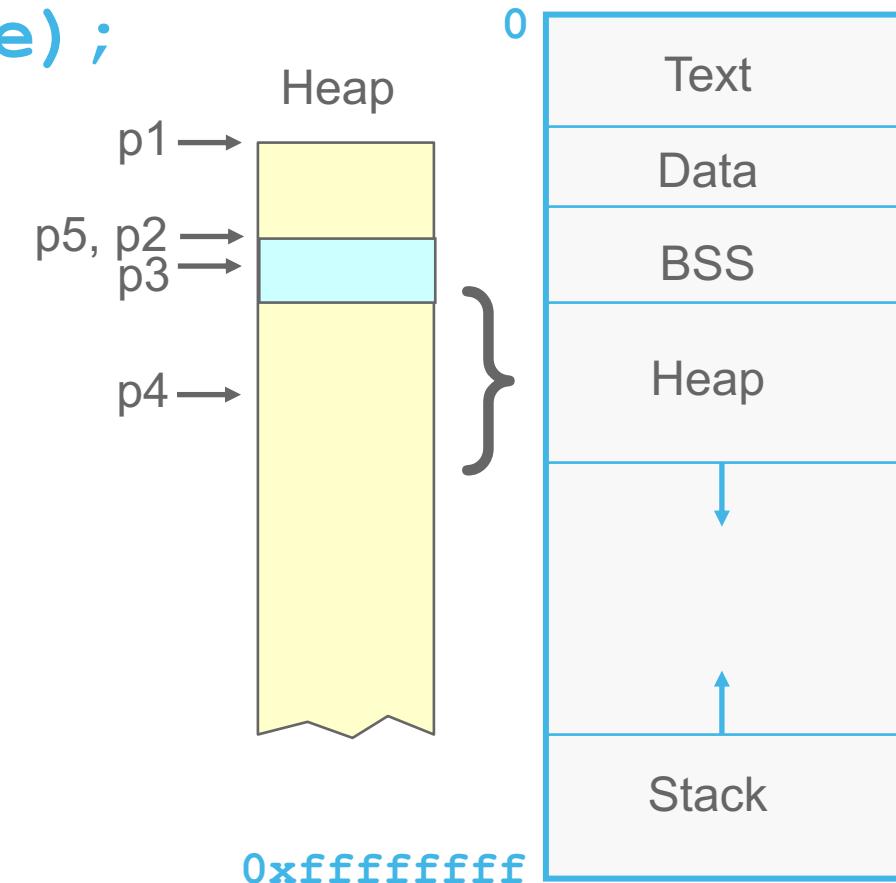
char *p1 = malloc(3);
char *p2 = malloc(1);
char *p3 = malloc(4);
free(p2);
char *p4 = malloc(6);
free(p3);
char *p5 = malloc(2);
free(p1);
free(p4);
free(p5);
```



Heap: Dynamic Memory

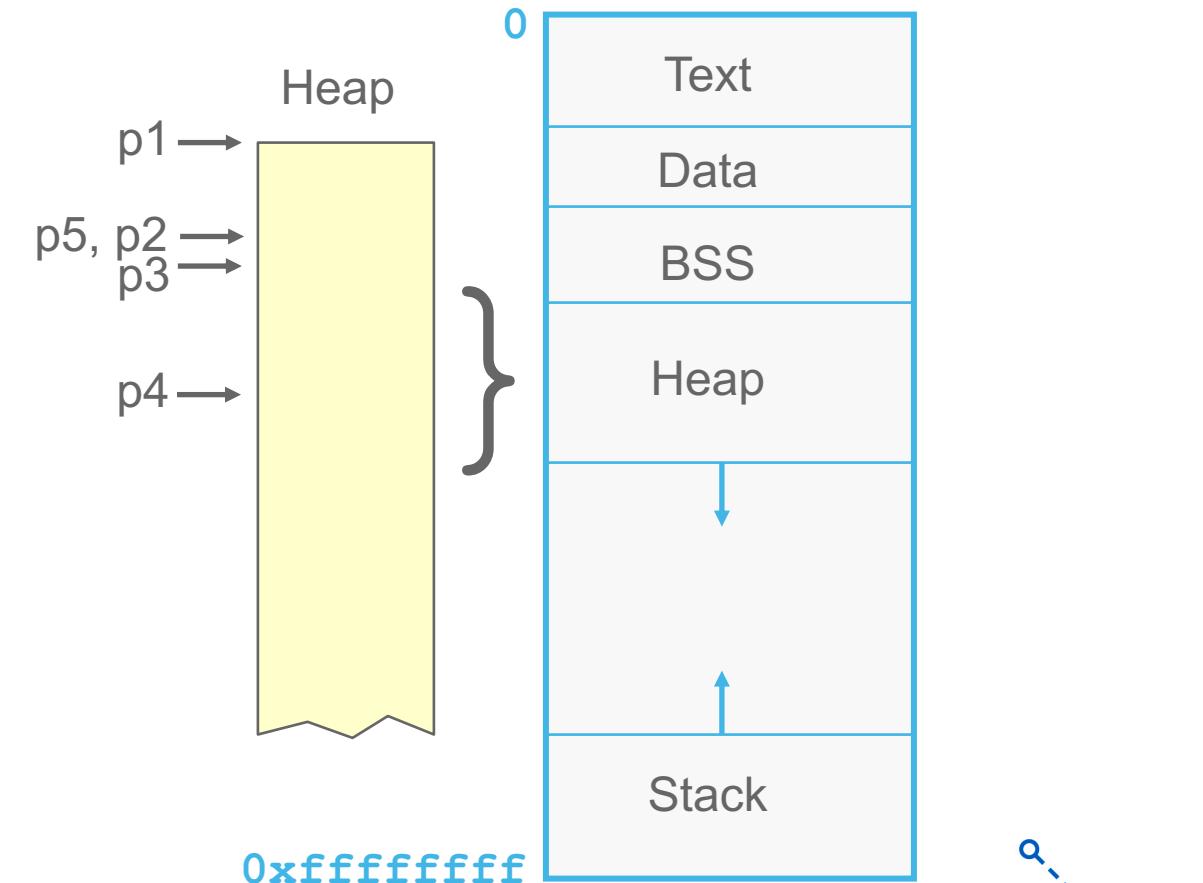
- ```
#include <stdlib.h>
void *malloc(size_t size);
void free(void *ptr);

char *p1 = malloc(3);
char *p2 = malloc(1);
char *p3 = malloc(4);
free(p2);
char *p4 = malloc(6);
free(p3);
char *p5 = malloc(2);
free(p1);
free(p4);
free(p5);
```



# Heap: Dynamic Memory

- #include <stdlib.h>  
void \*malloc(size\_t size);  
void free(void \*ptr);  
  
char \*p1 = malloc(3);  
char \*p2 = malloc(1);  
char \*p3 = malloc(4);  
free(p2);  
char \*p4 = malloc(6);  
free(p3);  
char \*p5 = malloc(2);  
free(p1);  
free(p4);  
free(p5);



# Readings

