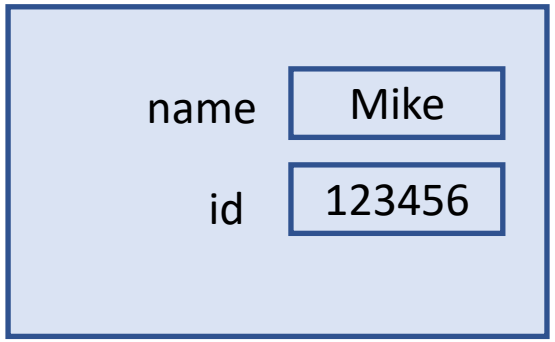
CS 354
Machine Organization and Programming
Lecture 08

Linked Lists

Michael Doescher
Summer 2020

Struct Review

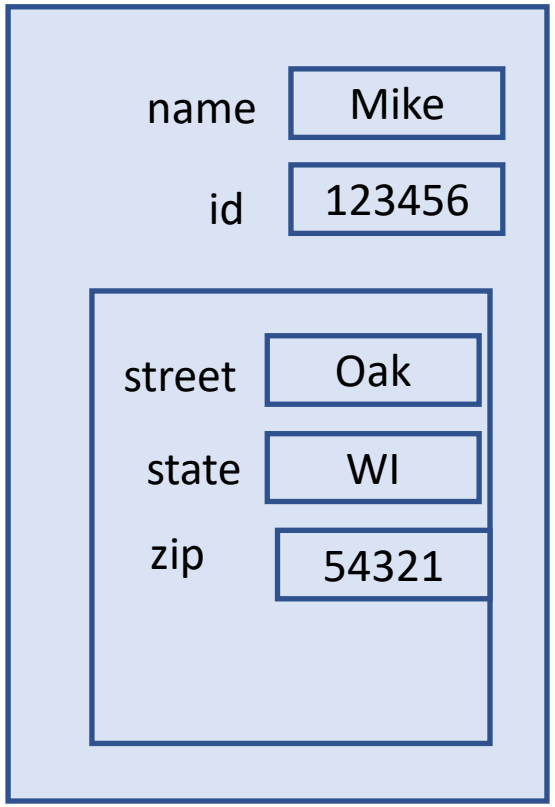
```c
#include <stdio.h>

struct person {
    char name[100];
    int id;

};

```



How many bytes does this struct need?

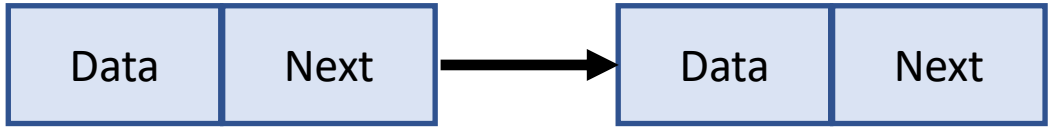
Struct Review

```
 1 #include <stdio.h>
 2
 3 struct address {
 4     char street[100];
 5     char state[3];
 6     int zip;
 7 };
 8
 9 struct person {
10     char name[100];
11     int id;
12     struct address addr;
13 };
```



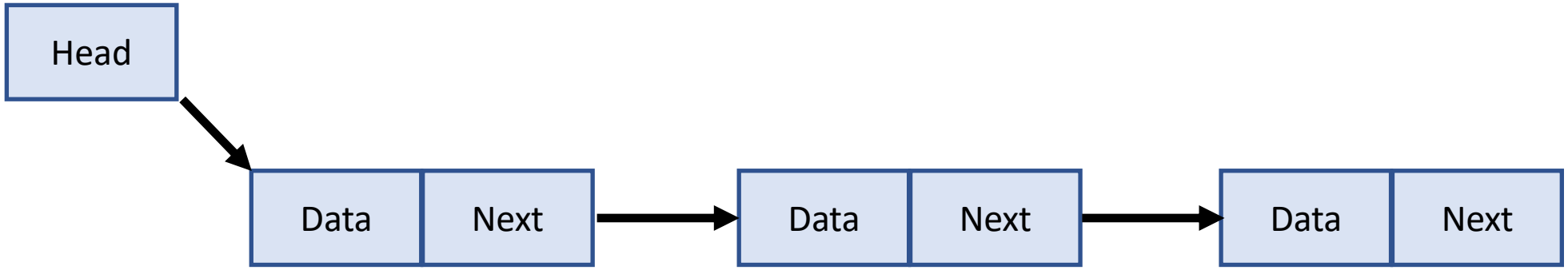
How many bytes does this struct need?

Linked Lists

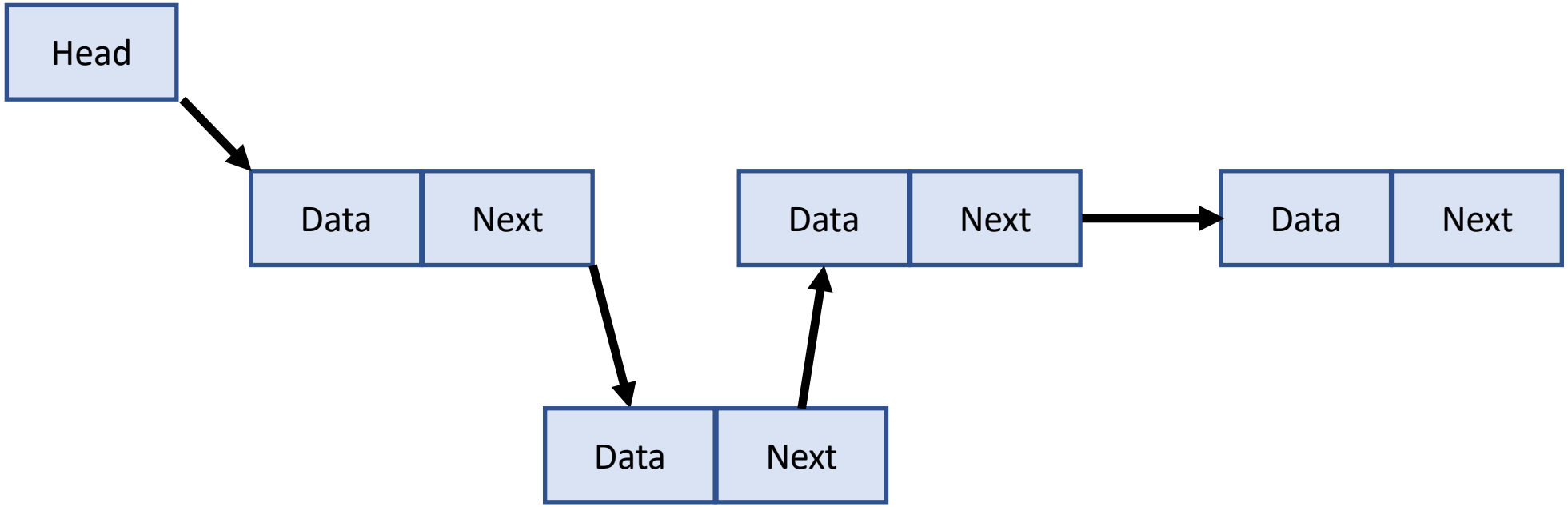


Linked Lists

Linked Lists



Linked Lists



Linked Lists

```
1 #include <stdio.h>
2
3 struct node {
4     int data;
5     struct node next;
6 };
```

Does this work?

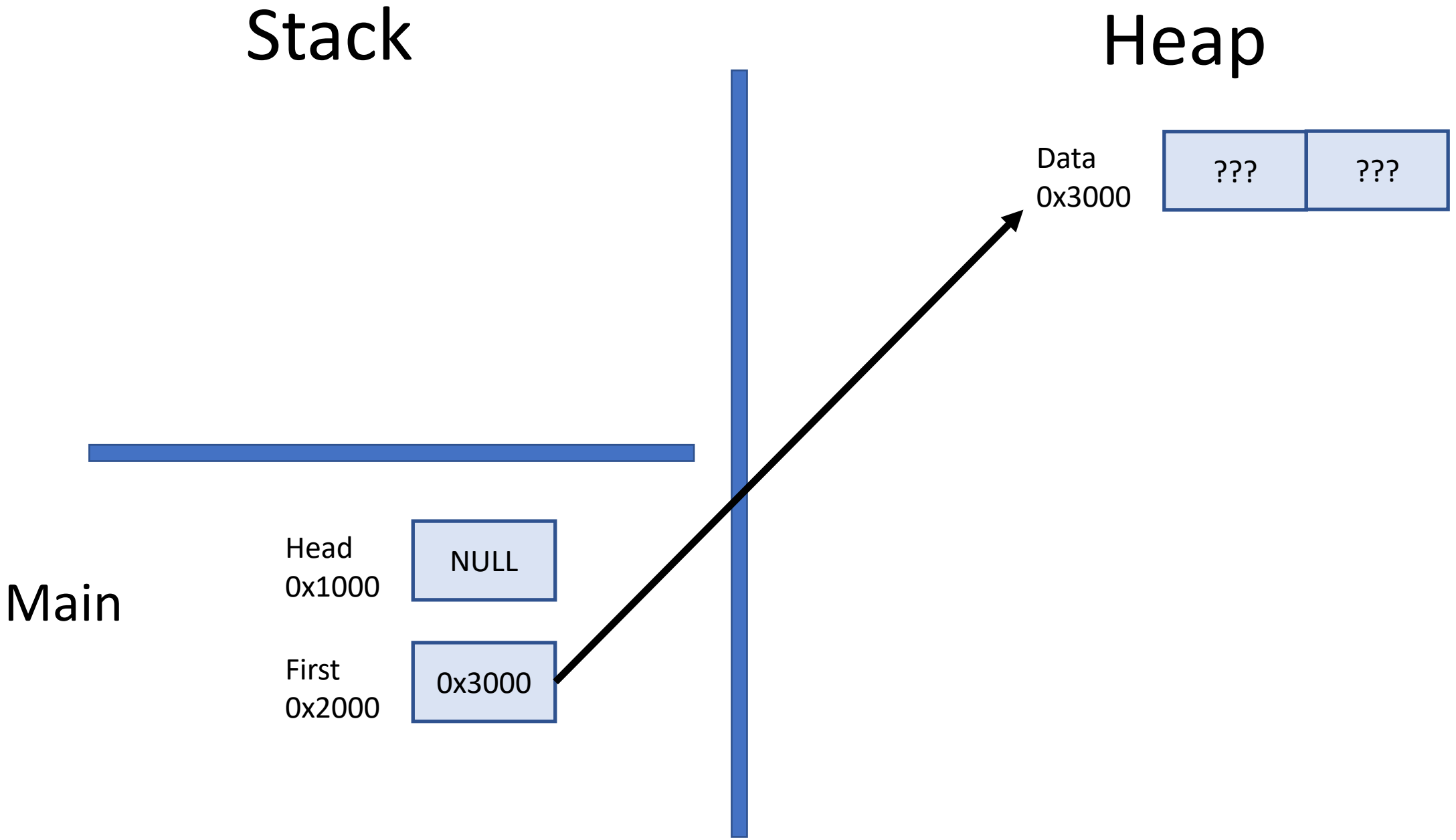
Linked Lists

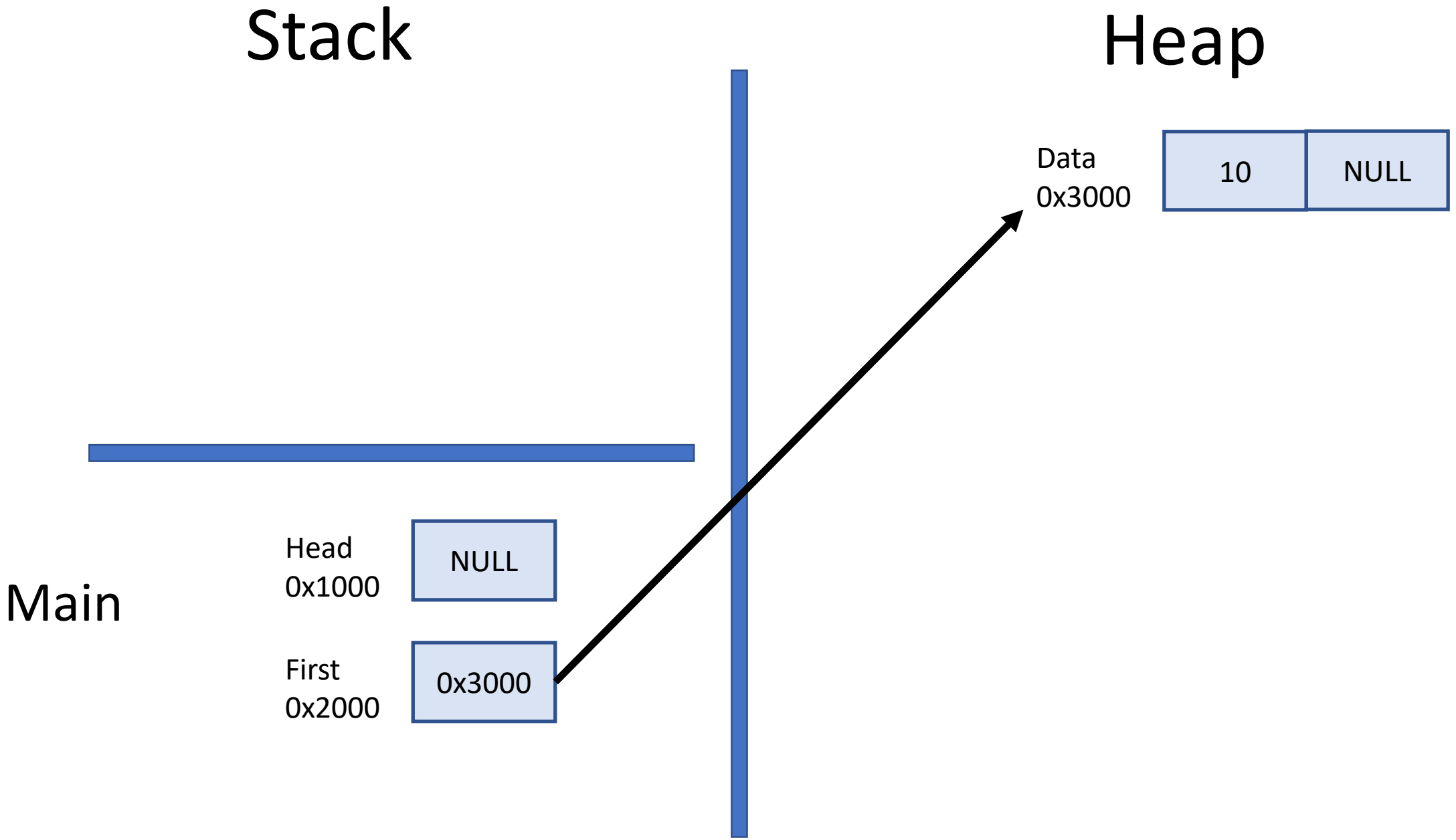
```
1 #include <stdio.h>
2
3 struct node {
4     int data;
5     struct node *next;
6 };
```

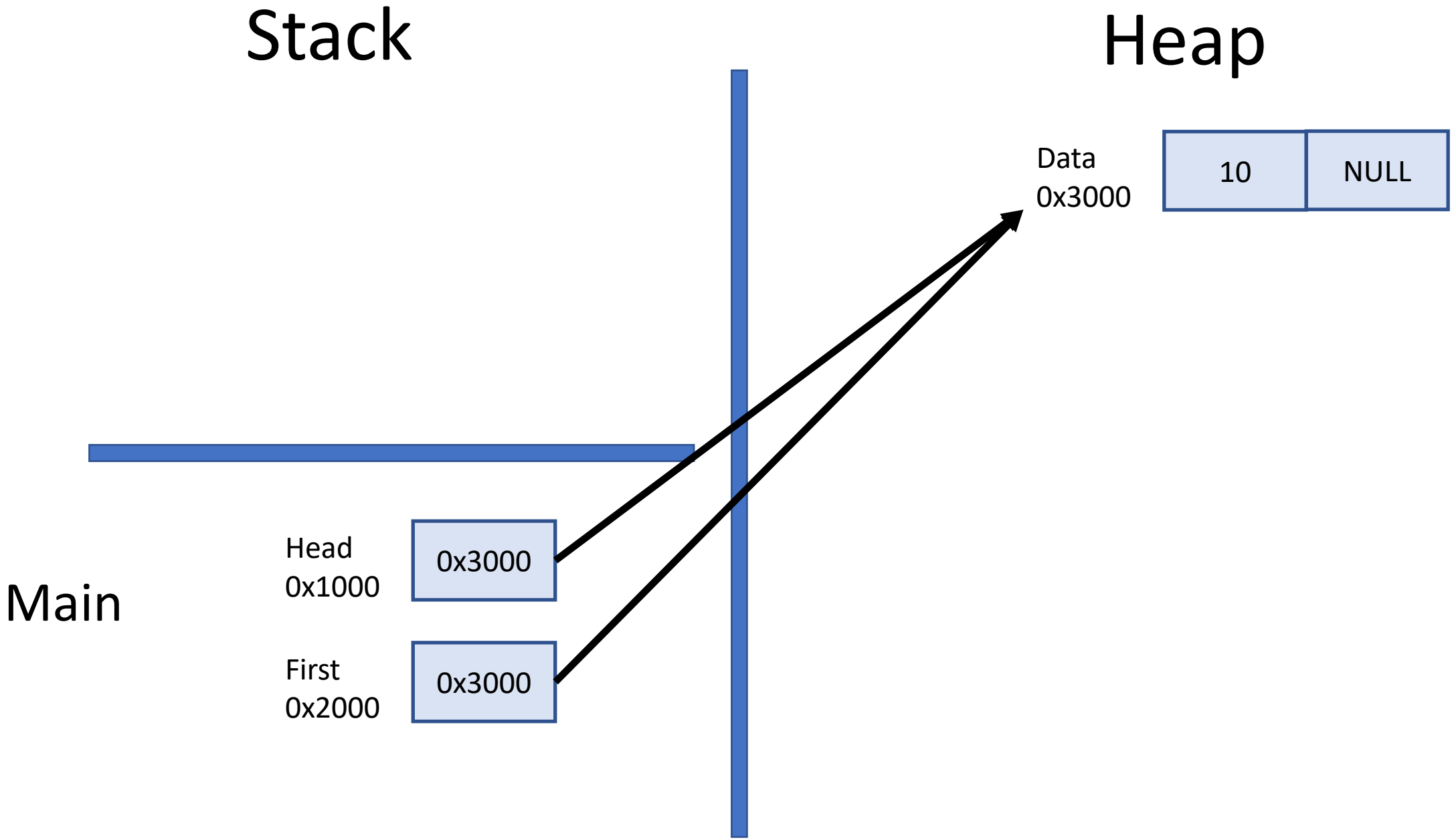
Use a pointer!

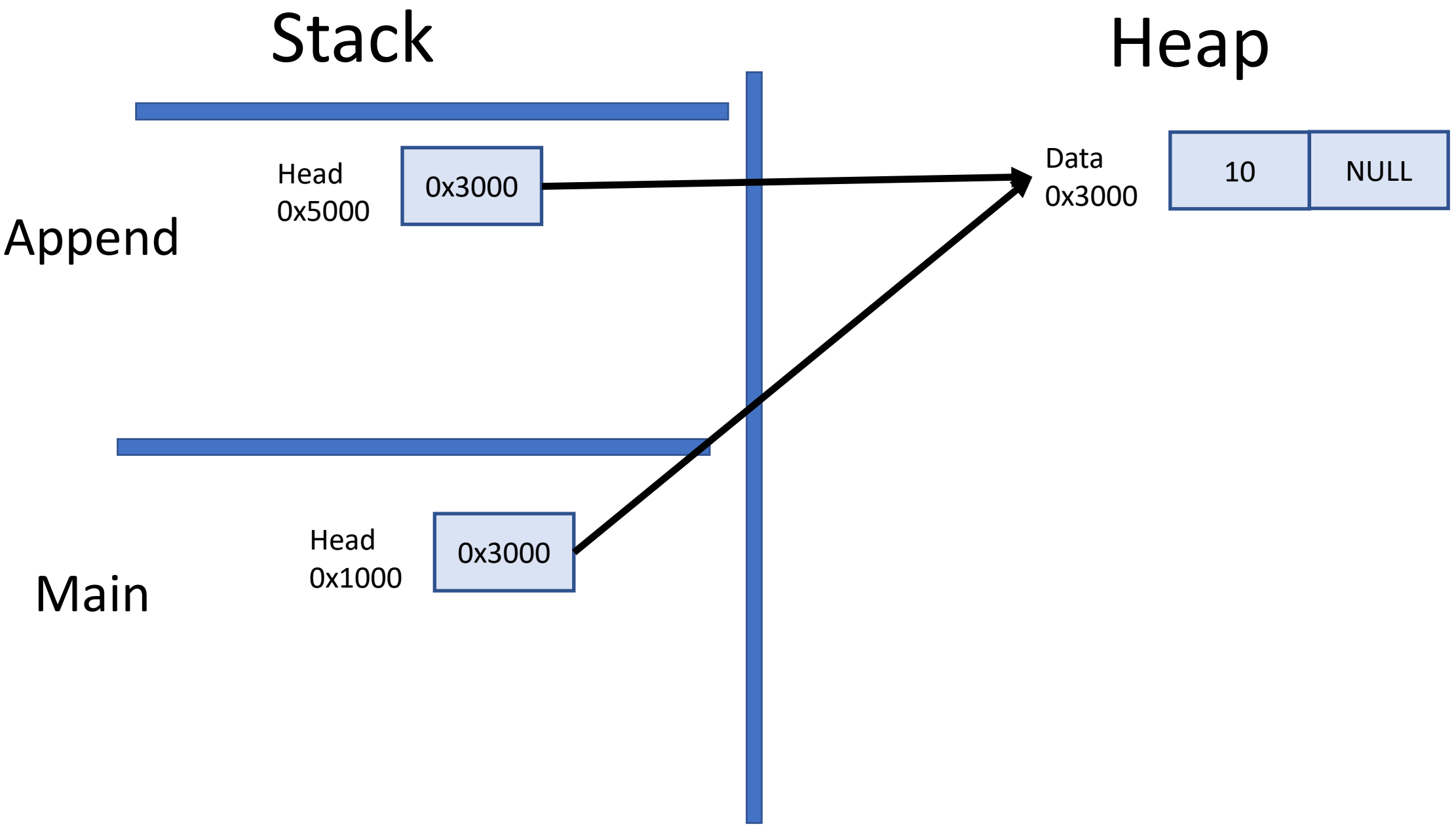
Stack

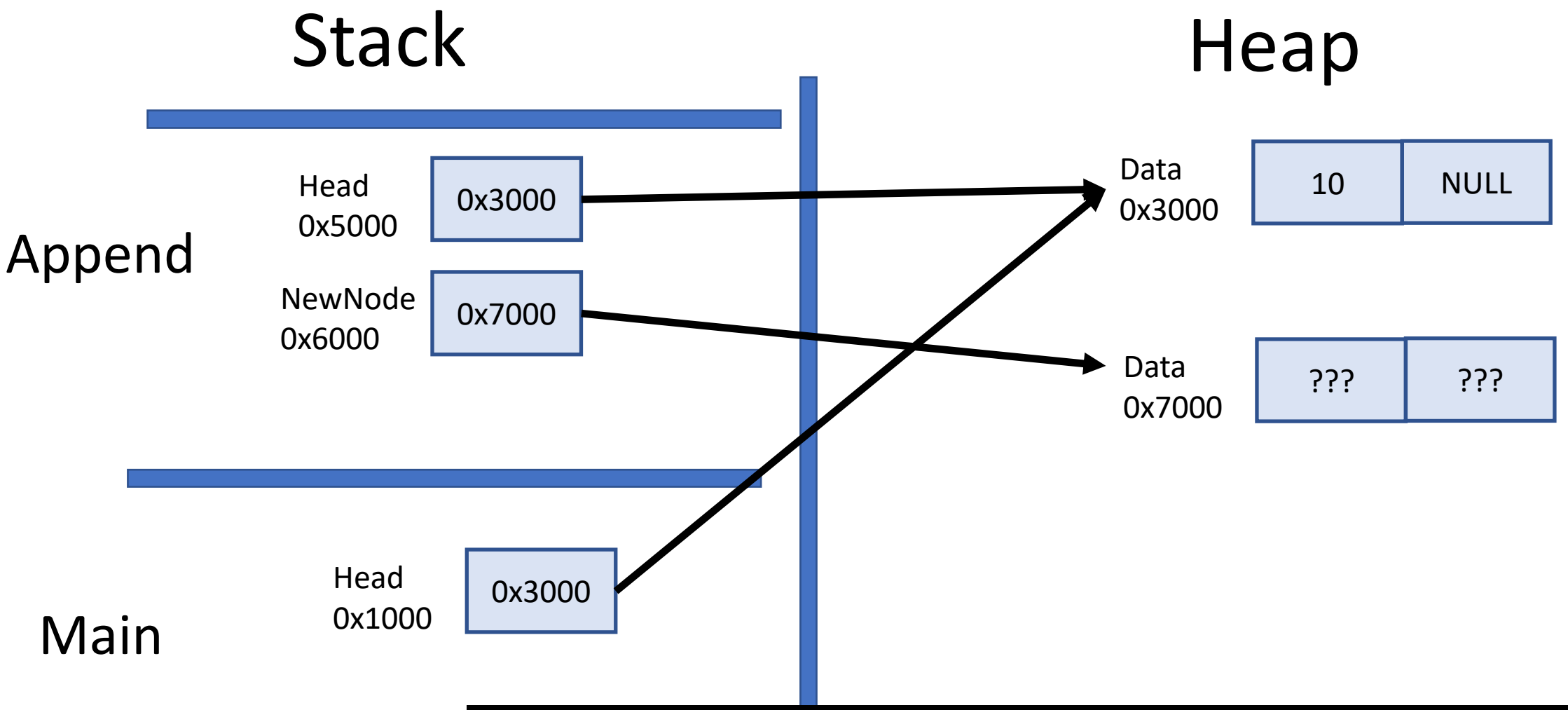
Heap


Stack
Heap
Data
0x3000
???
???
Main
Head
0x1000
NULL
First
0x2000
0x3000

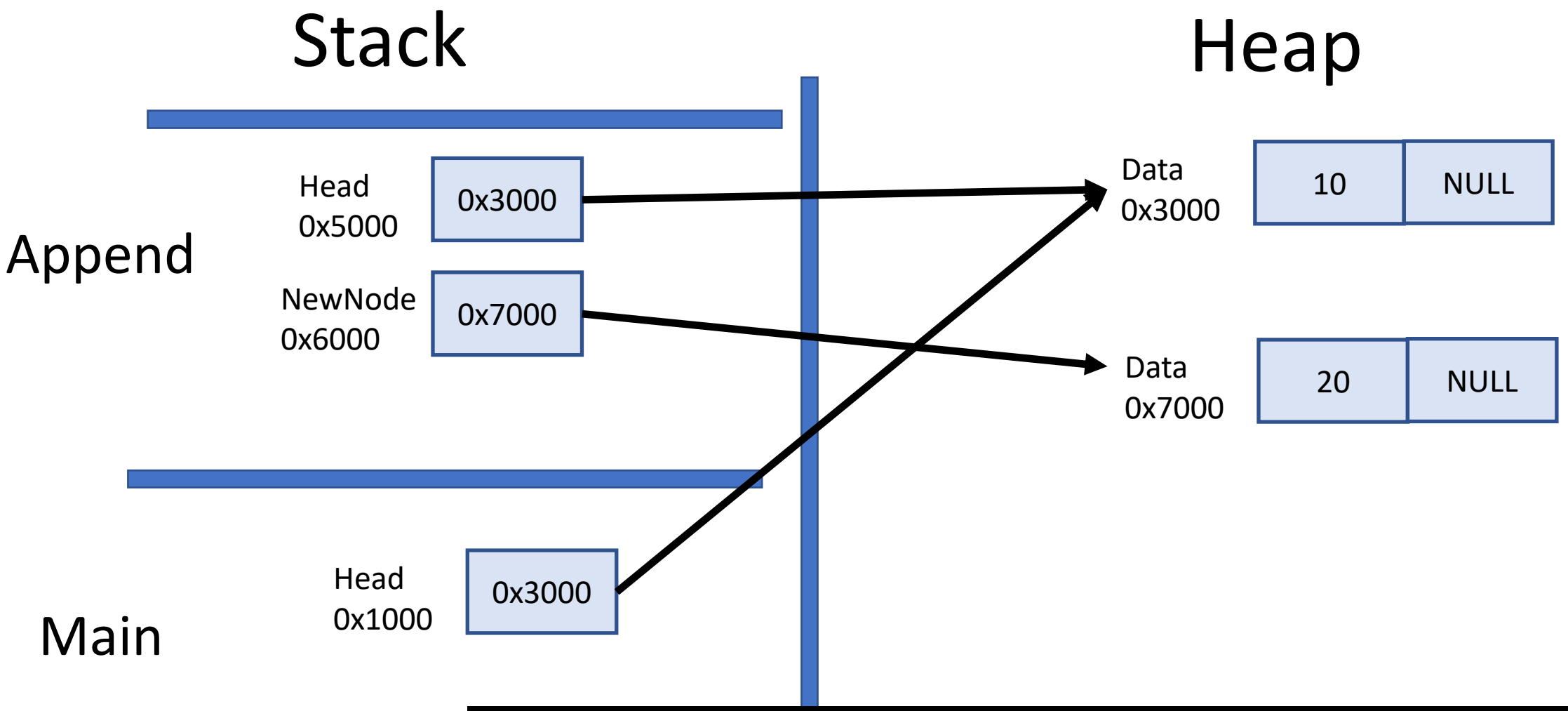

Stack
Heap
Data
0x3000
10
NULL
Head
0x1000
NULL
Main
First
0x2000
0x3000


Stack
Heap
Data
0x3000
10
NULL
Main
Head
0x1000
0x3000
First
0x2000
0x3000

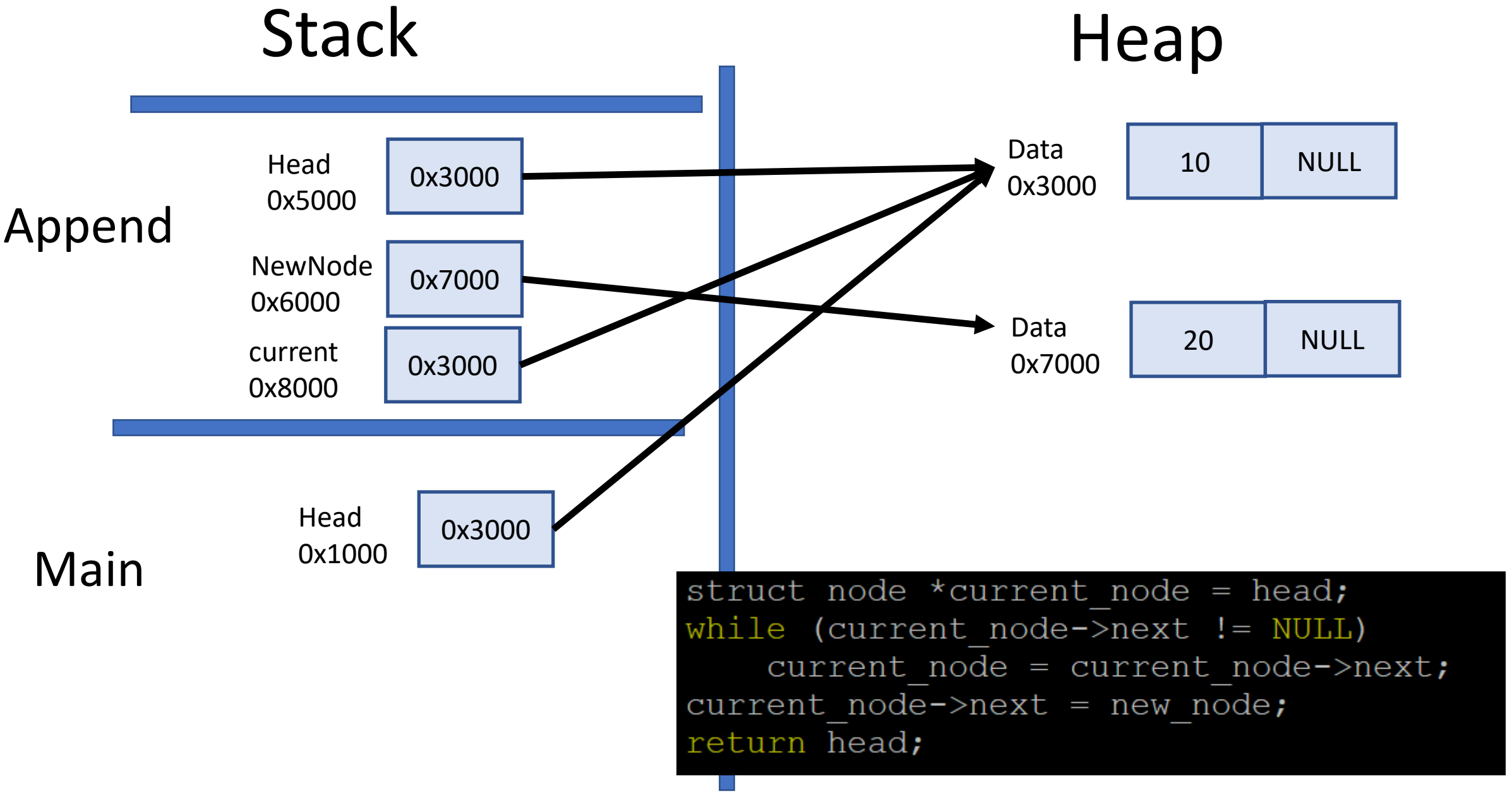

Stack
Heap
Append
Head
0x5000
0x3000
Data
0x3000
10
NULL
Main
Head
0x1000
0x3000

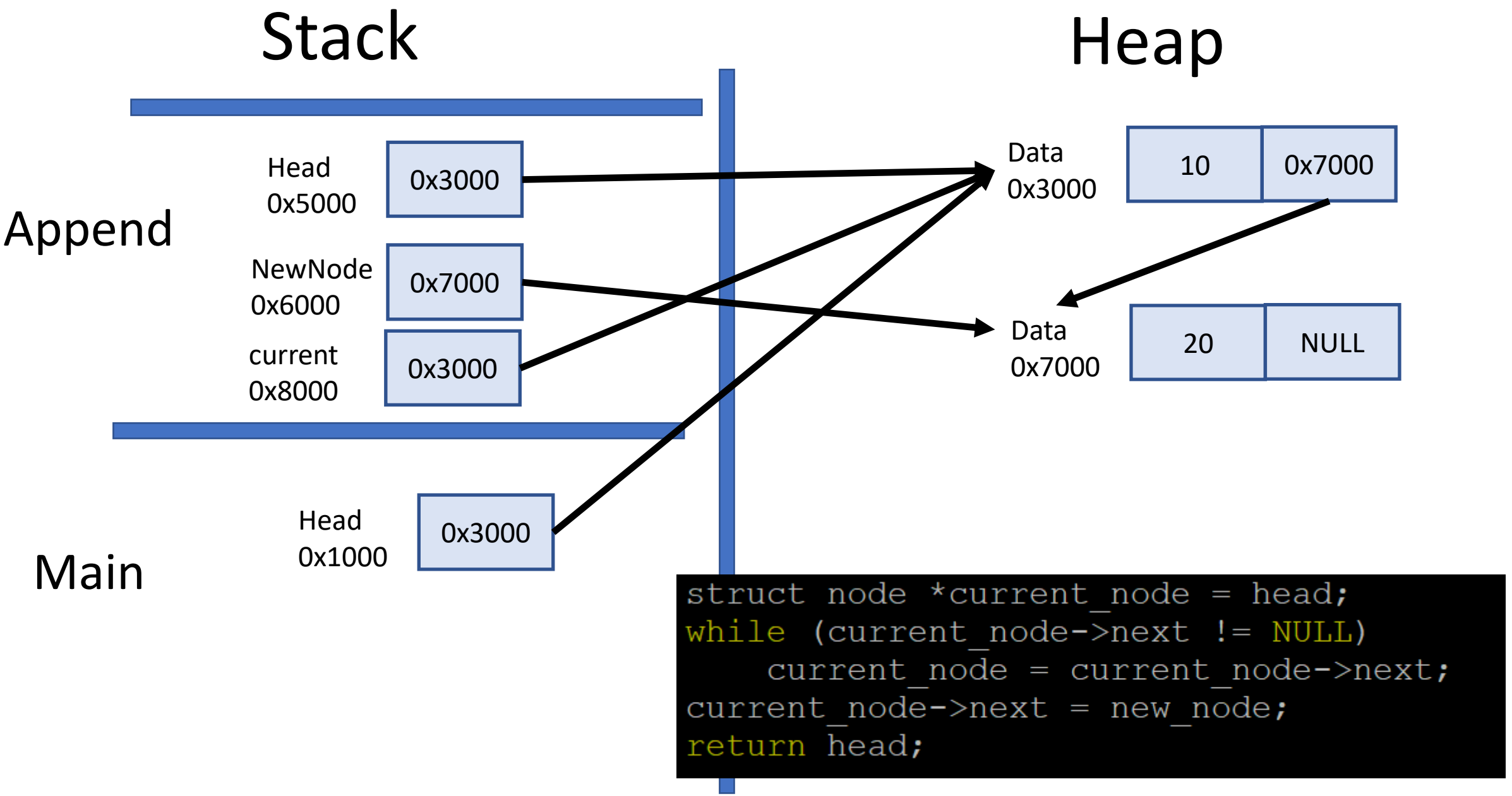


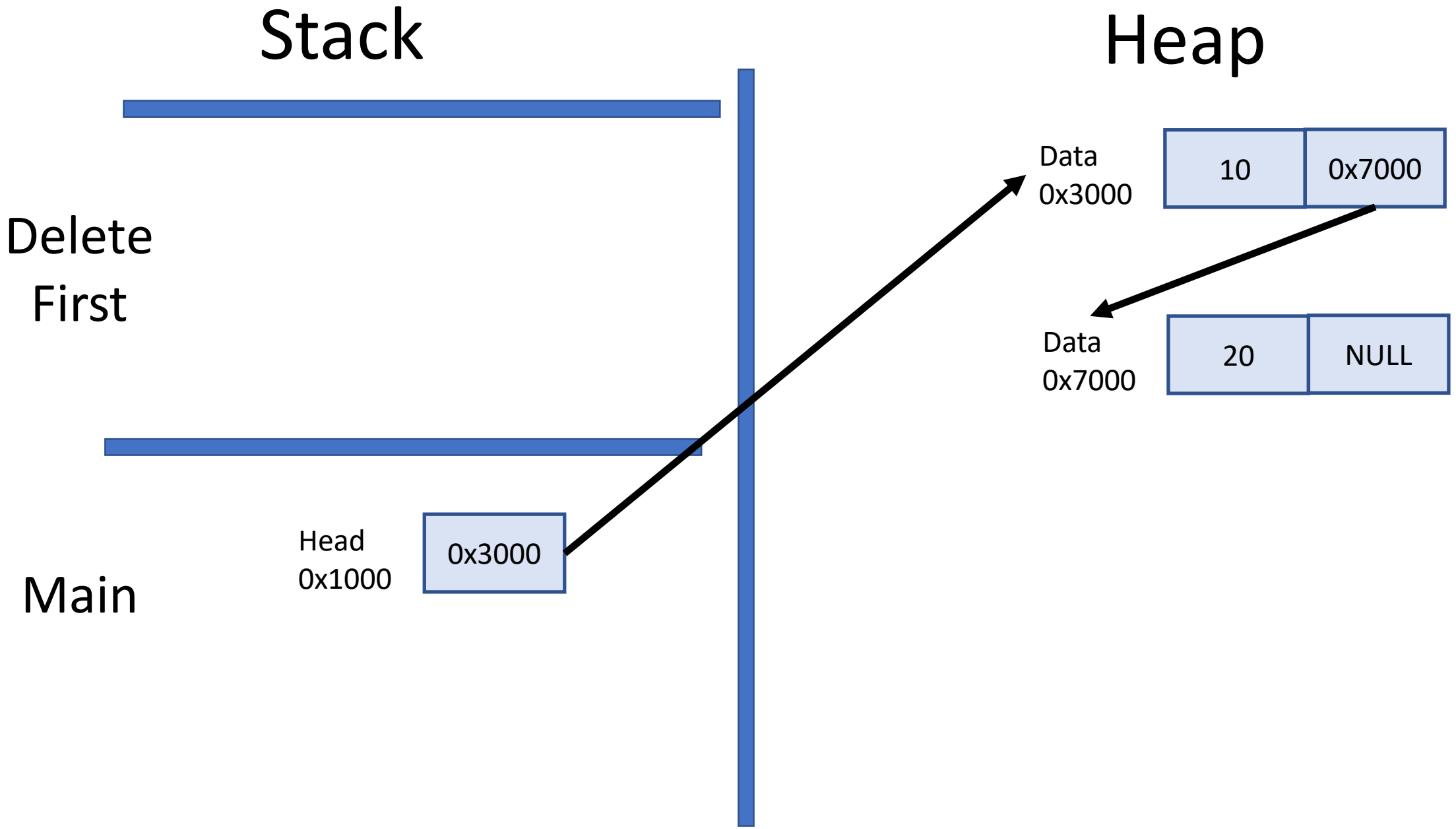

```
// create the new node
struct node *new_node = malloc(sizeof(struct node));
new_node->data = data;
new_node->next = NULL;
```

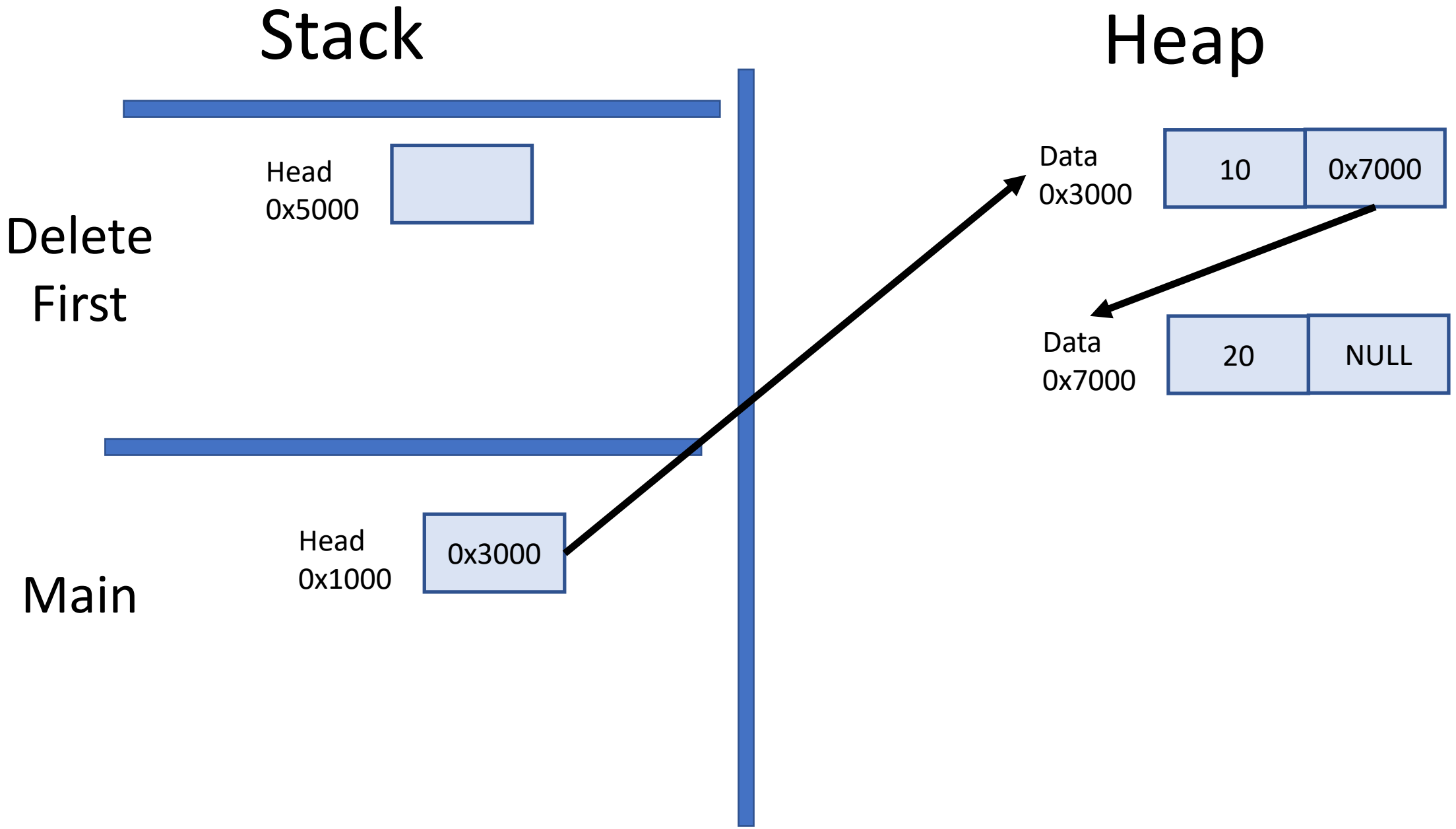


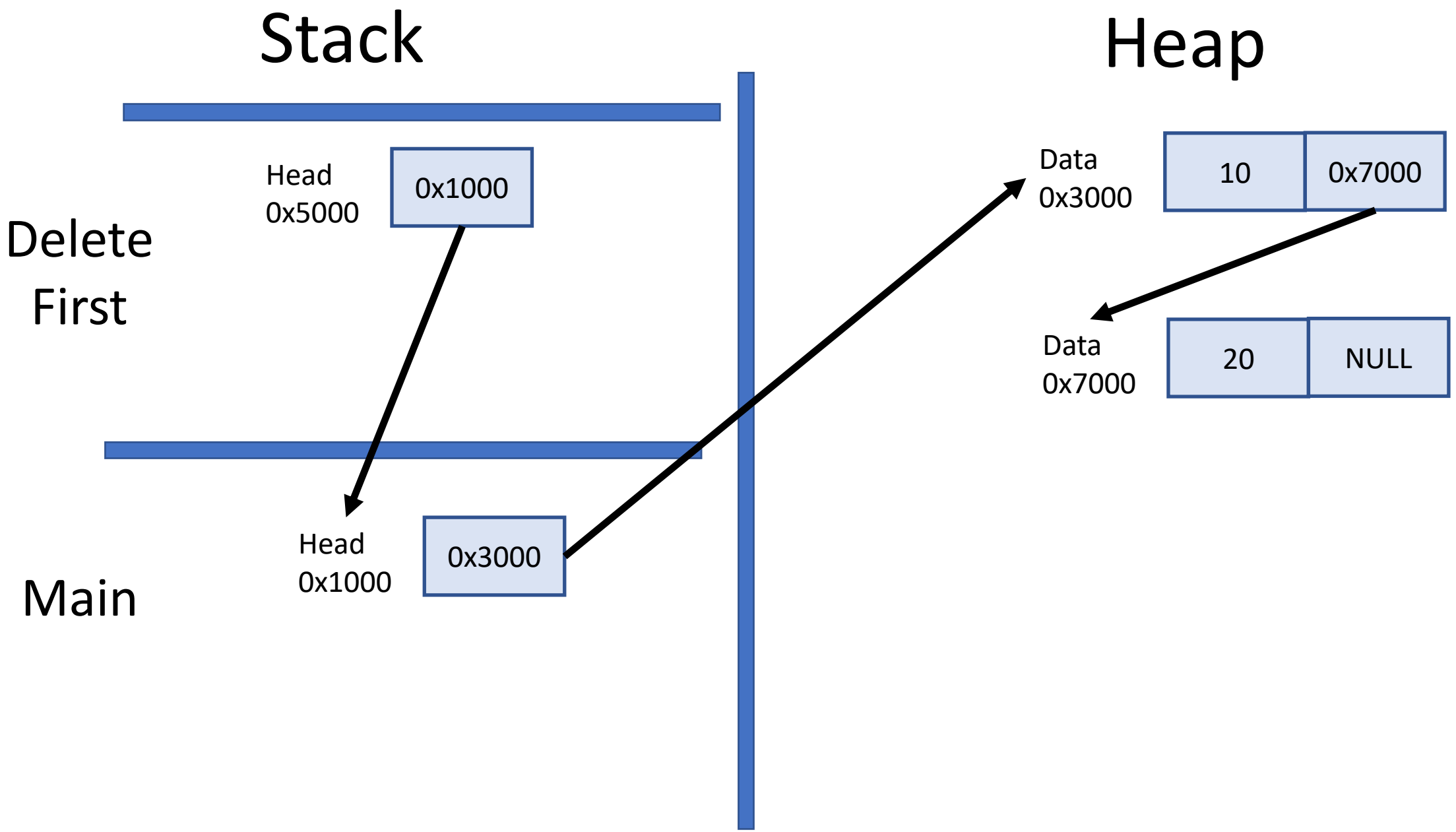

```
// create the new node
struct node *new_node = malloc(sizeof(struct node));
new_node->data = data;
new_node->next = NULL;
```

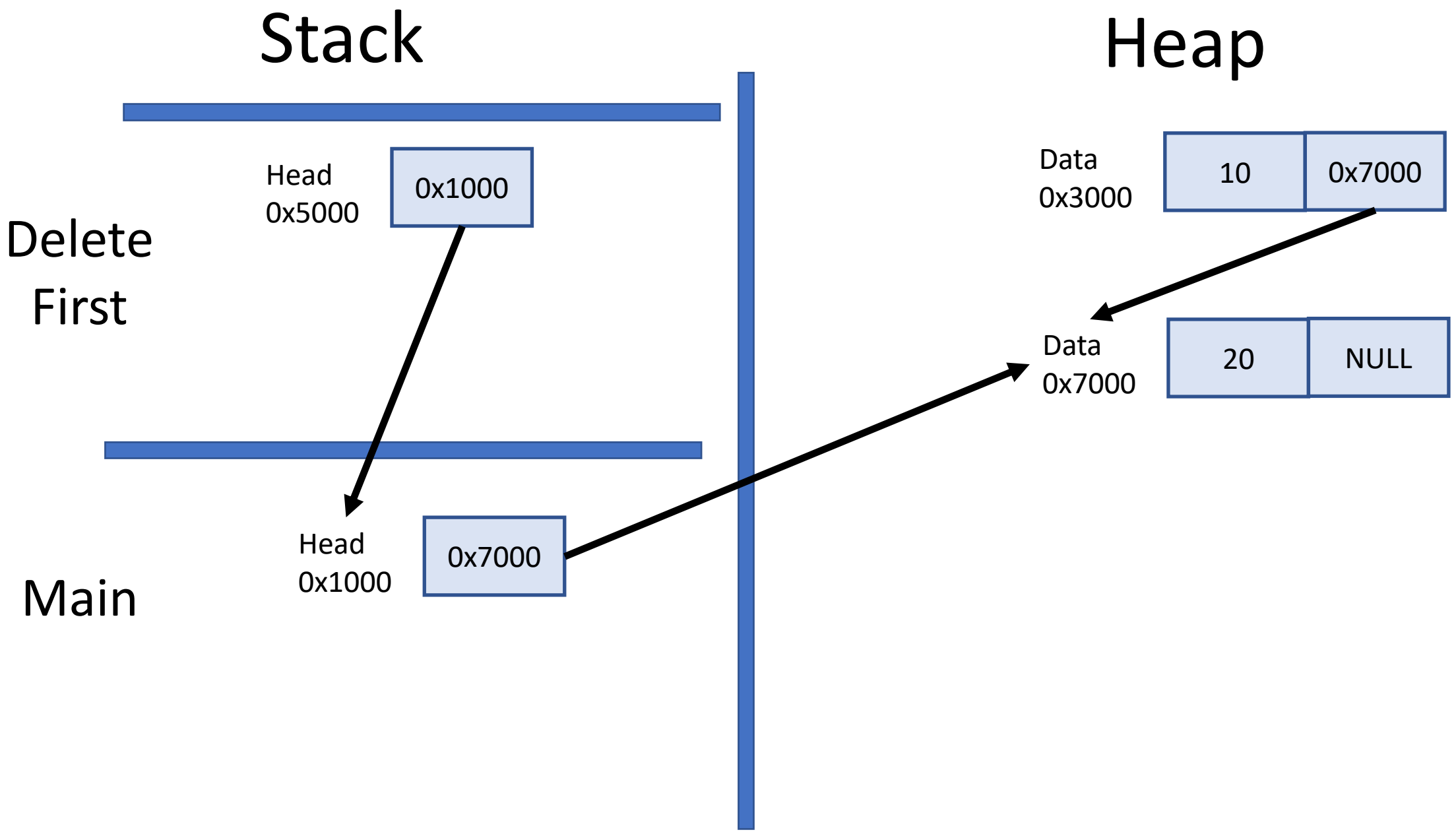

Stack
Heap
Append
Head
0x5000
0x3000
NewNode
0x6000
0x7000
current
0x8000
0x3000
Data
0x3000
10
NULL
Data
0x7000
20
NULL
Main
Head
0x1000
0x3000
struct node *current_node = head;
while (current_node->next != NULL)
current_node = current_node->next;
current_node->next = new_node;
return head;

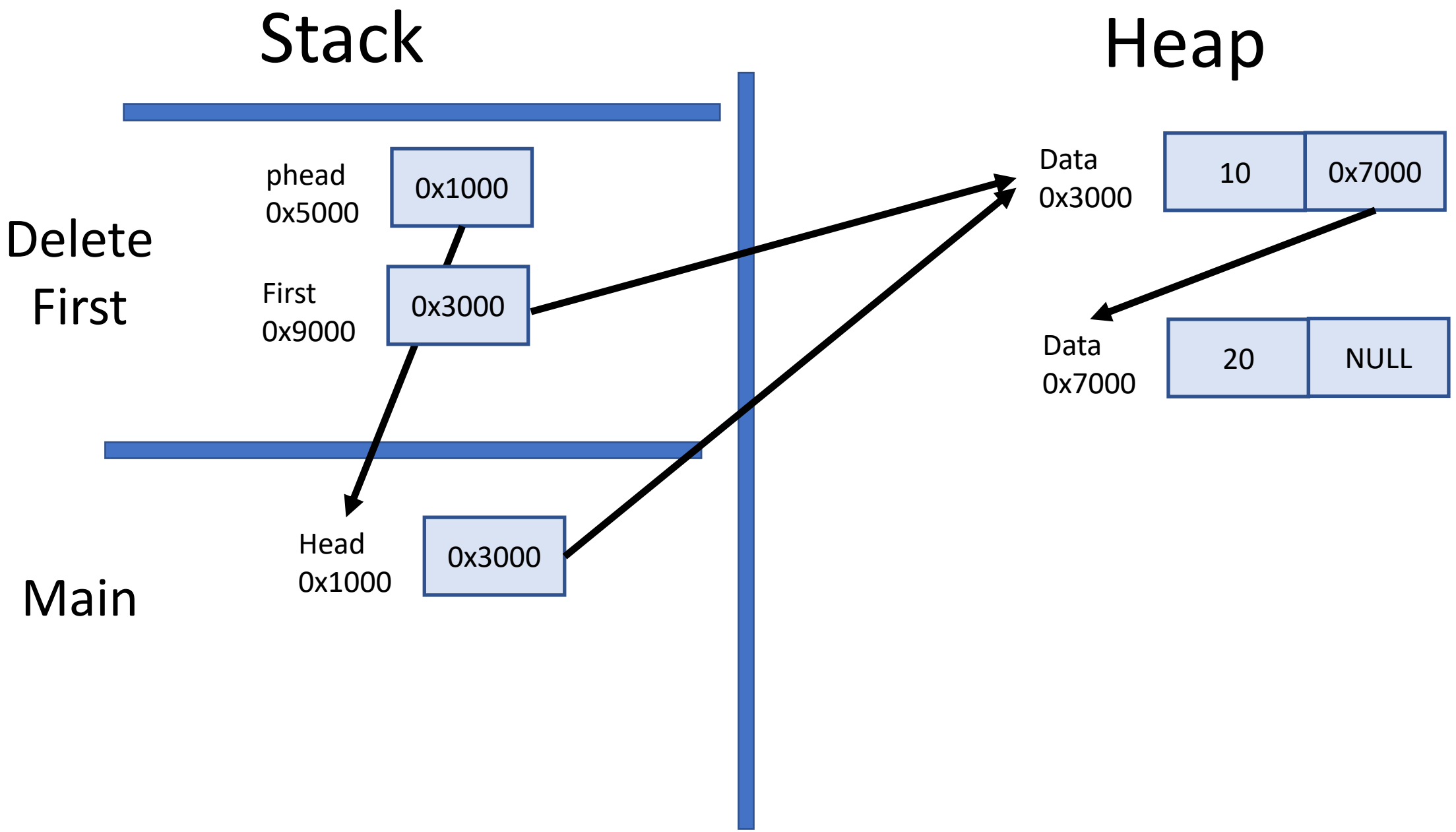

Stack
Heap
Append
Head
0x5000
0x3000
NewNode
0x6000
0x7000
current
0x8000
0x3000
Data
0x3000
10
0x7000
Data
0x7000
20
NULL
Main
Head
0x1000
0x3000
struct node *current_node = head;
while (current_node->next != NULL)
current_node = current_node->next;
current_node->next = new_node;
return head;


Stack
Heap
Delete
First
Main
Head
0x1000
0x3000
Data
0x3000
10
0x7000
Data
0x7000
20
NULL


Stack
Heap
Head
0x5000
Delete
First
Data
0x3000
10
0x7000
Data
0x7000
20
NULL
Head
0x1000
0x3000
Main


Stack
Heap
Head
0x5000
0x1000
Delete
First
Data
0x3000
10
0x7000
Data
0x7000
20
NULL
Head
0x1000
0x3000
Main


Stack
Heap
Head
0x5000
0x1000
Delete
First
Data
0x3000
10
0x7000
Data
0x7000
20
NULL
Head
0x1000
0x7000
Main


Stack
Heap
phead
0x5000
0x1000
Delete
First
First
0x9000
0x3000
Data
0x3000
10
0x7000
Data
0x7000
20
NULL
Head
0x1000
0x3000
Main