

Bài 10: Linked List

Nguyễn Hoàng Anh

Mảng

```
int arr[] = {2,7,4,5,3};
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	7	4	5	3

Khái niệm

```
int arr[] = {2,7,4,5,3};  
=> arr[] = {2,4,5,3};
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	7	4	5	3

1. Dịch trái 4 byte
2. realloc
3. arr[i] = arr[i+1]

Khái niệm

```
int arr[] = {2,7,4,5,3};
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	'\0'	4	5	3

Khái niệm

```
int arr[] = {2,7,4,5,3};
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	4	5	3	3

Khái niệm

```
int arr[] = {2,7,4,5,3};
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	4	5	3	null

Khái niệm

```
int arr[] = {2,7,4,5,3};
```

Address	0x01	0x05	0x09	0x0D
Value	2	4	5	3

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...,1,2};
```

Address	0x01	0x05	0x09	...	x	y
Value	2	7	4	...	1	2

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...,1,2};
```

Address	0x01	0x05	0x09	...	x	y
Value	2	7	4	...	1	2

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...,1,2};
```

Address	0x01	0x05	0x09	...	x	y
Value	2	4	5	...	2	null

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...,1,2};
```

Address	0x01	0x05	0x09	...	x
Value	2	4	5	...	2

Khái niệm

```
int arr[] = {2,7,4,5};  
=> arr[] = {2,7,8,4,5}
```

Address	0x01	0x05	0x09	0x0D
Value	2	7	4	5

1. realloc
2. dịch phải 4 byte
3. `arr[i] = arr[i-1]`

Khái niệm

```
int arr[] = {2,7,4,5};  
=> arr[] = {2,7,8,4,5}
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	7	4	5	null

Khái niệm

```
int arr[] = {2,7,4,5};  
=> arr[] = {2,7,8,4,5}
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	7	4	4	5

Khái niệm

```
int arr[] = {2,7,4,5};  
=> arr[] = {2,7,8,4,5}
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	7	null	4	5

Khái niệm

```
int arr[] = {2,7,4,5};  
=> arr[] = {2,7,8,4,5}
```

Address	0x01	0x05	0x09	0x0D	0x11
Value	2	7	8	4	5

Khái niệm

```
int arr[10000] = {2, 7, 4, 5, 3, ..., 1, 2};  
=> arr[10001] = {2, 1, 7, 4, 5, 3, ..., 1, 2};
```

Address	0x01	0x05	0x09	...	x
Value	2	7	4	...	2

Khái niệm

```
int arr[10000] = {2, 7, 4, 5, 3, ..., 1, 2};  
=> arr[10001] = {2, 1, 7, 4, 5, 3, ..., 1, 2};
```

Address	0x01	0x05	0x09	...	x	x+4
Value	2	7	4	...	2	null

Khái niệm

```
int arr[10000] = {2, 7, 4, 5, 3, ..., 1, 2};  
=> arr[10001] = {2, 1, 7, 4, 5, 3, ..., 1, 2};
```

Address	0x01	0x05	0x09	...	x	x+4
Value	2	null	7	...	1	2

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...,1,2};  
=> arr[10001] = {2,1,7,4,5,3,...,1,2};  
arr + 1
```

Address	0x01	0x05	0x09	...	x	x+4
Value	2	1	7	...	1	2

Khái niệm

Linked list là một cấu trúc dữ liệu trong lập trình máy tính, được sử dụng để tổ chức và lưu trữ dữ liệu. Một linked list bao gồm một chuỗi các "nút" (nodes), mỗi nút chứa một giá trị dữ liệu và một con trỏ (pointer) đến nút tiếp theo trong chuỗi.

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...};
```

Address	0x01	0x05	0x09
Value	2	7	4

Address	0xa1	0xb3	0xef
Value	2	7	4
*Next	NULL	NULL	NULL

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...};
```

Address	0xa1	0xb3	0xef
Value	2	7	4
Next	0xb3	NULL	NULL

Khái niệm

```
int arr[10000] = {2,7,4,5,3,...};
```

Address
Value
Next

0xa1
2
0xb3

0xb3
7
0xef

0xef
4
NULL

0xc5
10
NULL

Thêm 1 node vào phía cuối list (push_back)

Address	0xa1	0xb3	0xef	...	0xe3
Value	2	7	4	...	1
Next	0xb3	0xef	NULL

0xb7
2
NULL

Thêm 1 node vào phía cuối list



Thêm 1 node vào phía cuối list

Address	0xa1	0xb3	0xef	...	0xe3	0xb7
Value	2	7	4	...	1	2
Next	0xb3	0xef	0xb7	NULL

Chèn 1 node vào vị trí bất kỳ trong list

Node

0

1

2

3

4

Address

0xa1

0xb3

0xef

0xce

0xb7

Value

2

7

4

20

12

Next

0xb3

0xef

0xce

0xb7

NULL

- 1: p=0xa1, k=0 -> k=1, p=0xb3
- 2: p=0xb3, k=1 -> k=2, p=0xef

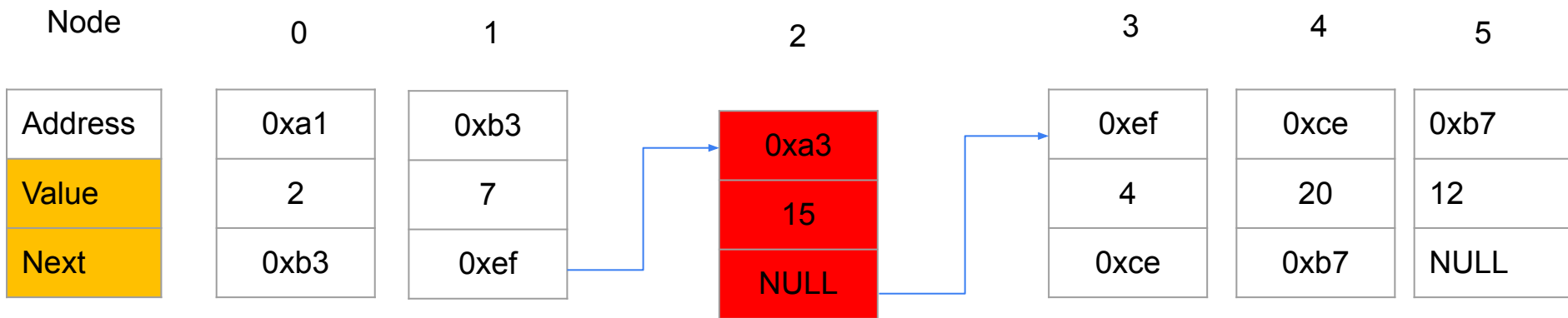
Chèn node 0xa3 vào vị trí thứ 3 của list

0xa3

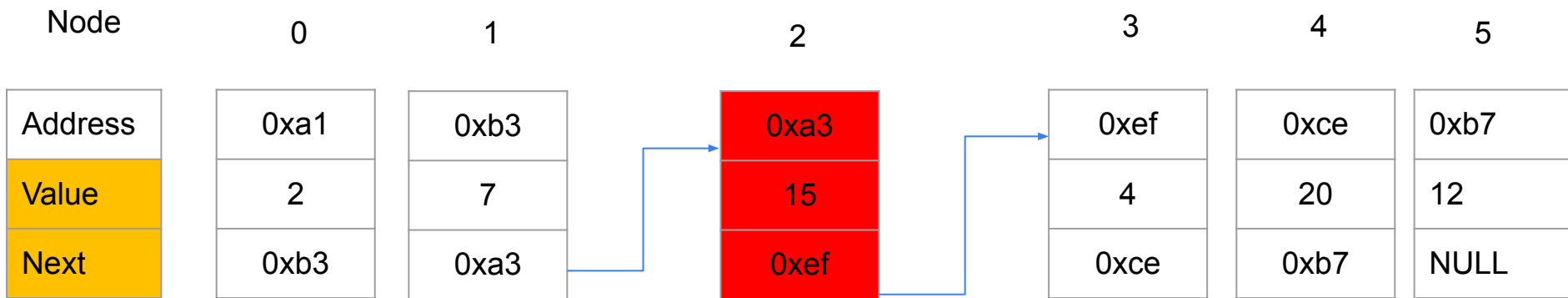
15

NULL

Chèn 1 node vào vị trí bất kỳ trong list



Chèn 1 node vào vị trí bất kỳ trong list



Xóa 1 node bất kỳ trong list

Node

0

1

2

3

4

5

Address

0xa1

0xb3

0xa3

0xef

0xce

0xb7

Value

2

7

15

4

20

12

Next

0xb3

0xa3

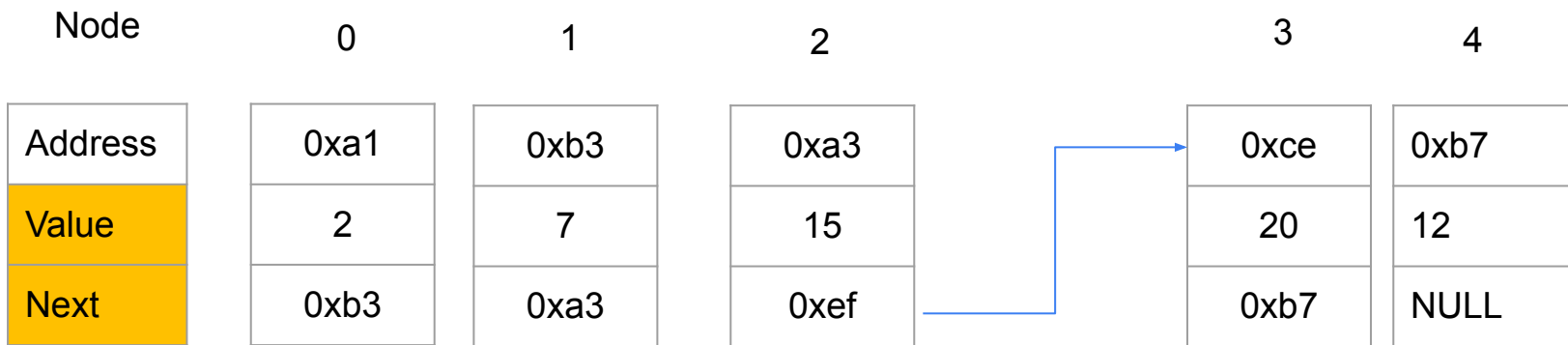
0xef

0xce

0xb7

NULL

Xóa 1 node bất kỳ trong list



Xóa 1 node bất kỳ trong list

Node

0

1

2

3

4

Address

0xa1

0xb3

0xa3

0xce

0xb7

Value

2

7

15

20

12

Next

0xb3

0xa3

0xce

0xb7

NULL

Khái niệm

```
#include <stdio.h>
#include <stdlib.h>

typedef struct node
{
    int value;
    struct node* next;
}node;

node* createNode(int value)
{
    node* ptr = (node*)malloc(sizeof(node));
    ptr->value = value;
    ptr->next = NULL;
    return ptr;
}
```

```
void push_back(node** array, int value)
```

Khái niệm

```
node *createNode(int value);  
void push_back(node** array, int value);  
void push_front(node **array, int value); // them 1 node vao phia truoc  
void pop_back(node **array);  
void pop_front(node **array); // xoa node dau tien  
int front(node *array); // lay gia tri cua node dau tien  
int back(node *array); // lay gia tri cua node cuoi cung  
void insert(node **array, int value, int pos); // them 1 node vao mot vi tri bat ky  
void erase(node **array, int pos); // xoa 1 node tai mot vi tri bat ky  
int size(node *array); // lay kich thước của list  
int get(node *array, int pos);  
bool empty(node *array); // kiểm tra list có rỗng hay không
```