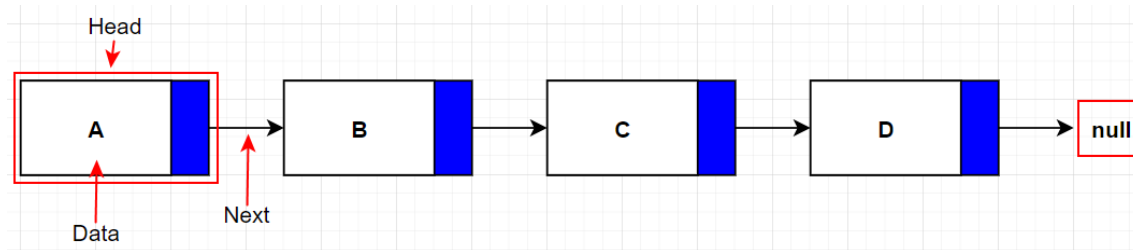


Bài 3.2: Chèn node vào danh sách

- ✓ Tạo danh sách liên kết đơn
- ✓ Thêm node vào đầu danh sách
- ✓ Thêm node vào cuối danh sách
- ✓ Thêm node vào sau node x
- ✓ Duyệt danh sách liên kết
- ✓ Ví dụ và bài tập thực hành

Tạo danh sách liên kết đơn

- Ta thực hiện hai bước: tạo node của danh sách và tạo danh sách.
- Để tạo node ta sử dụng kiểu dữ liệu người dùng tự định nghĩa như struct(trong C) hoặc struct/class trong C++.



- Ví dụ sau tạo node trong ngôn ngữ lập trình C++:

```
template<class T> class Node {
public:
    T data; // dữ liệu của node hiện tại
    Node<T>* next; // con trỏ trỏ tới node kế tiếp

    Node(T data) {
        this->next = nullptr; // khởi tạo mặc định khi tạo node
        this->data = data; // gán giá trị của data cho node
    }
};
```

Tạo danh sách liên kết đơn

Tiếp theo ta tạo danh sách liên kết đơn với các tùy chọn:

- Tạo danh sách liên kết chỉ gồm node head.
- Tạo danh sách liên kết bao gồm hai node head, tail.
- Trong khóa học này sẽ tạo danh sách liên kết theo cả hai cách. Ưu điểm của cách thứ nhất là đơn giản. Nhược điểm là phải thực hiện nhiều thao tác để chèn node vào cuối danh sách liên kết.
- Với cách thứ 2, ưu điểm là dễ dàng chèn thêm node mới vào cuối danh sách. Nhược điểm là tốn thêm chi phí bộ nhớ cho node tail.

Tạo danh sách liên kết đơn

```
template<class T> class LinkedList {  
private:  
    Node<T>* head; // con trỏ trỏ đến node đầu tiên trong danh sách  
    Node<T>* tail; // con trỏ trỏ tới node cuối của danh sách  
public:  
    LinkedList() { ... }  
  
    void add(T data) { ... }  
  
    void addAfterX(T data, T x) { ... }  
  
    void addTail(T data) { ... }  
  
    void showList() { ... }  
  
    bool isEmpty() { ... }  
};
```

Thêm node vào đầu danh sách liên kết

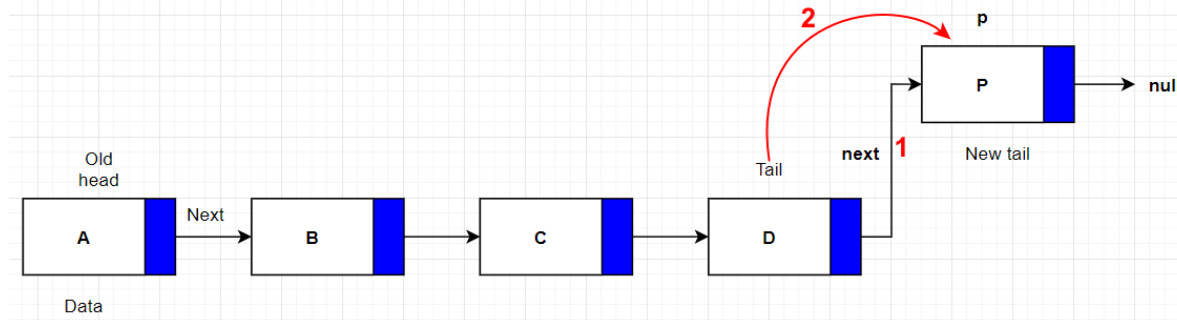
- Các bước thực hiện để chèn node p vào đầu danh sách liên kết:
- Kiểm tra xem danh sách có rỗng hay không.
- Nếu danh sách rỗng, ta gán **head = tail = p**.
- Nếu danh sách không rỗng:
 - Gán head cho p->next: **p->next = head;**
 - Gán lại head là p: **head = p;**

```
void add(T data) {  
    Node<T>* p = new Node<T>(data); // tạo node mới p  
    if (isEmpty()) { // kiểm tra xem danh sách rỗng không  
        head = tail = p; // gán giá trị cho head, tail  
    }  
    else { // nếu danh sách không rỗng  
        p->next = head; // cập nhật node next của p  
        head = p; // cập nhật lại node head  
    }  
}
```

Thêm node vào cuối danh sách

Giả sử ta muốn thêm node p vào cuối danh sách liên kết hiện tại:

- Nếu danh sách rỗng, ta gán **head = tail = p**.
- Nếu danh sách không rỗng, ta thực hiện:
 - Gán p làm next của tail: **tail->next = p**;
 - Cập nhật lại tail: **tail = p**;

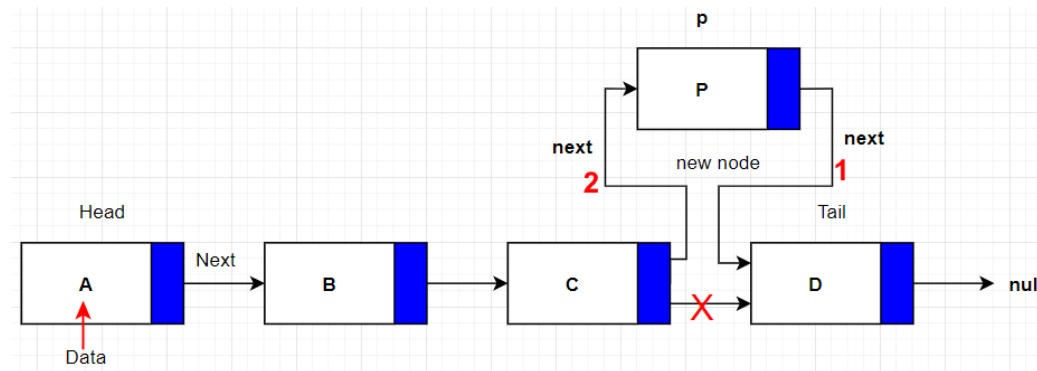


```
void addTail(T data) { // chèn node vào cuối danh sách liên kết
    Node<T>* p = new Node<T>(data); // tạo node mới p
    if (!isEmpty()) {
        tail->next = p; // cập nhật node next của tail
        tail = p; // cập nhật lại tail
    }
    else {
        head = tail = p; // gán head, tail cùng bằng p
    }
}
```


Thêm node vào sau node x

Giả sử ta muốn thêm node p sau node có giá trị C:

- Tìm node chứa data bằng C gọi là nodeX.
- Nếu tìm thấy:
 - Gán node next của nodeX cho next của p: $p \rightarrow \text{next} = \text{nodeX} \rightarrow \text{next}$;
 - Gán node p cho node next của nodeX: $\text{nodeX} \rightarrow \text{next} = p$;
- Nếu không tìm thấy node cần tìm, thông báo ra màn hình và kết thúc.



Thêm node vào sau node x

```
void addAfterX(T data, T x) {  
    Node<T>* p = new Node<T>(data); // tạo node mới p  
    Node<T>* nodeX = head; // bắt đầu từ node head  
    while (nodeX != nullptr) { // tìm nodeX  
        if (nodeX->data == x) { // nếu tìm thấy  
            break; // kết thúc việc tìm kiếm  
        }  
        nodeX = nodeX->next; // chuyển tới node kế tiếp  
    }  
    if (nodeX != nullptr) { // nếu tìm thấy  
        p->next = nodeX->next; // cập nhật next của p  
        nodeX->next = p; // cập nhật next của nodeX  
    }  
    else { // nếu không tìm thấy  
        cout << "Khong tim thay node muc tieu\n";  
    }  
}
```


Duyệt danh sách liên kết

- Khai báo node p và khởi tạo bằng head: `p = head;`
- Lặp chừng nào p chưa null:
 - Xuất ra giá trị của `p->data`;
 - Cập nhật p: `p = p->next;`

```
void showList() { // hàm hiển thị danh sách liên kết
    Node<T>* p = head; // khởi đầu từ node head
    while (p != nullptr) // lặp chừng nào p chưa null
    {
        cout << p->data << " "; // hiển thị data của p
        p = p->next; // cập nhật p
    }
}
```

Nội dung tiếp theo

Chèn node vào danh sách liên kết đôi