

LAB 1

TỔNG KẾT LÝ THUYẾT VÀ BÀI TẬP DANH SÁCH LIÊN KẾT ĐƠN

1. Tổ chức danh sách

Danh sách liên kết bao gồm các phần tử. Mỗi phần tử của danh sách đơn là một cấu trúc chứa 2 thông tin :

- Thành phần dữ liệu: lưu trữ các thông tin về bản thân phần tử .
- Thành phần nối liên kết: lưu trữ địa chỉ của phần tử kế tiếp trong danh sách, hoặc lưu trữ giá trị NULL nếu là phần tử cuối danh sách.

Ta có định nghĩa tổng quát

```
struct tagNode  
{  
    Data Info;           // Data là kiểu đã định nghĩa trước  
    struct tagNode* pNext; // con trỏ chỉ đến cấu trúc node  
};  
  
typedef struct tagNode NODE;
```

Ví dụ : Định nghĩa danh sách đơn lưu trữ hồ sơ sinh viên:

```
struct SinhVien //Data  
{  
    char Ten[30];
```

```

    int    MaSV;

};

typedef struct SinhVien SV;

struct SinhvienNode
{
    SV  Info;

    struct SinhvienNode* pNext;

};

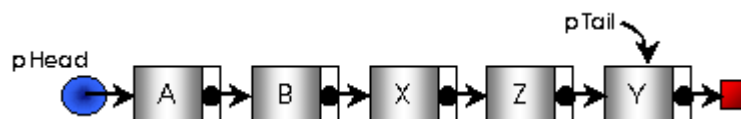
typedef struct SinhvienNode SVNode;

```

Các phần tử trong danh sách sẽ được cấp phát động. Biết phần tử đầu tiên ta sẽ truy xuất được các phần tử tiếp theo. Thường sử dụng con trỏ Head để lưu trữ địa chỉ đầu tiên của danh sách. Ta có khai báo: `NODE *pHead;`

Để quản lý địa chỉ cuối cùng trong danh sách ta dùng con trỏ TAIL. Khai báo như sau: `NODE *pTail;`

VD:



```

struct tagList
{
    NODE* pHead;

```

```
        NODE* pTail;

};

typedef struct tagList LIST;
```

2. Các thao tác cơ bản trên danh sách đơn

Giả sử có các định nghĩa:

```
struct tagNode

{

    Data Info;

    struct tagNode* pNext;

};

typedef tagNode NODE;

typedef struct tagList

{

    NODE* pHead;

    NODE* pTail;

};

typedef tagList LIST;

NODE    *new_ele // giữ địa chỉ của một phần tử mới được tạo

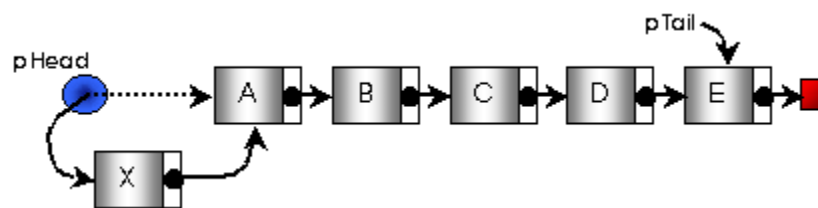
Data x;    // lưu thông tin về một phần tử sẽ được tạo
```

LIST lst; // lưu trữ địa chỉ đầu, địa chỉ cuối của danh sách liên kết

2.1 Chèn một phần tử vào danh sách:

Có 3 loại thao tác chèn new_ele vào xâu:

Cách 1: Chèn vào đầu danh sách



Thuật toán :

Bắt đầu:

Nếu Danh sách rỗng Thì

B11 : pHead = new_ele;

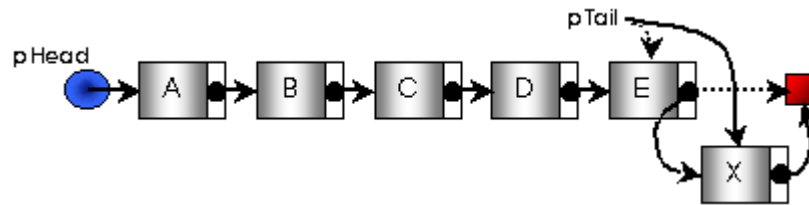
B12 : pTail = pHead;

Ngược lại

B21 : new_ele -> pNext = pHead;

B22 : pHead = new_ele ;

Cách 2: Chèn vào cuối danh sách



Thuật toán :

Bắt đầu :

Nếu Danh sách rỗng thì

B11 : $pHead = new_element;$

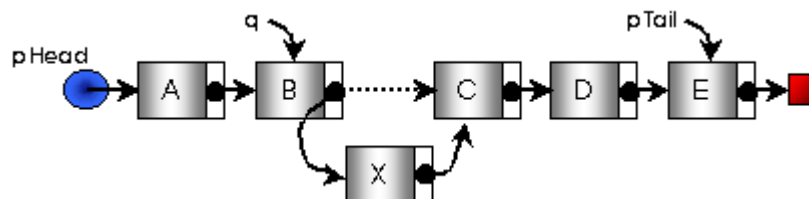
B12 : $pTail = pHead;$

Ngược lại

B21 : $pTail \rightarrow pNext = new_ele;$

B22 : $pTail = new_ele ;$

Cách 3 : Chèn vào danh sách sau một phần tử q



Thuật toán :

Bắt đầu :

Nếu ($q \neq \text{NULL}$) thì

B1 : $\text{new_ele} \rightarrow \text{pNext} = q \rightarrow \text{pNext};$

B2 : $q \rightarrow \text{pNext} = \text{new_ele};$

2.2 Tìm một phần tử trong danh sách đơn

Thuật toán :

Bước 1:

$p = \text{pHead};$ //Cho p trở đến phần tử đầu danh sách

Bước 2:

Trong khi ($p \neq \text{NULL}$) và ($p \rightarrow \text{Info} \neq k$) thực hiện:

$p := p \rightarrow \text{pNext};$ // Cho p trở tới phần tử kế

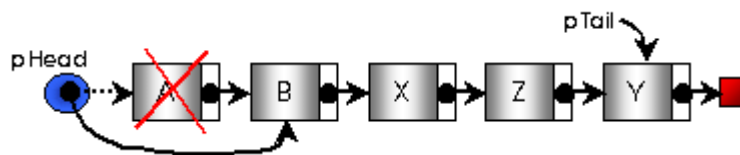
Bước 3:

Nếu $p \neq \text{NULL}$ thì p trở tới phần tử cần tìm

Ngược lại: không có phần tử cần tìm.

2.3 Hủy một phần tử khỏi danh sách

Hủy phần tử đầu xâu:



Thuật toán :

Bắt đầu:

Nếu (pHead != NULL) thì

B1: p = pHead; // p là phần tử cần hủy

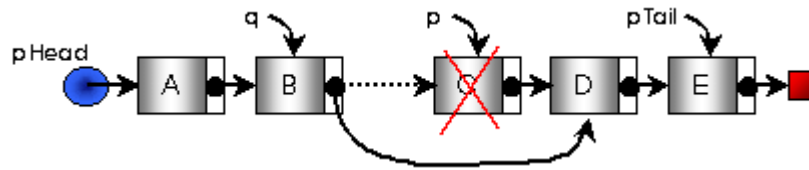
B2:

B21 : pHead = pHead->pNext; // tách p ra khỏi xâu

B22 : free(p); // Hủy biến động do p trỏ đến

B3: Nếu pHead=NULL thì pTail = NULL; //Xâu rỗng

Hủy một phần tử đứng sau phần tử q



Thuật toán :

Bắt đầu:

Nếu (q!= NULL) thì

B1: p = q->Next; // p là phần tử cần hủy

B2: Nếu (p != NULL) thì // q không phải là cuối xâu

B21 : q->Next = p->Next; // tách p ra khỏi xâu

B22 : free(p); // Hủy biến động do p trỏ đến

Hủy 1 phần tử có khoá k

Thuật toán :

Bước 1:

Tìm phần tử p có khóa k và phần tử q đứng trước nó

Bước 2:

Nếu (p!= NULL) thì // tìm thấy k

Hủy p ra khỏi cấu trúc tự hủy phần tử sau q;

Ngược lại

Báo không có k;

2.4 Thăm các nút trên danh sách

- Đếm các phần tử của danh sách,
- Tìm tất cả các phần tử thỏa điều kiện,
- Hủy toàn bộ danh sách (và giải phóng bộ nhớ)

Thuật toán xử lý các nút trên danh sách:

Bước 1:

p = pHead; //Cho p trở đến phần tử đầu danh sách

Bước 2:

Trong khi (Danh sách chưa hết) thực hiện

B21 : Xử lý phần tử p;

B22 : $p := p \rightarrow pNext$; // Cho p trở tới phần tử kế

Thuật toán hủy toàn bộ danh sách:

Bước 1:

Trong khi (Danh sách chưa hết) thực hiện

B11:

$p = pHead$;

$pHead := pHead \rightarrow pNext$; // Cho p trở tới phần tử kế

B12:

Hủy p;

Bước 2:

$Tail = NULL$; //Bảo đảm tính nhất quán khi xâu rỗng

3. BÀI TẬP:

Viết chương trình với menu tương ứng cho phép người dùng thực hiện các thao tác sau trên danh sách liên kết đơn, biết rằng thành phần dữ liệu của một nút là một số nguyên dương.

- a. Nhập vào và xuất ra n phần tử vào danh sách bằng cách thêm vào đầu danh sách và in ra danh sách vừa nhập.
- b. Nhập vào và xuất ra n phần tử vào danh sách bằng cách thêm vào cuối danh sách và in ra danh sách vừa nhập.
- c. In ra danh sách theo chiều cuối mà không dùng mảng phụ hoặc danh sách phụ VD: danh sách 3->4->5->6->NULL in ra là 6-5-4-3
- d. Đếm số nút trong danh sách liên kết đơn.
- e. Tìm một nút có thành phần dữ liệu bằng X được nhập vào từ bàn phím.
- f. Xoá nút đầu danh sách.
- g. Xoá nút cuối danh sách.
- h. Hãy viết xoá một nút có trường dữ liệu bằng X, nếu không có thì thông báo không có.

- i. Hãy chèn 1 nút có trường dữ liệu bằng X (X nhập từ bàn phím) vào vị trí thứ n được nhập vào từ bàn phím.