

Lab 03

Danh sách liên kết đơn: Các thao tác cơ bản

A. Mục tiêu

- Tìm hiểu và cài đặt danh sách liên kết đơn
- Thực hiện các thao tác cơ bản trên DSLK đơn.
- Thực hiện các ứng dụng liên quan.

B. Yêu cầu

- Sinh viên thực tập trong phòng lab.
- Hình thức thu bài: Giáo viên sẽ thu bài (tập tin nén trên) qua hệ thống thu bài trực tuyến tại phòng lab sau khi hết thời gian qui định.

C. Ôn tập lý thuyết

I. Tổ chức kiểu dữ liệu DSLK đơn

- DSLK đơn liên kết các phần tử hay còn gọi là nút (là các biến động) của DS theo một chiều đi từ phần tử đầu đến phần tử cuối, trong đó phần tử trước chứa thành phần liên kết (là con trỏ) chứa địa chỉ của phần tử kế sau. DSLK đơn được quản lý bằng 2 con trỏ: một chứa địa chỉ của phần tử đầu, một chứa địa chỉ phần tử cuối.

Ta sẽ đặt tên Kiểu DSLK đơn (khi cài đặt) là : LIST

- Phần tử (biến động) của DS là một cấu trúc có 2 thành phần: Một thành phần chứa dữ liệu của phần tử, thành phần còn lại là con trỏ, được dùng để chứa địa chỉ của cấu trúc cùng kiểu (Cấu trúc tự trỏ)

Ta sẽ đặt tên Kiểu phần tử (khi cài đặt) là : NODE

- Thành phần dữ liệu của phần tử (nút) của DS có kiểu dữ liệu là các kiểu đơn (ký tự, nguyên, thực) hay là các kiểu có cấu trúc tự đặt . . .

Đặt tên Kiểu của thành phần dữ liệu của phần tử trong DS là : data

(data thay đổi tùy theo các ứng dụng)

data → NODE → LIST

II. Cài đặt kiểu dữ liệu DSLK đơn

- Kiểu của thành phần dữ liệu trong DS:
(Giả sử là kiểu int)

typedef int data;

- Kiểu Phần tử (Nút) của DS:

➤ Định nghĩa ban đầu

struct tagNode

{

data

info;

```
tagNode* pNext;
```

```
};
```

➤ Đổi lại tên:

```
typedef tagNode NODE;
```

- Kiểu DSLK đơn:

```
struct LIST
```

```
{
```

```
    NODE* pHead; //Con trỏ lưu địa chỉ phần tử đầu DSLK
```

```
    NODE* pTail; //Con trỏ lưu địa chỉ phần tử cuối DSLK
```

```
};
```

III. Các thao tác thường dùng trên DSLK đơn (trên các nút, trên danh sách)

(Giả sử ta đã cài đặt kiểu DSLK đơn như trên)

1. Tạo nút mới: **GetNode (x)** :

- Chức năng: Tạo ra một phần tử (nút) của DSLK với thành phần dữ liệu là x (Thành phần liên kết của nút mới là con trỏ có giá trị NULL)
- Input : x
- Output : trả về con trỏ - lưu trữ địa chỉ của pt vừa tạo (nếu tạo thành công) - có giá trị NULL (nếu ngược lại).
- Nguyên mẫu của hàm: NODE* GetNode(Data x);

2. Khởi tạo DS rỗng: **CreatList (l)**

- Chức năng: Tạo ra một DSLK l rỗng.
- Input : l
- Output : l
- Nguyên mẫu của hàm: void CreatList(LIST &l);

3. Kiểm tra DSLK l có rỗng: **IsEmpty(l)**

- Chức năng: Kiểm tra DSLK l có rỗng rỗng?
- Input : l
- Output : 1; nếu l rỗng
0; ngược lại

- Nguyên mẫu của hàm: int IsEmpty(LIST l);

4. Duyệt danh sách:

- Chức năng: Duyệt danh sách từ đầu DS để xử lý dữ liệu của nút (có thể là xuất dữ liệu ra màn hình, đếm số nút, ...)
- Input : l
- Nguyên mẫu của hàm: void ProcessList (LIST l);

5. Tìm nút có info là x:

- Chức năng: Tìm trong DS có nút chứa thành phần Info là x?

- Input : l, x
 - Output : p (con trỏ p chứa nút đầu tiên có info là x); nếu có NULL; ngược lại
 - Nguyên mẫu của hàm: `NODE *Search(LIST l, Data x);`
6. Chèn nút (đã có) vào đầu DSLK đơn:
- Chức năng: Chèn một nút (đã có trước) vào đầu DS (Chú ý: thành phần liên kết của nút này là con trỏ có giá trị NULL)
 - Input : l, new_ele
 - Output : l (Nút đầu là new_ele)
 - Nguyên mẫu của hàm: `void AddFirst(LIST &l, NODE* new_ele);`
7. Chèn một giá trị dữ liệu vào đầu DSLK đơn:
- Chức năng: Tạo trước nút new_ele có info là x, con trỏ liên kết có giá trị NULL; sau đó Chèn nút này vào đầu DS.
 - Input : l, x (có kiểu data)
 - Output : - l (Nút đầu có info là x kiểu data)
- Con trỏ chứa địa chỉ của nút vừa tạo.
 - Nguyên mẫu của hàm: `NODE* InsertHead(LIST &l, data x);`
`void InsertHead(LIST &l, data x);`
8. Chèn nút (đã có) vào Cuối DSLK đơn:
- Chức năng: Chèn một nút (đã có trước) vào cuối DS (Chú ý: thành phần liên kết của nút này là con trỏ có giá trị NULL)
 - Input : l, new_ele
 - Output : l (Nút cuối là new_ele)
 - Nguyên mẫu của hàm: `void AddTail(LIST &l, NODE *new_ele);`
9. Chèn một giá trị dữ liệu vào cuối DSLK đơn:
- Chức năng: Tạo nút new_ele có info là x, con trỏ liên kết có giá trị NULL; sau đó chèn nút này vào cuối DS.
 - Input : l, x (có kiểu data)
 - Output : - l (Nút cuối có info là x kiểu data)
- Con trỏ chứa địa chỉ của nút vừa tạo.
 - Nguyên mẫu của hàm: `NODE* InsertTail (LIST &l, data x);`
`void InsertTail (LIST &l, data x);`
10. Chèn một nút (chưa có trước) vào sau nút do con trỏ q trỏ tới :
- Chức năng: Tạo nút new_ele có info là x, con trỏ liên kết có giá trị NULL; sau đó chèn nút này vào sau nút do con trỏ q trỏ tới.
 - Input : l, q
 - Output : - l (Nút cuối có info là x kiểu data)
- Con trỏ chứa địa chỉ của nút vừa tạo.
 - Nguyên mẫu của hàm: `void InsertAfter(LIST &l, NODE *q, data x);`

11. Hủy nút đầu ra khỏi DSLK đơn:
 - Chức năng: Hủy nút đầu ra khỏi DSLK đơn
 - Input : l
 - Output : l (bớt nút đầu của l input)
 - Nguyên mẫu của hàm: void RemoveHead(LIST &l);
12. Hủy nút ở vị trí sau nút do con trỏ q trỏ tới :
 - Chức năng: Hủy nút sau nút có vị trí do con trỏ q trỏ tới.
 - Input : l, q
 - Output : l (bớt 1 nút)
 - Nguyên mẫu của hàm: void RemoveAfter (LIST &l, NODE *q);
13. Hủy nút có thành phần info là x :
 - Chức năng: Hủy nút có info là x.
 - Input : l, x
 - Output : l; Nếu có nút
0; Nếu ngược lại
 - Nguyên mẫu của hàm: int RemoveNode(LIST &l, Data x);
14. Hủy toàn bộ danh sách
 - Chức năng: Hủy toàn bộ danh sách
 - Input : l
 - Output : l rỗng
 - Nguyên mẫu của hàm: void RemoveList(LIST &l);

D. Bài tập thực hành

Bài tập 1: Sinh viên tạo project với tên: **MSSV_Lab03_Bai1**. Có dữ liệu sử dụng để kiểm tra gồm 2 file sau:

- “Test1.txt” :

9 7 -1 7 -1 0 7 -8 1 7 1 0

- “Test2.txt” :

0 2 -9 7 6 -3 6 7 6 0

Viết chương trình thực hiện tùy chọn các thao tác cơ bản trên danh sách liên kết đơn, dữ liệu của các nút trong danh sách là số nguyên sau:

0. Thoát.
1. Nhập danh sách liên kết file.
2. Xuất dữ liệu.
3. Tính tổng các phần tử trong danh sách.
4. Tìm phần tử lớn nhất trong danh sách.
5. Tìm X – Trả về vị trí nút đầu tiên nếu có.
6. Tìm X – Trả về nút cuối cùng nếu có.
7. Chèn X vào đầu danh sách.

8. Chèn X vào cuối danh sách.
9. Chèn X sau nút có dữ liệu Y trong danh sách.
10. Hủy nút đầu
11. Hủy nút cuối
12. Hủy nút đầu tiên có dữ liệu X.
13. Hủy toàn bộ danh sách.
14. Sắp tăng danh sách bằng phương pháp chọn trực tiếp (hoán đổi dữ liệu).
15. Sắp tăng danh sách bằng phương pháp chọn trực tiếp (hoán đổi liên kết).

Bài tập 2: Sinh viên tạo project với tên: **MSSV_Lab03_Bai2.**

Giả sử có một danh sách nhân viên, mỗi nhân viên được lưu trữ các thông tin:

- Mã nhân viên, // Một chuỗi có 7 ký tự, không có ký tự trắng
- Họ nhân viên, // một chuỗi có không quá 10 ký tự
- Tên lót nhân viên, // một chuỗi có không quá 10 ký tự
- Tên nhân viên, // một chuỗi có không quá 10 ký tự
- Ngày tháng năm sinh của nhân viên, dạng : dd/mm/yyyy
- Địa chỉ, // một chuỗi có không quá 15 ký tự
- Lương, // Một số thực dương làm tròn, không quá 7 chữ số

Danh sách nhân viên được cho trong tập tin văn bản :

- testNV.txt :

Ma NV	Ho	tLot	Ten	NTNSinh	Dia Chi	Luong
2523480	Vo	Minh	Duong	3 /1 /1980	Da_Lat	15000543
1134547	Hoang	Hoa	Tra	12/12/1987	Khanh_Hoa	18183210
2130007	Truong	Thanh	Hang	2 /2 /1980	Phu_Yen	15000543
1212345	Nguyen	Thi	Hang	10/8 /1987	Binh_Dinh	20232343
2223453	Le	Van	Trong	12/5 /1991	Hue	18183210
2003205	Van	Minh	Duong	10/10/1988	Da_Lat	18183210
1500348	Tran	Hoang	Hoa	15/10/1996	Binh_Dinh	18183210
2134007	Ly	Thi	Hang	18/7 /1988	Khanh_Hoa	20232343
2030007	Dinh	Thai	Duong	21/12/1980	Da_Lat	15000543
2011453	Le	Van	Tam	25/6 /1988	Hue	18183210
2022205	Trinh	Hoai	Duong	19/10/1991	Da_Lat	18183210
1652345	Nguyen	Minh	Tam	22/9 /1980	Binh_Dinh	20232343

- testnv2.txt :

Ma NV	Ho	tLot	Ten	NTNSinh	Dia Chi	Luong
LD12045	Nguyen	Tuan	Vo	1/1/1986	Lam_Dong	2500000
LD13210	Truong	Thi	Hoa	12/12/1985	Ninh_Thuan	3700000
LD13452	Tran	Ngoc	Ninh	2/2/1974	Khanh_Hoa	8000000
LD14432	Nguyen	_	Vo	10/8/1975	Phu_Yen	15000000
LD15332	Le	Thi	Lieu	12/5/1974	Binh_Dinh	12000000
LD22032	Van	Thi	Hoa	10/10/1984	Lam_Dong	6000000
LD22052	Vo	Ngoc	Hoa	1/10/1984	Lam_Dong	7000000
LD22140	Nguyen	Van	Vu	1/7/1990	Binh_Dinh	6200000
LD22145	Nguyen	Thi	Vo	10/6/1986	Khanh_Hoa	8000000
LD23045	Tran	Trong	Hieu	10/10/1991	Ha_Noi	15000000
LD24042	Ly	Van	Hoa	12/9/1983	Ninh_Thuan	30000000
LD30432	Le	_	Vo	12/12/1975	Lam_Dong	12000000

Viết chương trình quản lý nhân viên sử dụng cấu trúc dữ liệu danh sách liên kết đơn, dữ liệu của các nút trong danh sách là nhân viên:

0. Thoát.
1. Nhập danh sách nhân viên từ file.
2. Xuất danh sách nhân viên.
3. Tìm theo tên nhân viên – trả về nút cuối cùng (nếu có).
4. Tìm nhân viên có lương cao nhất.
5. Đếm số nhân viên có tiền lương $\geq x$.
6. Hủy toàn bộ nhân viên có địa chỉ x trong danh sách.
7. Hủy nhân viên đầu tiên có tên là x.
8. Sắp giảm theo **Năm sinh nhân viên** bằng phương pháp nổi bọt (hoán đổi dữ liệu).
9. Sắp giảm theo **Mã nhân viên** bằng phương pháp chèn trực tiếp (hoán đổi dữ liệu).
10. Sắp tăng theo **Lương nhân viên** bằng phương pháp chọn trực tiếp (thay đổi liên kết).
11. Sắp tăng theo **Tên nhân viên** bằng phương pháp đổi chỗ trực tiếp (thay đổi liên kết).