

Chương 9

Kỹ thuật duyệt mảng sử dụng con trỏ

- Các phép toán trên địa chỉ, con trỏ
- Duyệt địa chỉ phần tử của mảng
- Hàm trả về địa chỉ

TOÁN TỬ *

Con trỏ vô kiểu

```
1  /*giasu.c*/
2  #include <stdio.h>
3
4  void main()
5  {
6      int i;
7      float b[10];
8      void *z;           //z la bien con tro vo kieu
9
10     z = &i;
11     *(int *)z = 8;      // tuong duong lenh i = 8;
12
13     z = &b[i];
14     *(float *)z = 10.11; // tuong duong lenh b[i] = 10.11
15                          // tuong duong lenh b[8] = 10.11
16     printf("\n b[%d] = %f\n",i,b[i]); // b[8] = 10.110000
17
18 }
19
20
```

CON TRỎ

Thao tác trên địa chỉ bộ nhớ

Cộng trừ với số nguyên: **ptr+n** biểu diễn (giá trị của **ptr**)+n x **sizeof(T)**

Phép trừ hai con trỏ: **ptr1-ptr2** --> (giá trị **ptr1** - giá trị **ptr2**) / **sizeof(T)**
với **ptr1**, **ptr2** là con trỏ cùng kiểu **T** nào đó.

=> **sizeof(void)** không xác định do đó, không thể trừ 2 con trỏ kiểu **void**

Quy ước ký pháp.

+ ***(ctr+sn)** tương đương với **biến** tại địa chỉ **ctr+sn**

+ Sử dụng ký hiệu mảng, **ctr[sn]** truy xuất tương đương **biến** trên.

Trong đó:

+ **biến** được hiểu là có kiểu phù hợp với con trỏ.

+ **sn** là số nguyên, **ctr** là con trỏ hay biến địa chỉ.

+ giả thiết các địa chỉ ở trên đều hợp lệ.

CON TRỎ

Thao tác trên địa chỉ bộ nhớ

```
1  #include <stdio.h>
2  void main()
3  {
4      int *x, y;
5      float z[4], *t;
6      y=5; x = &y; x++
7
8      printf("\n kích thước (con trỏ) x %d",sizeof(x));    //8
9      printf("\n kích thước (địa chỉ) &z[0] %d",sizeof(&z[0]));//8
10     printf("\n kích thước (phần tử) z[0] %d",sizeof(z[0])); //4
1     printf("\n kích thước (tên mảng) z %d",sizeof(z));    //16
2     printf("\n giá trị (con trỏ) x-&y %d",x-&y); //1
3     printf("\n giá trị (con trỏ) x %d",x);    //1781319664
4     printf("\n giá trị (địa chỉ) &y %d",&y);    //1781319660
5     printf("\n giá trị (con trỏ) x+1 %d",x+1); //1781319668
6     printf("\n giá trị (con trỏ) t %d",t);    //1781319904
7     printf("\n giá trị (tên mảng) z %d",z);    //1781319632
8     printf("\n giá trị (địa chỉ) &z[1] %d",&z[1]);//1781319636
9     printf("\n giá trị (địa chỉ) &z[2] %d",&z[2]);//1781319640
20 }
```

DUYỆT PHẦN TỬ MẢNG

Thảo luận: Chỉ số

```
1  #include <stdio.h>
2  int chieudaickt(char *ckt);
3
4  void main()
5  {
6      char *hovaten = "Nguyen Van Abc";
7      int n;
8      n = chieudaickt(hovaten);
9      printf("\n \"%s\" bao gom %d ky tu",hovaten,n);
10 }
11 int chieudaickt(char *ckt) {
12     int cd = 0;
13     while (ckt[cd]!='\0') cd++;
14     return cd;
15 }
16
17
18
19
20
```

DUYỆT PHẦN TỬ MẢNG

Địa chỉ

```
1  #include <stdio.h>
2  int chieudaickt(char *ckt);
3  int chieudaickt2(char *ckt);
4  void main()
5  {
6      char *hovaten = "Nguyen Van Abc";
7      int n;
8      n = chieudaickt(hovaten);
9      printf("\n \"%s\" bao gom %d ky tu",hovaten,n);
10 }
11 int chieudaickt(char *ckt) {
12     int cd = 0;
13     while (ckt[cd]!='\0') cd++;
14     return cd;
15 }
16 int chieudaickt2(char *ckt) {
17     int cd = 0;
18     while (*ckt++) cd++;
19     //toan tu * va ++ co cung uu tien, thu tu: trai <-- phai
20     return cd;
21 }
```

DUYỆT VÀ THAO TÁC TRÊN ĐỊA CHỈ

```
1  #include <stdio.h>
2  int chieudaickt3(char *ckt);
3  int chieudaickt2(char *ckt);
4  void main()
5  {
6      char *hovaten = "Nguyen Van Abc";
7      int n;
8      n = chieudaickt3(hovaten);
9      printf("\n \"%s\" bao gom %d ky tu",hovaten,n);
10 }
11 int chieudaickt3(char *ckt) {
12     char *ptr = ckt;
13     while (*ptr++); //it hon mot nua phep toan
14     return (ptr-ckt-1);//so voi chieudaickt1,2
15 }
16 int chieudaickt2(char *ckt) {
17     int cd = 0;
18     while (*ckt++) cd++;
19     //toan tu * va ++ co cung uu tien, thu tu: trai <-- phai
20     return cd;
21 }
```

HÀM TRẢ VỀ PHẦN TỬ CỦA MẢNG

Thảo luận: Chỉ số

```
1  #include <stdio.h>
2  int chisoNN(int mang[], int nptu);
3  int *controNN(int mang[], int nptu);
4  void main()
5  {
6      int i, n, *ct;
7      int daysn[] = {2, 5, 1, 6, 7};
8      n = sizeof(daysn)/sizeof(daysn[0]);
9      i = chisoNN(daysn,n);
10     ct = controNN(daysn,n);
11     printf("\nso nho nhat %d ?? %d",daysn[i],*ct);
12     printf("\ntai dia chi: 0x%x ?? 0x%x",&daysn[i],ct);
13 }
14 int chisoNN(int src[], int n) {
15     int i = 1; cs = 0;
16     while (i<n){
17         if (src[i] < src[cs]) cs = i;
18         i++;
19     }
20     return cs;
21 }
```


HÀM TRẢ VỀ PHẦN TỬ CỦA MẢNG

Địa chỉ

```

1  #include <stdio.h>
2  int chisoNN(int mang[], int nptu);
3  int *controNN(int mang[], int nptu);
4  void main()
5  {
6      int i, n, *ct;
7      int daysn[] = {2, 5, 1, 6, 7};
8      n = sizeof(daysn)/sizeof(daysn[0]);
9      i = chisoNN(daysn,n);
10     ct = controNN(daysn,n);
11     printf("\nso nho nhat %d ?? %d",daysn[i],*ct);
12     printf("\ntai dia chi: 0x%x ?? 0x%x",&daysn[i],ct);
13 }
14 int *controNN(int src[], int n) {
15     int i = 1; cs = 0;
16     while (i<n){
17         if (src[i] < src[cs]) cs = i;
18         i++;
19     }
20     return &src[cs];

```

HÀM DUYỆT VÀ TRẢ VỀ ĐỊA CHỈ

```
1  #include <stdio.h>
2  int *contross(int [], int , int);
3  int *chepmang(int *, int [], int );
4  void xuatmang(int [], int );
5  void main()
6  {
7      int i, n, gtktra = 7, *ct, nct;
8      int daysn[] = {2, 5, 1, 6, 7, 9, 20};
9      n = sizeof(daysn)/sizeof(daysn[0]);
10     xuatmang(daysn,n);
11     while(ct=contross(daysn,n,gtktra)){
12         nct = n-(ct-daysn);
13         ct = chepmang(ct, ct+1,nct);n--;
14         xuatmang(daysn,n);
15     }
16     void xuatmang(int src[], int n) {
17         int i=0;
18         printf("\ndaysn = {");
19         while (i<n) printf("%d,",src[i++]);
20         printf("nothing};\n");
21     }
```

HÀM DUYỆT VÀ TRẢ VỀ ĐỊA CHỈ

Minh họa: Khử phần tử mảng

```
1  #include <stdio.h>
2  void xuatmang(int [], int);
3  int *chepmang(int *, int [], int);
4  int *contross(int [], int , int);
5  int *khuptumang(int [], int *, int);
6  void main()
7  {
8      int n, gtktra = 5;
9      int daysn[] = {2, 5, 1, 6, 7, 9, 20};
10     n = sizeof(daysn)/sizeof(daysn[0]);
11     xuatmang(daysn,n);
12     khuptumang(daysn, &n, gtktra);
13     xuatmang(daysn,n);
14 }
15 int *khuptumang(int src[], int *n, int gtss) {
16     int *ctr, nct;
17     while (ctr = contross(src,*n,gtss)) {
18         nct = ((*n)--)-(ctr-src);
19         chepmang(ctr, ctr+1, nct);
20     }
21 }
```

XOÁ PHẦN TỬ DÃY KÝ TỰ

Minh họa

```
1  #include <stdio.h>
2  #include <string.h>
3  void khukhoangtrang(char *, char *);
4  void kktctro(char *src);
5
6  void main()
7  {
8      char ckt[] = "Nguy en- Va n -ABC";
9      char ckt2[] = "Nguy en- Va n -ABC";
10     khukhoangtrang(ckt,ckt);
1     puts(ckt);
2     kktctro(ckt2);
3     puts(ckt2);
4 }
5
6 void kktctro(char *src) {
7     char *ctr;
8     while (ctr = strchr(src,' ')) strcpy(ctr,ctr+1);
9 }
20
```

DẪY CON TRONG DẪY ĐÃ CHO

Minh họa: diễn giải hàm

```
1  int *contromcon(int src[], int n, int mss[], int nmc) {
2      int i=0, *a=src, *b, *c;
3      if (mss==NULL) return src;
4      while ((a++)-src<n-nmc) {
5          if(*a==*mss){
6              b = mss; c = a;
7              while (*b++==*c++);
8              if (b>mss+nmc) return a;
9          }
10     }
11     if (a-src>=n-nmc) return NULL;
12 }
```

3
4
5
6
7
8
9
20

DẪY CON TRONG DẪY ĐÃ CHO

Minh họa: Sử dụng hàm

```
1  #include <stdio.h>
2  int *contromcon(int [], int , int [], int );
3  void xuatmang(int [], int);
4
5  void main()
6  {
7      int n, nmc, *ct;
8      int daysn[] = {2, 5, 1, 2, 5, 1, 6, 7, 9, 20};
9      int dayc[] = {5,1,6};
10     int *test = NULL;
11     n = sizeof(daysn)/sizeof(daysn[0]);
12     nmc = sizeof(dayc)/sizeof(dayc[0]);
13     xuatmang(daysn,n);
14     xuatmang(dayc,nmc);
15     ct = contromcon(daysn,n,test,nmc);
16     ct = contromcon(daysn,n,dayc,nmc);
17     if (ct){ printf("\\n ct - daysn = %d",ct-daysn);
18             printf("\\n noi dung tai ct la %d",*ct);
19     }
20 }
```

ĐỆ QUY

Kỹ thuật lặp

```
1  int tong(int a[], int n) {  
2      if(n<=0) return 0;  
3      return a[n-1] + tong(a,n-1);  
4  }  
5  
6  int giaithua(int n) {  
7      if(n<=0) return 1;  
8      return n*giaithua(n-1);  
9  }  
10  
1  double tongeuler(int n, int k) {  
2      if(n<=0||k<1) return 0;  
3      return 1./pow(n,k)+tongeuler(n-1,k);  
4  }  
5  
6  
7  
8  
9  
20
```

ĐỆ QUY

Dùng mảng con

```
1 void sapxeptang(int a[], int n) {
2     int i, csn=0, csl=0, *tam;
3     for(i=0;i<n;i++) {
4         if(a[csn]>a[i]) csn = i;
5         if(a[csl]<a[i]) csl = i;
6     }
7     hoanvi(&a[0],&a[csn]);
8     if(!csl) csl = csn;
9     hoanvi(&a[n-1],&a[csl]);
10    if(n>3) {
1        tam = &a[1];
2        sapxeptang(tam,n-2);
3    }
4 }
5
6 void hoanvi(int *a, int *b) {
7     int tam = *a;
8     *a = *b;
9     *b = tam;
20 }
```