数据结构：线性表

结构体：

自定义的结构，类型。

struct student

{

int number;

char name[20];

float math;

};

typedef 取别名。

变量：定义 赋值（初始化） 引用

数组：定义 赋值（初始化） 引用

数据结构: 总结出来一些好的方便的，对于数据的存储管理。

内存四区：

全局区

代码区

栈区：系统来操控的。int a；

堆区：自己来操控 开辟内存 释放内存。

动态内存分配：

#include <stdlib.h>

void \*malloc(unsigned int size);

malloc开辟内存，void\*

int \*p = NULL;

p = (int\*)malloc(sizeof(int));

if (p == NULL)

{

printf("内存开辟失败！\n");

return -1;

}

\*p = 5;

printf("%d\n", \*p);

void free(void \*ptr); 释放内存

开辟的内存如果不用了 一定要释放。

free(p);

顺序表（数组）：

动态数组：

1维：int \*p = (int\*)malloc(sizeof(int)\* 10);

free(p);

二维：int\*\*p = (int \*\*)malloc(sizeof(int\*)\* 3);

for (int i = 0; i < 3; i++)

{

p[i] = (int \*)malloc(sizeof(int)\* 5);

}

for(int i=0;i<3;i++)

{

free(p[i]);

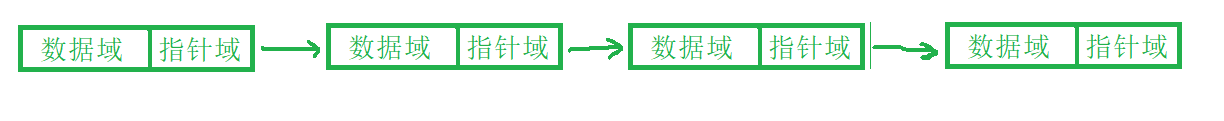
}

free(p);

其他的跟普通的数组操作一样。

relloc函数 自己去看。

链表：链式结构。



数据域：存放需要的有用的数据

指针域：用来连接各个节点，形成连式结构的指针。

节点：每一个单独数据内存空间叫做一个节点。

头指针：第一个节点的地址。

尾指针：最后一个节点的地址。

常见的链表有2种：

带空头节点的链表：头节点不存放数据，从头节点后面一个节点开始存放。

不带空头节点的链表：链表中每一个节点都用来存放数据。

带空头节点的链表：

//链表的初始化

Pt\_Stu ListInit()

{

//开辟一段空间 STU

Pt\_Stu plist = (Pt\_Stu)malloc(sizeof(STU));

//填充数据

plist->pnext = NULL;

return plist;

}

//增 删查改

void AddListNode(Pt\_Stu plist) //plist形参

{

//开辟空间

Pt\_Stu pnode = (Pt\_Stu)malloc(sizeof(STU));

//填充数据

printf("input number:");

scanf("%d", &pnode->number);

printf("input your name:");

scanf("%s", pnode->name);

printf("input score of math:");

scanf("%f", &pnode->math);

pnode->pnext = NULL;

////尾部增加

////plist 移动到最后一个节点。

//while (plist->pnext != NULL)

//{

// plist = plist->pnext;

//}

//plist->pnext = pnode;

//头部增加

pnode->pnext = plist->pnext; //只有头结点plist->pnext NULL

plist->pnext = pnode;

}

void DeleteNode(Pt\_Stu plist,int number)

{

Pt\_Stu p = plist->pnext;

Pt\_Stu p1= plist;

while (p != NULL)

{

if (p->number == number)

{

p1->pnext = p->pnext;

free(p);

p = p1->pnext;

}

else

{

p = p->pnext;

p1 = p1->pnext;

}

}

}

//遍历 查找

void TraverseList(Pt\_Stu plist)

{

//plist指向第一个存放数据的节点。

plist = plist->pnext;

while (plist != NULL)

{

if (plist->number==1)

{

plist->math = 600;

printf("number:%d\tname:%s\tscore:%.2f\n", plist->number, plist->name, plist->math);

}

plist = plist->pnext;

}

}

不带空头节点的链表：链表中每一个节点都用来存放数据。