[宏定义 1](#_Toc27582873)

[常量定义 1](#_Toc27582874)

[带参宏定义 2](#_Toc27582875)

[例1： 2](#_Toc27582876)

[例2： 2](#_Toc27582877)

[例3：准确表达如下： 2](#_Toc27582878)

[结构体指针 3](#_Toc27582879)

[指向结构体的指针 3](#_Toc27582880)

[1.1结构体占字节空间 3](#_Toc27582881)

[1.2 传统的指针结构体的指针，采用普通结构体变量过渡 3](#_Toc27582882)

[1.3采用动态内存分配的新方法 3](#_Toc27582883)

[链表（涉及结构体的指针） 4](#_Toc27582884)

[文件 4](#_Toc27582885)

[文件的基本操作 4](#_Toc27582886)

[向文件中写入字符 5](#_Toc27582887)

[从文件中读入字符到显示器上 5](#_Toc27582888)

[格式化读写函数 6](#_Toc27582889)

[例1：向文件写入内容 6](#_Toc27582890)

[例2：从文件中读出内容 6](#_Toc27582891)

## 宏定义

### 常量定义

#include<stdio.h>

#define PI 3.14//定义符号常量，以后一改全改

int main()

{float x=2;

printf("%.2f",PI\*2\*2);

}

### 带参宏定义

#### 例1：

#include<stdio.h>

#define SUB(a,b) a-b

int main()

{ int z;

z=SUB(5,3)-SUB(4,2);//**-4**

//带参宏，第一步，替换，第二，代入计算

//SUB(5,3) ===>5-3

//sub(4,2) ====>4-2

//上式变成5-3-4-2

**//普遍错误理解成#define SUB(a,b) (a-b)**

printf("%d",z);

}

#### 例2：

#include<stdio.h>

#define SUB(a,b) a-b

int main()

{ int z;

z=SUB(5,3)\*SUB(4,2);/**/-9**

printf("%d",z);

}

#### 例3：准确表达如下：

#include<stdio.h>

#define SUB(a,b) (a-b)

int main()

{ int z;

z=SUB(5,3)\*SUB(4,2);**//4**

printf("%d",z);

}

## 结构体指针

### 指向结构体的指针

#### 1.1结构体占字节空间

#include<stdio.h>

struct student

{ char name[20];

int age;

};

int main()

{printf("%d",sizeof(student));//24

}

#### 1.2 传统的指针结构体的指针，采用普通结构体变量过渡

#include<stdio.h>

struct student

{ char name[20];

int age;

};

int main()

{ struct student \*p,stu1;

p=&stu1;

scanf("%s%d",stu1.name,&stu1.age);//法一

//法二scanf("%s%d",(\*p).name,&(\*p).age);

//法三：因为(\*P).name<====等价写法====>p->name,于是，改进的法三如下：

//scanf("%s%d",p->name,&p->age)

}

#### 1.3采用动态内存分配的新方法

#include<stdio.h>

#include<stdlib.h>

struct student

{ char name[20];

int age;

};

int main()

{ struct student \*p;

p=(struct student \*)malloc(sizeof(struct student ));

scanf("%s%d",p->name,&p->age);

printf("姓名是%s,年龄是%d",p->name,p->age);

}

### 链表（涉及结构体的指针）

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct student

{ char name[20];

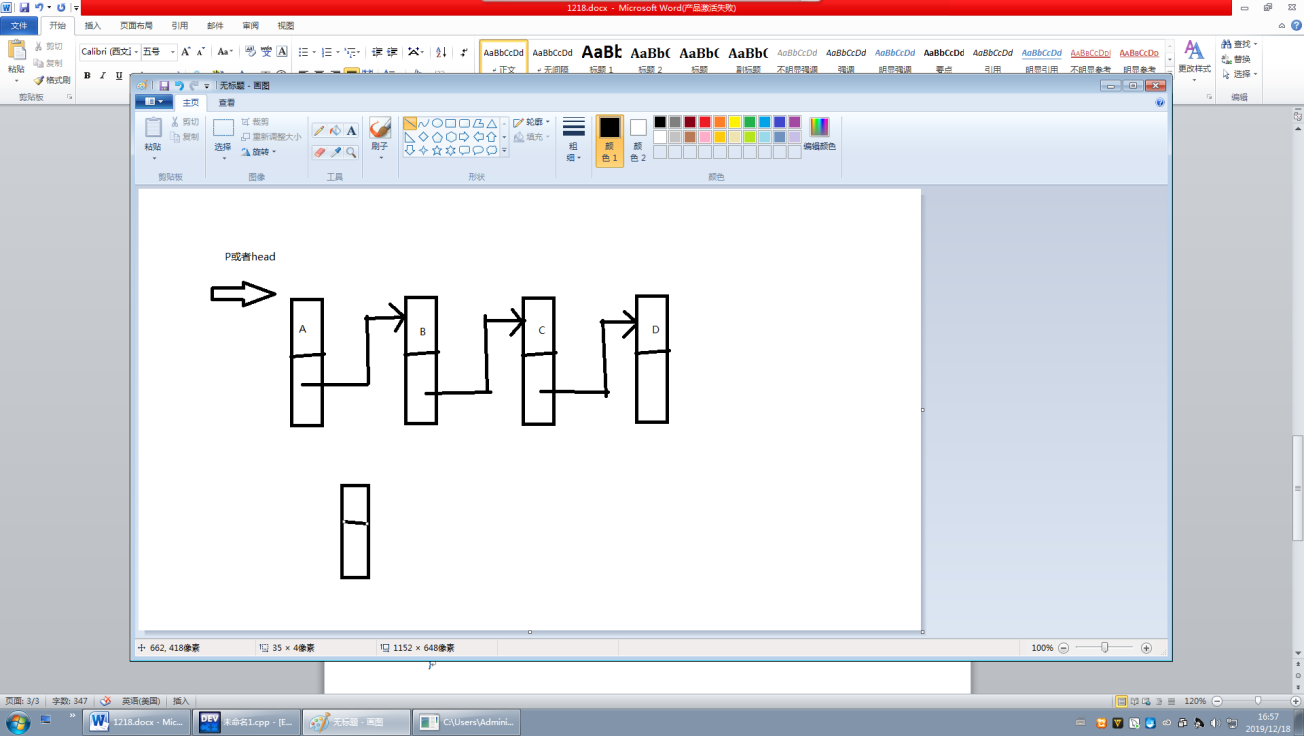
struct student \*next;

}a,b,c,d,\*p,\*head;

int main()

{ int i=0;

strcpy(a.name,"zhang");strcpy(b.name,"wang");strcpy(c.name,"cheng");strcpy(d.name,"li");

 p=head=&a;

a.next=&b; b.next=&c;c.next=&d;d.next=NULL;

while(p!=NULL)

{printf("%6s",p->name);p=p->next;

}

printf("\n");

p=head;

p->next=p->next->next;//功能实现了删除第二个结点

while(p!=NULL)

{printf("%6s",p->name);p=p->next;

}

printf("\n");

}

## 文件

### 文件的基本操作

#include<stdio.h>

#include<stdlib.h>

int main()

{FILE \*fp;

fp=fopen("kkrnew.txt","w");//若不存在，则新建，若存在，则覆盖

if (fp==NULL)

{printf("文件名不存在，或文件坏");exit(0);

}

printf("操作正常");

fclose(fp);

return 0;

}

### 向文件中写入字符

#include<stdio.h>

#include<stdlib.h>

int main()

{FILE \*fp;

char ch;

fp=fopen("kkrnew.txt","w");//若不存在，则新建，若存在，则覆盖

if (fp==NULL)

{printf("文件名不存在，或文件坏");exit(0);

}

printf("操作正常");

ch=getchar();

while(ch!='#')**//从键盘读入若干字符，只要不是#,就依次写入到文件**

{

**fputc(ch,fp);**

ch=getchar();

}

fclose(fp);

return 0;

}

### 从文件中读入字符到显示器上

#include<stdio.h>

#include<stdlib.h>

int main()

{FILE \*fp;

char ch;

fp=fopen("kkrnew.txt"**,"r");**

if (fp==NULL)

{printf("文件名不存在，或文件坏");exit(0);

}

ch=**fgetc(fp);**

while(ch!=**EOF**)

**//此句可以改写成while(!feof(fp))**

{printf("%c",ch);

ch=fgetc(fp);

}

fclose(fp);

return 0;

}

### 格式化读写函数

#### 例1：向文件写入内容

#include<stdio.h>

#include<stdlib.h>

int main()

{FILE \*fp;

char ch='a';

int x=4;

float y=2.1;

fp=fopen("kkrnew.txt","**w**");//若不存在，则新建，若存在，则覆盖

if (fp==NULL)

{printf("文件名不存在，或文件坏");exit(0);

}

**fprintf(fp,"ch=%c,x=%d,y=%4.2f",ch,x,y);**

printf("ch=%c,x=%d,y=%4.2f",ch,x,y);

fclose(fp);

return 0;

}

#### 例2：从文件中读出内容

#include<stdio.h>

#include<stdlib.h>

int main()

{FILE \*fp;

int x,y,z;

fp=fopen("kkrnew.txt",**"r"**);//若不存在，则新建，若存在，则覆盖

if (fp==NULL)

{printf("文件名不存在，或文件坏");exit(0);

}

**fscanf(fp,"%d%d%d",&x,&y,&z);**

printf("x=%d,y=%d,z=%d",x,y,z);

fclose(fp);

return 0;

}