# <stdlib.h>

strcpy()

malloc()//开辟堆空间

calloc()//空内存

连续内存空间都可以使用下标

realloc()//重新分配malloc或calloc内存空间，并不会清空内存空间，如果没有足够的内存空间，将重新申请空间（与malloc一样）

free()//释放堆空间

memset(\*p, 0, size) //重置指定内存的值

memcpy(void \*dest, const void \*src, size\_t n)

//拷贝指定大小的，与n有关，可以复制一块内存

strcpy() //以“/0”结尾，只能复制字符串

memmove（）dest,与src的位置重叠，可以移动，效率低（重新开启一个新空间）

memcmp()比较内存中空间是否相同（字节）相同返回0

FILE \* fopen(const char \* filename, “r”)

int fputc(int ch, FILE \*stream); //写入文件

int fgetc(FILE \*stream) //读取文件

fgetc(fp) != EOF //读取文件到结尾

fclose(FILE \* stream) //关闭

r读取，w创建新文件，a追加，创建新文件 ，+ 可读可写，b二进制读写（windows有效）

srand((unsigned int)time(NuLL));

操作二进制文件

fread()

fwrite()

strlen()

strcpy() strncpy() 复制

strcmp() strncmp() 字符串比较

strcat() strncat() 字符串追加

strchr() 查找字符第一次出现的位置

strstr() 查找字符串第一次出现的位置

strtok() 字符串切割,会破坏源字符串

atoi atol atof

# 字符串处理

#include<stdio.h>

对于未知大小数据，不能超过字符数组大小，不然缓存溢出

char ch[100]; //数组必须有大小

char \* gets(char \* ch)

从输入设备上获取y一个字符串，直到换行或\0结束

gets(ch) ; //将输入的字符串放到ch中,会接受任意大小字符，不安全

char \* p = gets(ch); //p与ch的结果是一样的

gets与scanf的区别：gets可以输入空格

sscanf(\*, “正则”, buf) //根据正则规则复制字符串

puts()

输出一个字符串并换行

puts(ch);//带换行符

int vlu = puts(ch) //vlu不为-1都为成功

char \* fgets(char \*s, int size, FILE \* stream)

始终会将\0加入

文本操作stdin标准输入库,size接收的大小

fgets(ch, 100, stdin)//将输入字符放到ch字符指针上

当输入的数据大于fgets的ch字符串指针能接收数据时，fgets将把多余的数据都扔掉，留下一个内存接收\0

int fputs(const char \*str, FILE \* stream)

stream固定为stdout

输出\0前的所有字符

## scanf格式化

scanf(“%[^\n]”, arr)

scanf(“%d %d”, &a, &b)

scanf(“%1d”, &a) //接收1位数据

scanf(“%1s%2s”, arr, arr2) //接收指定字符

scanf(“%\*s%d”, &a) //\*忽略字符

scanf(“%\*[123456]%c”, &a) // 忽略123456任意字符

scanf(“%\*[a-z]%c”, &a)

printf(“%.2f”, 3.145) //保留2位小数

printf(“%5d”, 12) //一共显示5位数

printf(“%4s”, “ab”) //显示4位数字

printf(“%.4s”, “ab”) //截取

## strlen

读取到\0结束

include<string.h>

char arr[] = “hello”

int len = strlen(arr) //字符串有效长度，忽略\0

与szieof区别是字符数组的长度

## char \* strcp(char \*dest, const char \*src)

从src复制到dest

目标字符串，源字符串，字符串长度,有限复制，不会复制\0

strncp(dest, src, 5)

## char \*strcat(char \* dest, const char \*src)

将src字符串追加到dest变量中

char arr1[] = “hello”;

char arr2[] = “word”;

strcat(arr1, arr2) //这里数组下标越界，需要修改arr1的数组大小

strncat()有限追加

strncat(arr1, arr2, 3)

有限追加会将\0加入

## strcmp(const char \*s1, const char \*s2)

比较字符串，比较的是ASCII吗，相等会返回0，不同的操作系统，返回值不同，比较\0之前的所有有效字符

strncmp(arr1, arr2, 2)//比较几个数之前的数

## strchr(const char \*ch, int c)

查找字符第一次出现的位置

char \*p = trrchr(arr, “l”); //p的结果为从l到\0的字符

失败返回NULL

strstr()//在一个字符串中查找另一个字符串

## strtok()

会修改原始字符串

char a[] = “www.baidu.com”;

char \*p = strtok(a, “\*”)

p // www

while(p!=Null)

{

p = strtok(NULL, “.”); //

}

## atoi()

将数转换为数字

atof()

atol()

字符串分割