动态内存空间申请函数malloc的使用

malloc的全称是memory allocation，中文叫动态内存分配，用于申请一块连续的指定大小的内存块区域以void\*类型返回分配的内存区域地址。其函数原型为：

extern void \*malloc(unsigned int num\_bytes);

数据参数num\_bytes表示需要动态申请的内存空间大小。

返回值类型void\*类型表示未确定类型的指针。C,C++规定，void\*类型可以通过类型转换强制转换为任何其它类型的指针。

调用malloc函数时，如果分配成功则返回指向被分配内存的指针(此存储区中的初始值不确定)，否则返回空指针NULL。当内存不再使用时，务必使用free()函数将内存块释放。free函数能释放某个动态分配的地址，表明不再使用这块动态分配的内存了，实现把之前动态申请的内存返还给系统。

malloc()函数被包含在malloc.h里面，使用malloc函数时，其输入参数通常会配合sizeof函数使用。sizeof函数用于指明一个数据类型所需的内存空间大小。

程序示例1：

……

typedef struct data\_type{

int age;

char name[20];

}data;

data \*bob=NULL;

bob=(data\*)malloc(sizeof(data));

if(bob!=NULL)

{

bob->age=22;

strcpy(bob->name,"Robert");

printf("%sis%dyearsold\n",bob->name,bob->age);

}

else

{

printf("mallocerror!\n");

exit(-1);

}

free(bob);

bob=NULL;

……

程序示例2：

#include "stdio.h"

#include "malloc.h"//malloc()函数被包含在malloc.h里面

int main(void)

{

char \*a = NULL;//声明一个指向a的char\*类型的指针

a = (char\*)malloc(100\*sizeof(char));//使用malloc分配内存的首地址，然后赋值给a

if(!a)//如果malloc失败，可以得到一些log

{

perror("malloc");

return -1;

}

sprintf(a,"%s","HelloWorld\n");//"HelloWorld\n"写入a指向的地址

printf("%s\n",a);//输出a中内容

free(a);//释放掉使用的内存地址

a=NULL;

return 0;

}