**cs:app学习笔记(1）：show-bytes**

2017年10月26日 01:11:12 [旭日之冕](https://me.csdn.net/wxh928408225) 阅读数：844

 版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/wxh928408225/article/details/78348538

* [showbytes](https://blog.csdn.net/wxh928408225/article/details/78348538#showbytes)
  + [代码总览](https://blog.csdn.net/wxh928408225/article/details/78348538#%E4%BB%A3%E7%A0%81%E6%80%BB%E8%A7%88)
  + [代码说明](https://blog.csdn.net/wxh928408225/article/details/78348538#%E4%BB%A3%E7%A0%81%E8%AF%B4%E6%98%8E)
  + [编译与结果显示](https://blog.csdn.net/wxh928408225/article/details/78348538#%E7%BC%96%E8%AF%91%E4%B8%8E%E7%BB%93%E6%9E%9C%E6%98%BE%E7%A4%BA)

**show—bytes**

**代码总览**

#include <stdio.h>

typedef unsigned char\* byte\_pointer;

void show\_bytes(byte\_pointer start, int len){

int i;

for (i = 0; i < len; i++)

printf ("%.2x", start[i]);

printf("\n");

}

void show\_int(int x){

show\_bytes((byte\_pointer)&x, sizeof (int));

}

void show\_float(float x){

show\_bytes((byte\_pointer)&x, sizeof (float));

}

void show\_pointer(void \*x){

show\_bytes((byte\_pointer)&x, sizeof (void \*));

}

void test\_show\_bytes(int val){

int ival = val;

float fval = (float) val;

int \*pval = &ival;

show\_int(ival);

show\_float(fval);

show\_pointer(pval);

}

int main(int argc, char const \*argv[])

{

int test = 12345;

test\_show\_bytes(test);

}

**代码说明**

typedef unsigned char\* byte\_pointer;

* 将unsigned char\*命名为byte\_pointer。
* 在cs:app中，提供的代码格式是如下：

typedef unsigned char \*byte\_pointer;

* 虽然对于程序不影响，但是我更喜欢我的写法，这样更像指向无符号字符型的指针。

void show\_bytes(byte\_pointer start, int len){

int i;

for (i = 0; i < len; i++)

printf ("%.2x", start[i]);

printf("\n");

}

* show-bytes 这个函数将传入无符号字符对象的指针，和该对象的字节数，然后通过在for循环中，使用printf ("%.2x", start[i]); 进行格式化输出。%.2x确保整数都以两位16进制数输出。
* start代表对象的首地址，start[i] 代表从start[0] 开始第i个位置处的字节。

void show\_int(int x){

show\_bytes((byte\_pointer)&x, sizeof (int));

}

void show\_float(float x){

show\_bytes((byte\_pointer)&x, sizeof (float));

}

void show\_pointer(void \*x){

show\_bytes((byte\_pointer)&x, sizeof (void \*));

}

* 这三个函数，将不同类型的对象的指针都强制转化成unsigned char\* 类型，并用sizeof 关键字表示该数据所用字节数。

void test\_show\_bytes(int val){

int ival = val;

float fval = (float) val;

int \*pval = &ival;

show\_int(ival);

show\_float(fval);

show\_pointer(pval);

}

int main(int argc, char const \*argv[])

{

int test = 12345;

test\_show\_bytes(test);

}

* 主函数与测试函数。

**编译与结果显示**

* 编译命令

gcc show-bytes.c -o main

./main

* 在linux 64位机的输出结果

39300000

00e44046

987de0a9fe7f0000

解释：

123456的16进制表示为 0x00003039，由于linux 64位采用小端存储，所以先输出低位39，然后再输出30。所以结果如上述表示。