2016汇编语言第1次作业 参考答案

1. 在VS下编制程序，分别打印32位和64位环境下，如下数据类型占用的字节数:

int，short int，long，long long，float，double，bool，char，指针p,

答案：32位：4、2、4、4、8、4、8、1、1、4

64位：4、2、4、4、8、4、8、1、1、**8**

注：long double也是8个字节

解释：本题目20分，每个1分

用i=sizeof(int) 、i=sizeof(&i) 等获取类型的占用空间。

在菜单或按钮上选择X86（win32）、X64可以切换环境

int i = sizeof(int);

i = sizeof(short int);

i = sizeof(long);

i = sizeof(long long); //同long

i = sizeof(float);

i = sizeof(double);

i = sizeof(bool); //不是4个，是1个字节

i = sizeof(char);

i = sizeof(&i);

i = sizeof(long double); //同double

2. 用你的信息替换下列变量的值，在VS（X86）下直接到内存查看其机器数(16进制表示)

int stno = 1140310188; //学号

int stnon = -1140310188; //-学号

char \*stnos = "1140310188"; //学号串

char \*stname = "学霸王"; //姓名

bool stsex = true; //男生=true 女生=false

char \*idno = "230106981221001"; //15位的身份证号

int sfzhi = 230106981221001; //15位的身份证号

long sfzhl = 230106981221001; //15位的身份证号

long long sfzhl = 230106981221001; //15位的身份证号

float sfzhf = 230106981221001; //15位的身份证号

double sfzhd = 230106981221001; //15位的身份证号

请分别写出你的学号、姓名、性别、身份证号；依次列出每个变量的内存起始地址、变量内容（16进制）。

请编写程序，依次用printf显示每个变量的值。(不用粘贴源程序)

答案：学生可能选择以上变量为全部为全局变量，或者全部为局部变量。都可以。但

全局全局变量的地址是逐个变量依次往后存放（字符串占4个字节，只存放其地址）

局部变量的地址是逐个变量依次往前存放（字符串占4个字节，只存放其地址）

下列变量的初值和内容与每个学生的学号与15位的身份证号对应。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 类型 | 变量名 | 初值 | 查看地址 | 内存起始地址 | 字节数 | 内容 |
| 1 | int | stno | 1140310188; | &stno | 0x01217020 | 4 | ac c0 f7 43 |
| 2 | int | stnon | -1140310188 | &stnon | 0x01217024 | 4 | 54 3f 08 bc |
| 3 | char \* | stnos | "1140310188" | stnos &stnos | 0x01215B30 0x01217028 | 11 4 | 31 31 34 30 33 31 30 31 38 38 00 |
| 4 | char \* | stname | "学霸王" | stname &stname | 0x01215B44 0x0121702C | 7 4 | d1 a7 b0 d4 cd f5 00 |
| 5 | bool | stsex | true | &stsex | 0x01217030 | 1 | 01 （由于内存对齐，也占4个字节） |
| 6 | char \* | idno | "230106981221001" | idno &idno | 0x01215BE4 0x01217034 | 16 4 | 32 33 30 31 30 36 39 38 31 32 32 31 30 30 31 00 |
| 7 | int | sfzhi | 230106981221001; | &sfzhi | 0x01217038 | 4 | 89 42 e0 f4 |
| 8 | long | sfzhl | 230106981221001; | &sfzhl | 0x0121703C | 4 | 89 42 e0 f4 |
| 9 | long long | sfzhll | 230106981221001; | &sfzhll | 0x01217040 | 8 | 89 42 e0 f4 47 d1 00 00 |
| 10 | float | sfzhf | 230106981221001; | &sfzhf | 0x01217048 | 4 | f5 47 51 57 |
| 11 | double | sfzhd | 230106981221001; | &sfzhd | 0x01217050 | 8 | 20 51 08 9c fe 28 ea 42 (内存对齐，地址从8的倍数开始) |
| 全局变量的地址是逐个变量依次往后存放（字符串占4个字节，只存放其地址） | | | | | | | |
|  |  |  |  |  |  |  |  |
| 局部变量的地址是逐个变量依次往前存放（字符串占4个字节，只存放其地址） | | | | | | | |
| 1 | int | stno | 1140310188; | &stno | 0x0107F91C | 4 | ac c0 f7 43 |
| 2 | int | stnon | -1140310188 | &stnon | 0x0107F918 | 4 | 54 3f 08 bc |
| 3 | char \* | stnos | "1140310188" | stnos &stnos | 0x01215B30 0x0107F914 | 11 | 31 31 34 30 33 31 30 31 38 38 00 |
| 4 | char \* | stname | "学霸王" | stname &stname | 0x01215B44 0x0107F910 | 7 | d1 a7 b0 d4 cd f5 00 |
| 5 | bool | stsex | true | &stsex | 0x0107F90F | 1 | 01 （由于内存对齐，也占4个字节） |
| 6 | char \* | idno | "230106981221001" | idno &idno | 0x01215BE4 0x0107F908 | 16 | 32 33 30 31 30 36 39 38 31 32 32 31 30 30 31 00 |
| 7 | int | sfzhi | 230106981221001; | &sfzhi | 0x0107F904 | 4 | 89 42 e0 f4 |
| 8 | long | sfzhl | 230106981221001; | &sfzhl | 0x0107F900 | 4 | 89 42 e0 f4 |
| 9 | long long | sfzhll | 230106981221001; | &sfzhll | 0x0107F8F8 | 8 | 89 42 e0 f4 47 d1 00 00 |
| 10 | Float | sfzhf | 230106981221001; | &sfzhf | 0x0107F8F4 | 4 | f5 47 51 57 |
| 11 | Double | sfzhd | 230106981221001; | &sfzhd | 0x0107F8EC | 8 | 20 51 08 9c fe 28 ea 42 |

打印输出： 1140310188

-1140310188

1140310188

学霸王

1 ===女、False，值为0

230106981221001

-186629495

-186629495

230106981221001

230106983301120.000000

230106981221001.000000

解释：本题22分，每一变量的地址和内容得1分，打印结果各得1分。

注意：起始地址可以各不相同，但地址保持连续性，每个变量地址增加的数应如上。

Int与long的内容要一致，是负数，float与double不一致。

变量的地址与内容，需在内存查看窗口查看，如输入 &stno可在地址栏查看地址。

printf("%d\n", stno);

printf("%d\n", stnon);

printf("%s\n", stnos);

printf("%s\n", stname);

printf("%d\n", stsex);

printf("%s\n", idno);

printf("%d\n", sfzhi);

printf("%ld\n", sfzhl);

printf("%lld\n", sfzhll);

printf("%f\n", sfzhf);

printf("%lf\n", sfzhd);

3. 请填空 （b二进制 h十六进制 d十进制，缺省十进制） (8分)

1000=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_b 0.8125=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_b

10100101=\_\_\_\_\_\_\_\_\_\_\_\_\_\_d 0.1=\_\_\_\_\_\_\_\_\_\_b （24位有效数字）

C源程序中10或16进制无符号常数由\_\_\_\_\_\_\_\_\_\_软件转换成\_\_\_\_\_进制

答案：0000 0011 1110 1000 0.1101 0000

165 0.0001 1001 1001 1001 1001 1001

编译程序-编译器 二

4. 请写出下列有符号数的原码、反码、补码（8位2进制数） (20分)

0, 1, -1, 55h， 0aah， 128，-128，127，-127

C源程序中10或16进制整数由\_\_\_\_\_\_\_\_\_\_软件转换成\_\_\_\_\_进制\_\_\_\_\_\_编码

答案：0： 0000 0000B 0000 0000B 0000 0000B

1： 0000 0001B 0000 0001B 0000 0001B

-1： 1000 0001B 1111 1110B 1111 1111B

55H： 0101 0101B 0101 0101B 0101 0101B

0aah: 超出有符号数的范围，无法表示

128： 超出有符号数的范围，无法表示

-128： 无 无 1000 0000B

127： 0111 1111B 0111 1111B 0111 1111B

-127: 1111 1111B 1000 0000B 1000 0001B

编译程序-编译器 二 补码

解释：8位2进制有符号数的范围为-128~127。超出范围要说明，不要写出其原码、反码、补码。

5. 请查阅文献，了解浮点数的编码表示（ieee754编码标准）。 (10分)

并写出0.1的float、double的机器数表示。（可自己编程确认）

答案: 3D CC CC CD 3F B9 99 99 99 99 99 9A (也可表示成2进制)

内存中：cd cc cc 3d 9a 99 99 99 99 99 b9 3f

解释：每题5分

float： ieee754 符号位+阶码的移码(+7F)+尾数（规格化为1.XXXX…X后）

1位 8位 23位(去掉整数位的1后) 共32位

Double：ieee754 符号位+阶码的移码（+3FF）+尾数（规格化为1.XXXX…X后）

1位 11位 52位 共64位

6.请自己编程实现如下函数(不得用sprintf等实现) (20分)

int atoi(char \*s) s为十进制字符串如“123456” 返回串对应整数

int itoa(int i,char \*s) 返回字符串的长度 返回整数对应串

请把源程序粘贴到作业中提交。

答案：每个程序各10分。

用最后的main验证程序来验证，查看局部变量t和s。错一个扣1分。

int atoi(char \*s)

{

int i = 0;

int l = strlen(s);

int r = 0;

int sn= 1;

if (l < 1) return 0;

if (s[0] == '-')

{

i = 1;

sn = -1;

}

for (; i < l; i++)

if (s[i] >= '0' && s[i] <= '9')

r = r \* 10 + (s[i] - '0');

else

return 0;

return r\*sn;

}

int itoa(int i,char \*s)

{

//缺省认为s足够存放

// -2, 147, 483, 648 到 2, 147, 483, 647 共11位

long long li = i; 因为有-2, 147, 483, 648，取反时 int或long放不下

char ts[12];

int l=0,b,j,sn=1;

if (li < 0)

{

sn = -1;

li = -li;

}

if (li == 0)

{

s[0] = '0';

s[1] = 0;

return 1;

}

for (;;)

{

if (li == 0)

break;

b = li % 10;

li = li / 10;

ts[l] = b + '0'; l++;

}

if (sn == -1)

{

i = 1;

s[0] = '-';

}

else

i = 0;

for (j=0; j < l; i++,j++)

s[i] = ts[l - j - 1];

s[i] = 0;

return i; //字符个数

}

**//可以用一下主程序验证个函数是否执行正确！**

void main()

{

int t;

char buf[20];

t = atoi("abcd"); //应返回为0

t = atoi("0");

t = atoi("1");

t = atoi("-1");

t = atoi("2147483647");

t = atoi("-2147483648");

t = itoa(0,buf); //buf 应为“0”

t = itoa(1,buf); //buf 应为“1”

t = itoa(-1, buf); //buf 应为“-1”

t = itoa(2147483647, buf); //buf 应为“2147483647”

t = itoa(-2147483648, buf); //buf 应为“-2147483648”

}