

1. C 函数返回 struct 类型是如何实现的？

《解》

调用者在调用的时候,先把需要的参数反向入栈,最后再把变量的地址入栈. 在被调用的函数中,我们就完成了返回值的赋值工作,因为有了返回值赋值的变量的地址,所以依次将 struct 的各成员变量赋值给返回变量的对应内存即可.

2. C 函数是如何传入 struct 类型参数的？

请针对讲义“C 与汇编语言-4”中最后两张 slides 的内容, 回答上述问题。

具体要求：语言描述的同时请画出相应的程序栈的内容示意图。

《解》

2. struct 类型是如何实现的？

2014080013 计62 姜志祥

栈空间: subq \$32, %rsp
其次将当前栈指针地址放入 callee_saved 中: movq %rsp, %edi
堆栈如下

%rsp+24	/	/
%rsp+20	/	/
%rsp+16	n	/
%rsp+12	n	/
%rsp+8	2n	/
%rsp+4	n	/
%rsp	n	(Test struct)

Stack ~~main~~

如何传入 struct 类型参数的？

在 call input_struct 之前, 栈如下

%rsp+48	/	/
%rsp+44	/	/
%rsp+40	/	/
%rsp+36	i	/
%rsp+32	2i	/
%rsp+28	i	/
%rsp+24	i	← main_struct
%rsp+20	/	/
%rsp+16	i	/
%rsp+12	i	/
%rsp+8	2i	/
%rsp+4	i	/
%rsp	i	← copied main_struct

YANGJL

3、有如下 C 代码 (X86-64

copy_element:

```
pushq    %rbp
movq     %rsp, %rbp
movl     %edi, -4(%rbp)
movl     %esi, -8(%rbp)
movl     -8(%rbp), %eax
movslq   %eax, %rcx
movl     -4(%rbp), %eax
movslq   %eax, %rdx
movq     %rdx, %rax
addq     %rax, %rax
addq     %rdx, %rax
salq     $2, %rax
addq     %rdx, %rax
addq     %rcx, %rax
salq     $4, %rax
leaq     mat1(%rax), %rcx
```

Linux 系统) , 编译成对应的汇编代码, 请对照**两边**代码给出 M、N 的值。同时
请在汇编代码段中标出**哪些**语句的作用是**直接将源结构的字段复制**给目标结构
(每一个字段的**复制**指令都需分别标出) 。

```
typedef struct typeTAG
{
```

```

    double  attribute_3;
    char    attribute_2;
    int     attribute_1;
} TAG;

TAG mat1[M][N];
TAG mat2[N][M];
int copy_element(int i, int j)
{
    mat1[i][j] = mat2[j][i];
    return 0;
}

```

《解》

```

    movl    -8(%rbp), %eax  参数j复制

    movslq  %eax, %rcx

    movl    -4(%rbp), %eax  参数i复制

    movslq  %eax, %rdx

    movq    %rdx, %rax

```

第 6 行到 10 行是直接将源结构的字段复制给目标结构

—

4、 设计编写汇编程序， 要求如下：

(1) 程序命令行输入**两个**文件名， 都是文本文件， 名字分别为 input.txt 与 output_line_num.txt。 要求是统计第一个文件中的文字行数， 'LF'（即换行符， ASCII 码为 0x0A）为分割行的标记， 如果最后的文字直到文件末尾**没有** LF 的话， 也算做一行； 然后将**行数**以字符串形式**写入**第二个文件。

(2) 要求在程序**内**实现一个名为 getline 的**函数**， 带三个**参数**， 第一个为读取的正文文件的文件描述符， 第二个是存放一行文字（包含结尾的 0， 但不包括 LF）的**内存地址**， 第三个是该块内存的可用长度。 返回值是读入的文字的长度（不包括 0 与 LF）， 返回-1 表示调用失败， 如果已**达到**文件结尾， 可返回 0。

要求提供原码，运行截图，以及单独成文的设计说明文字。