第三章 程序的机器级表示

1. （控制）写出下面函数Func1汇编代码对应的C程序，其中参数1为x，参数2为y：

Func1:

cmpq %rsi, %rdi

jge .L2

leaq 3(%rsi), %rdi

jmp .L3

.L2:

leaq (%rdi,%rdi,4), %rsi

addq %rsi, %rsi

.L3:

leaq (%rdi,%rsi), %rax

ret

答：

int Func(int a, int b) { // %rdi = a, %rsi = b  
 if (b > a) {

a = 3 + b;  
 } else { // b <= a  
 b = 5\*a;  
 b += b;  
 }  
 return a + b;  
}

1. （多重数组+lea指令）对于数组int B[8][5],需要将B[i][j]保存到eax中，数组起始地址在rdi，i保存在rsi，j保存在rdx中，请完成以下代码中的空缺

leaq ( ,%rsi, ), %rax

leaq ( , , ), %rax

movl ( , , ), %eax

答：

B[i][j]等价于\*(B + i\*5 + j), 则：

leaq ( %rsi ,%rsi, 4 ), %rax # %rax = 5 \* %rsi  
leaq ( %rax, %rdx , 1 ), %rax # %rax = %rax + j  
movl ( %rdi , %rax , 4 ), %eax

1. （数组+函数+乘法的移位实现）已知int P[M][N]和int Q[N][M]，有以下函数：

int addfun( int i,int j){

return P[i][j]+Q[j][i];

}

对应有汇编代码如下，请问M和 N分别是多少？

addfun:

movl %edi, %edx

shl $2,%edx

addl %esi,%edx

movl %esi,%eax

shll $2,%eax

addl %eax,%edi

movl Q(,%edi,4),%eax

addl P(,%edx,4), %eax

ret

答：

分析如下：

addfun: # %edi = i, %esi = j  
movl %edi, %edx # %edx = i  
shl $2,%edx # edx = edx \* 4  
addl %esi,%edx # edx = edx + j = 4\*i+j  
movl %esi,%eax # eax = j  
shll $2,%eax # eax = eax \* 4  
addl %eax,%edi # edi = eax + i = 4\*j + i  
movl Q(,%edi,4),%eax   
addl P(,%edx,4), %eax  
ret

则M和N都为4.

1. （union+结构体）

union a1{

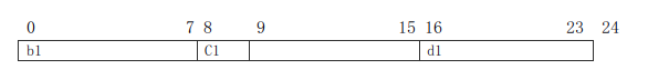
struct { int \* b1; char c1; long d1 } str1;

double data[3];

}

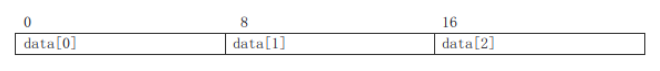
请问按照默认的对齐方式，上述a1占用多少字节空间？

a1.str1:



共24字节。

a1.data:



同样是共24字节。

即union中需要最大的空间为24字节，则a1占用24字节。

1. （结构体+函数+控制）已知node 结构体定义如下

struct node{

long a;

struct node \*next;

}

请对以下init函数进行逆向分析，写出其C代码

Init:

movl $12,$eax

jmp .TestExprStat

.Loop:

addq (%rdi),%rax

movq 8(%rdi),%rdi

.TestExprStat:

testq %rdi,%rdi

jne .Loop

ret

答：

long Init(struct node \*p) {  
 long res = 12;  
 while(p != NULL) {  
 res += p->a;  
 p = p->next;  
 }  
 return res;  
}

1. （结构体）已知结构体定义如下

struct{

char a;;

char \*b;;

short c;

int d;

}

请问在紧凑布局和对齐布局中a/b/c/d字段的偏移量各是多少？

* 紧凑布局中：

struct{  
 char a; // 0  
 char \*b; // 1   
 short c; // 9  
 int d; // 11  
};

* 对齐布局中：

struct{  
 char a; // 0  
 char \*b; // 8   
 short c; // 16  
 int d; // 20  
};

1. （堆栈破坏问题）函数echo定义如下：

void echo(){

char buf[8];

gets(buf);

puts(buf)；

}

返回地址（64位）

echo

的栈帧

%rsp

echo

的栈帧

%rsp

对应的汇编代码如下：

echo:

subq $24,$rsp

movq %fs:40,%rax

movq %rax,8(%rsp)

xorl %eax,%eax

movq %rsp,%rdi

call gets

movq %rsp,%rdi

call puts

movq 8(%rsp),%rax

xorq %fs:40,%rax

je .L9

call \_\_stack\_chk\_fail

addq $24,%rsp

ret

观察代码，判定该函数是否具有堆栈破坏的检测能力？如果%fs:38地址开始存放了0x00/01/02/03/04/05/06/07/08/09/0a/0b/0c/0d/0e/0f。请问刚进入echo函数时，echo栈帧中%rsp+8位置存放的8字节数值是？如果此时输入按键abcdefg并回车，程序将如何执行？如果此时输入按键123456789并回车，程序能否正常返回？如果不能将执行什么处理？

答：

* 该函数具有堆栈破话的检测能力。
* 06/07/08/09/0a/0b/0c/0d
  + 首先题目问的是刚进入echo函数时，echo栈帧中%rsp+8位置存放的8字节数值是多少，此时%rsp指向的是返回地址才对，而%rsp+8应该在调用echo()的函数的栈帧中，从题目给出的c代码来看，echo函数是没有参数的，所以这里只能猜测这个位置存放的是Canary值，即%fs:40存放的值。
  + 然后题目给出了%fs:38地址开始的的值，这里的%fs:38指向的地址应该为%fs + 38而%fs:40指向的地址为%fs + 40，考虑到大小端问题，这里8字节数值应该为0x06/07/08/09/0a/0b/0c/0d.
* 显示abcdefg
* 不能正常返回，因为占用了9个字节，调用了\_\_stack\_chk\_fail