姓名： 学号： 分数：

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，数组起始地址保存在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(,%rdi,4),%eax

addl P(,%rdx,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  
movl %esi,%eax # eax = j  
shll $2,%eax # eax = eax \* 4  
addl %eax,%edi # edi = eax + i  
movl Q(,%rdi,4),%eax   
addl P(,%rdx,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;

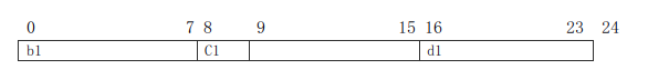
double data[3];

}

请问按照默认的对齐方式，上述a1.str1占用多少字节空间？ a1占用多少字节空间？

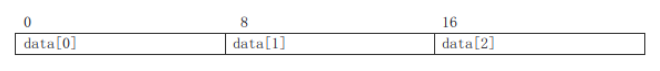
参考答案：

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？

答：

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)；

}

对应的汇编代码如下：

echo:

subq $24,%rsp

movq %fs:40,%rax

返回地址（64位）

echo

的栈帧

%rsp

echo

的栈帧

%rsp

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(不能确定是否正确)
  + 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(如果我没弄错的话),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

1. 对于以下代码：
2. int a=100;
3. short proc(int inputarg)
4. { int a;

a=inputarg;

return a+0;

1. }

请说明边第1行变量a和3行定义变量a的作用域。如果将第3行变量类型修改为static int，则第一次调用函数proc时传入参数inputarg的值为12，此时返回值为多少？再次调用时传入inputarg为5，则第二次调用的返回值是多少？

答：

a=inputarg; 修改为 a+=inputarg;

第一行变量a的作用域是全局，第三行变量a的作用域是函数proc内。

第一次：12

第二次：17

1. 对于以下myfile.c程序代码

int hello1;

char myfun(int a)

{ int b=10;

hello1=b+a;

hello1=hello1\*f1(a);

printf(“%d\n”,hello1);

return hello1;

}

请指出那些符号是链接器符号，哪些符号需要重定位？

参考答案：

链接器符号：hello1、myfun

重定位：hello1、f1、 printf

1. 请简单说明什么是连接器相关的强符号、弱符号？

答：

强符号是指已经初始化的全局变量和定义的函数名

弱符号是指未初始化的全局变量

1. 已知一个硬盘容量为1TB，由两个盘（4个盘面）构成，每个盘面由3200个磁道构成，每个磁道平均有128个扇区。由于磁记录密度的提升，新工艺维持相同的容量下，只需要一个盘（2个盘面），每个盘面的磁道数为3200，则新工艺要求每个磁道多少个扇区？

256

1. 请写出磁盘扇区访问时间的计算公式，并简单解释寻道时间、旋转时间、传送时间。

答：

访问时间=寻道时间+延迟时间+传输时间

寻道时间，也称寻找时间：磁头移动到指定磁道需要的时间

延迟时间：磁头定位到某一磁道的扇区所需要的时间

传输时间：从磁盘读出或者写入经历的时间

1. 如果cache块（行）的大小为64字节，cache总容量为1KB，请问cache的块数有多少？如果采用2路组相联则32位的内存地址，将划分成标记、组索引、块内偏移三个部分各需几个bit?

答：

内存地址32位，块内6 bits，cache有24= 16块，2路组相联，则cache有8组（3bits），每组内有2块。

23 3 6

|  |  |  |
| --- | --- | --- |
| 标记tag | 组索引 | 块内偏移 |

1. 请简单分析一下各项因素单独变化时对cache命中率的影响：cache容量大小变化、cache块大小、组的大小。
2. 什么是数据访问的时间局部性和空间局部性？

答：

时间局部性指的是：被引用过一次的存储器位置在未来会被多次引用（通常在循环中）

空间局部性: 如果一个存储器的位置被引用，那么将来他附近的位置也会被引用。