**3.32**

int fun(short c, char d, int \*p, int x)

**3.33**

**A.**

0x80003c

**B.**

0x800014

**C.**

0x800038, 0x800034

**D.**

|  |  |
| --- | --- |
| 0x80003c | 0x800060 |
| 0x800038 | 0x53 |
| 0x800034 | 0x46 |
| 0x800030 |  |
| 0x80002c |  |
| 0x800028 |  |
| 0x800024 |  |
| 0x800020 |  |
| 0x80001c | 0x800038 |
| 0x800018 | 0x800034 |
| 0x800014 | 0x300070 |

**E.**

0x800020 – 0x800033未使用

**3.34,**

**A.**

x

**B.**

int rfun (unsigned x)

{

if (x == 0)

return 0;

unsigned nx = x >> 1;

int rv = rfun(nx);

return rv + (x & 1);

}

**C.**

统计x的二进制表示中1的个数.

**3.36**

|  |  |  |  |
| --- | --- | --- | --- |
| 表达式 | 类型 | 值 | 汇编代码 |
| S + 1 | short\* | xs + 2 | leal 2(%edx), %eax |
| S[3] | short | M[xs + 6] | movw 6(%edx), %ax |
| &S[i] | short\* | xs + 2\*i | leal (%edx, %ecx, 2), %eax |
| S[4\*i + 1] | short | M[xs + 8\*i + 2] | movw 2(%edx, %ecx, 8), %ax |
| S + i - 5 | short\* | xs + 2\*i - 10 | leal -10(%edx, %ecx, 2), %eax |

**3.37**

5, 7.

**3.39**

**A.**

0 4 8 12

**B.**

16字节

**C.**

sp->s.y

&(sp->s.x)

sp

**3.40**

|  |  |  |
| --- | --- | --- |
| EXPR | TYPE | Code |
| up->t1.s | int | movl 4(%eax), %eax  movl %eax, (%edx) |
| up->t1.v | short | movw (%eax), %ax  movw %ax, (%edx) |
| &up->t1.d | short\* | leal 2(%eax), %eax  movl %eax, (%edx) |
| up->t2.a | int\* | movl %eax, (%edx) |
| up->t2.a[up->t1.s] | int | movl 4(%eax), %ecx  movl (%eax, %ecx, 4), %eax  movl %eax, (%edx) |
| \*up->t2.p | char | leal 8(%eax), %eax  movb (%eax), %al  movb %al, (%edx) |

**3.41**

**A.**

0 4 8 12; 16; 4.

**B.**

0 4 5 8; 12; 4.

**C.**

0 6; 10; 2.

**D.**

0 8; 20; 4.

**E.**

0 32; 36; 4.

**3.60**

**A.**

A(, i\*S\*T + j\*T + k, 4)

**B.**

11, 7, 9.

**3.62**

**A.**

19

**B.**

%edi, %ecx.

**C.**

int transpose(int A[M][M])

{

int i, j;

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

{

int \*a = &A[i][0];

int \*b = &A[0][i];

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

{

int t = \*a;

\*a = \*b;

\*b = t;

++a;

b += M;

}

}

}

**3.63**

3\*n, 2\*n – 1.

**3.64**

**A.**

8(%ebp)为&result;

12(%ebp)为s1.p;

16(%ebp)为s1.v.

**B.**

|  |
| --- |
| s2.sum |
| s2.prod |
| s1.v |
| s1.p |
| &s2 |

**C.**

结构体的每一个变量可以看做是单独的参数进行传入.

**D.**

将返回变量的地址作为第一个参数传入函数, 而不是在函数中为该变量分配栈空间.

因为%eax存不下一个结构体, 所以%eax里存放返回变量的指针.

**3.66**

**A.**

7

**B.**

typedef struct

{

int idx;

int x[6];

};

**3.67**

**A.**

0 4 0 4

**B.**

8

**C.**

void proc(union ele \*up){

up->e2.next->e1.x = \*(up->e2.next->e1.p) - up->e2.y;

}