第二章

【整数的表示】

1. 在x86-64机器上，定义unsigned int A = 0x123456。请画出A在内存中的存储方式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ... | 低地址 | A | | | | 高地址 | ... |
| ... | | 0x56 | 0x34 | 0x12 | 0x00 | ... | |

定义unsigned short B[2] = {0x1234, 0x5678}。请画出B在内存中的存储方式：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ... | 低地址 | B | | | | 高地址 | ... |
| ... | | 0x34 | 0x12 | 0x78 | 0x56 | ... | |

2. 在x86-64机器上，有下列C代码

|  |
| --- |
| int main() {  unsigned int A = 0x11112222;  unsigned int B = 0x33336666;  void \*x = (void \*)&A;  void \*y = 2 + (void \*)&B;  unsigned short P = \*(unsigned short \*)x;  unsigned short Q = \*(unsigned short \*)y;  printf("0x%04x", P + Q);  return 0;  } |

运行该代码，结果为：0x 。【答案：0x5555】

3. 在x86-64机器上，有下列C代码

|  |
| --- |
| int main() {  char A[12] = "11224455";  char B[12] = "11445577";  void \*x = (void \*)&A;  void \*y = 2 + (void \*)&B;  unsigned short P = \*(unsigned short \*)x;  unsigned short Q = \*(unsigned short \*)y;  printf("0x%04x", Q - P);  return 0;  } |

运行该代码，结果为：0x 。【答案：0x0303】

【整数的运算】

4. 在x86-64机器上，有如下的定义：

|  |
| --- |
| int x = \_\_\_\_\_\_\_\_;  int y = \_\_\_\_\_\_\_\_;  unsigned int ux = x;  unsigned int uy = y; |

判断下列表达式是否等价：

提示：减法的运算优先级比按位异或高。布尔运算的结果都是有符号数。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 表达式A | 表达式B | 等价吗？ |
| (1) | x > y | ux > uy | Y N |
| (2) | (x > 0) || (x < ux) | 1 | Y N |
| (3) | x ^ y ^ x ^ y ^ x | x | Y N |
| (4) | ((x >> 1) << 1) <= x | 1 | Y N |
| (5) | ((x / 2) \* 2) <= x | 1 | Y N |
| (6) | x ^ y ^ (~x) - y | y ^ x ^ (~y) - x | Y N |
| (7) | (x == 1) && (ux – 2 < 2) | (x==1) && ((!!ux)–2<2) | Y N |

【答】(1)取x=1, y=-1即不正确；(2)取x=-1即不正确；(3)正确，利用交换律、结合律，以及x ^ x == 0；(4)正确，即使是对负数；(5)不正确，负奇数该运算向0舍入；(6)正确，(~x) – y 也就是 (~x) + (~y) + 1，注意运算优先级；(7)不正确，!!ux是有符号数。

5. 下列代码的目的是将字符串A的内容复制到字符串B，覆盖B原有的内容，并输出”Hello World”；但实际运行输出是”Buggy Codes”。尝试找到代码中的错误。

|  |
| --- |
| int main() {  char A[12] = "Hello World";  char B[12] = "Buggy Codes";  int pos;  for (pos = 0; pos - sizeof(B) < 0; pos++)  B[pos] = A[pos];  printf("%s\n", B);  } |

【答：sizeof的结果是无符号数，因此pos – sizeof(B) < 0恒假，于是不会进行任何复制。】