Милосердова Лидия

Лабораторная работа 5 (0101 = 5)

Использование ассемблерных вставок в программах на C++. Передача параметров во вставку. Перезаписываемые элементы

**Задание 1**

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout << "Ассемблер: " << endl;**

**unsigned int x =10, y = 3, z, w;**

**asm (**

**"movl %[X], %%eax\n\t"**

**"movl %[Y], %%ebx\n\t"**

**"subl $23, %%ebx\n\t"**

**"divl %%ebx\n\t"**

**:"=a"(z), "=d"(w)**

**:[X]"m"(x), [Y]"m"(y)**

**:"cc"**

**);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**cout << "C++: " << endl;**

**z = x/(y-23);**

**w = x%(y-23);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**return 0;**

**}**

**Ассемблер:**

**z = 0**

**w = 10**

**C++:**

**z = 0**

**w = 10**

**mov eax,DWORD PTR [rbp-0xc]**

**mov ebx,DWORD PTR [rbp-0x10]**

**sub ebx,0x17**

**div ebx**

**mov DWORD PTR [rbp-0x4],eax**

**mov DWORD PTR [rbp-0x8],edx**

**c++  
mov eax,DWORD PTR [rbp-0x10]**

**lea ecx,[rax-0x17]**

**mov eax,DWORD PTR [rbp-0x4]**

**mov edx,0x0**

**div ecx**

**mov DWORD PTR [rbp-0x8],eax**

**mov eax,DWORD PTR [rbp-0x10]**

**lea ecx,[rax-0x17]**

**mov eax,DWORD PTR [rbp-0x4]**

**mov edx,0x0**

**div ecx**

**mov DWORD PTR [rbp-0xc],edx**

**Задание 2**

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout << "Ассемблер: " << endl;**

**unsigned int x =10, y = 3, z, w;**

**asm (**

**"subl $23, %[Y]\n\t"**

**"divl %[Y]\n\t"**

**"movl %%edx, %%ebx\n\t"**

**:"=a"(z), "=b"(w)**

**:[X]"a"(x), [Y]"r"(y)**

**:"cc","edx"**

**);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**cout << "C++: " << endl;**

**z = x/(y-23);**

**w = x%(y-23);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**return 0;**

**}**

**Ассемблер:**

**z = 0**

**w = 10**

**C++:**

**z = 0**

**w = 10**

**mov eax,DWORD PTR [rbp-0x14]**

**mov ecx,DWORD PTR [rbp-0x18]**

**sub ecx,0x17**

**div ecx**

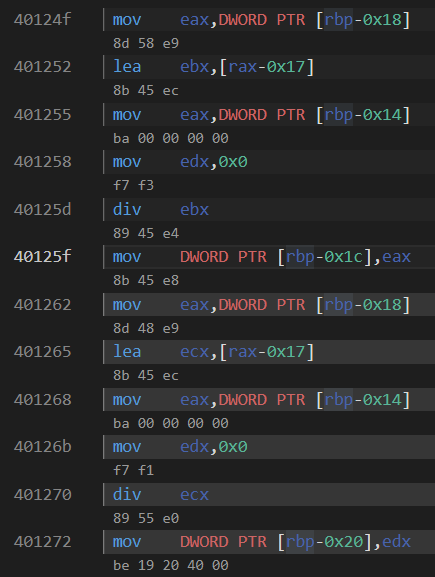
**mov ebx,edx**

**mov ecx,ebx**

**mov DWORD PTR [rbp-0x1c],eax**

**mov DWORD PTR [rbp-0x20],ecx**

**C++**

****

**3**

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout << "Ассемблер: " << endl;**

**unsigned int x =10, y = 3, z, w;**

**asm (**

**"movl (%[pX]), %%eax\n\t"**

**"movl (%[pY]), %%ebx\n\t"**

**"subl $23, %%ebx\n\t"**

**"xorl %%edx, %%edx\n\t"**

**"divl %%ebx\t"**

**:"=a"(z), "=d"(w)**

**:[pX]"r"(&x), [pY]"r"(&y)**

**:"cc"**

**);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**cout << "C++: " << endl;**

**z = x/(y-23);**

**w = x%(y-23);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**return 0;**

**}**

**Ассемблер:**

**z = 0**

**w = 10**

**C++:**

**z = 0**

**w = 10**

**4**

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout << "Ассемблер: " << endl;**

**unsigned short x =10, y = 3, z, w;**

**asm (**

**"movw %[X], %%ax\n\t"**

**"movw %[Y], %%bx\n\t"**

**"subw $23, %%bx\n\t"**

**"divw %%bx\n\t"**

**:"=a"(z), "=d"(w)**

**:[X]"m"(x), [Y]"m"(y)**

**:"cc"**

**);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**cout << "C++: " << endl;**

**z = x/(y-23);**

**w = x%(y-23);**

**cout << "z = " << z << endl;**

**cout << "w = " << w << endl << endl;**

**return 0;**

**}**

**Ассемблер:**

**z = 0**

**w = 10**

**C++:**

**z = 0**

**w = 10**

**5**

**#include <cstdio>**

**#include <iostream>**

**#include <random>**

**using namespace std;**

**const int N = 10;**

**int main() {**

**srand(time(NULL));**

**int M[N];**

**size\_t i = rand()%N;**

**for (int j = 0; j < N; j++)**

**M[j] = 0x44332211;**

**cout << "До: ";**

**for (int j = 0; j < N; j++)**

**cout << hex << showbase << M[j] << " ";**

**cout << endl;**

**asm(**

**"movl $65535, (%[M], %[I], 4)\n"**

**: [I] "+r"(i)**

**: [M] "d"(M)**

**: "memory"**

**);**

**cout << "После: ";**

**for (int j = 0; j < N; j++)**

**cout << hex << showbase << M[j] << " ";**

**cout << endl;**

**}**

**До: 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211**

**После: 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0xffff 0x44332211 0x44332211**

**6**

**#include <cstdio>**

**#include <iostream>**

**#include <random>**

**using namespace std;**

**const int N = 10;**

**int main() {**

**srand(time(NULL));**

**int M[N];**

**size\_t i = rand()%N;**

**size\_t j = rand()%N;**

**while (j == i)**

**j = rand()%N;**

**for (int j = 0; j < N; j++)**

**M[j] = 0x44332211;**

**cout << "До: ";**

**for (int k = 0; k < N; k++)**

**cout << hex << showbase << M[k] << " ";**

**cout << endl;**

**asm(**

**"movl $65535, (%[M], %[I], 0x4)\n"**

**"movb $0xFF, 0x-1(%[M], %[J], 0x4)\n"**

**: [I] "+r"(i), [J] "+r"(j)**

**: [M] "d"(M)**

**: "memory"**

**);**

**cout << "После: ";**

**for (int k = 0; k < N; k++)**

**cout << hex << showbase << M[k] << " ";**

**cout << endl;**

**}**

**До: 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211 0x44332211**

**После: 0x44332211 0x44332211 0x44332211 0xff332211 0x44332211 0x44332211 0x44332211 0x44332211 0xffff 0x44332211**