

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | Системы автоматизированного проектирования | | |
| Форма обучения | |  | очная | | |
|  | |  |  | | |
| Факультет | |  | Информационных технологий и управления | | |
| Кафедра | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Вычислительные системы, сети и телекоммуникации | | |
| Курс | II | | | Группа | 423 |

Отчёт по контрольной работе №4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнители: |  |  |  |  |
| обучающиеся группы 423 |  |  |  | Ефремов Иван Андреевич  Агишев Даниил Рустамович  Сафаров Рустам Фахриддинович |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Макарук Роман Валерьевич |
|  |  | (дата, подпись) |  | Федин Алексей Константинович |
|  |  |  |  |  |

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc158160395)

[2 Исходные данные 3](#_Toc158160396)

[3 Текст учебной программы 3](#_Toc158160397)

[4 Выводы по заданию 6](#_Toc158160398)

## 1 Постановка задачи

Восстановить исходную программу на языке C++ по части

дизассемблированного кода

## 2 Исходные данные

Учебная программа – подсчет суммы двух целочисленных значений:

push ebp

mov ebp,esp

mov dword ptr [ebp-8],0

mov eax,dword ptr [ebp+8]

sub eax,20h

imul eax,eax,5

mov ecx,9

idiv eax,ecx

mov dword ptr [ebp-8],eax

mov eax,dword ptr [ebp-8]

mov esp,ebp

pop ebp

ret

push ebp

mov ebp,esp

mov eax,dword ptr [ebp+8]

add eax,dword ptr [ebp+0Ch]

mov dword ptr [ebp+8],eax

mov eax,dword ptr [ebp+8]

mov esp,ebp

pop ebp

ret

push ebp

mov ebp,esp

push 64h

call 01A10E6h

add esp,4

mov dword ptr [a],eax

push 10h

push 8

call 01A11D1h

add esp,8

mov dword ptr [b],eax

mov esp,ebp

pop ebp

ret

## 3 Текст учебной программы

Исходный код программного решения:

#include <iostream>

using namespace std;

#include <iostream>

using namespace std;

int\* a;

int\* b;

int FirstFunction(int arg) {

int res = 0;

int eax = arg;

eax -= 32;

eax \*= 5;

int ecx = 9;

eax /= ecx;

res = eax;

return res;

}

int SecondFunction(int argFirst, int argSecond){

int eax = argFirst;

eax += argSecond;

argFirst = eax;

eax = argFirst;

return argFirst;

}

void ThirdFunction() {

int argFirst = 100;

\*a = FirstFunction(argFirst);

argFirst = 16;

int argSecond = 8;

\*b = SecondFunction(argFirst, argSecond);

return;

}

int main() {

a = new int();

b = new int();

ThirdFunction();

cout << "Register [a] has value of: " << \*a;

cout << endl;

cout << "Register [b] has value of: " << \*b;

delete a;

delete b;

return 0;

}

Далее представлены фрагменты ассемблер кода для каждой арифметической операции.

Таблица 1 – Ассемблерный код

|  |  |
| --- | --- |
| **Команда ассемблера** | **Комментарий** |
| push ebp | Сохранение содержимого ebp в стеке |
| Mov ebp, esp | Создание «рамки» стека для текущей функции |
| mov dword ptr [ebp-8],0 | Создание локальной переменной |
| mov eax,dword ptr [ebp-8] | Перемещение аргумента функции в eax |
| sub eax,20h | Вычитание 20h из eax |
| imul eax,eax,5 | Умножение eax на 5 |
| mov ecx,9 | Запись 9 в регистр ecx |
| idiv eax,ecx | Деление eax на ecx |
| mov dword ptr [ebp-8],eax | Присваивание локальной переменной значения из аккумулятора |
| mov eax,dword ptr [ebp-8] | Получение значения для возврата |
| mov esp,ebp | Устанавливаем размер стека на исходное значение |
| pop ebp | Возвращение указателя размера стека на исходное значение |
| ret | Возврат |
| push ebp | Сохранение содержимого ebp в стеке |
| mov ebp,esp | Создание «рамки» стека для текущей функции |
| mov eax,dword ptr [ebp+8] | Перемещение первого аргумента в аккумулятор |
| add eax,dword ptr [ebp+0Ch] | Добавление второго аргумента к аккумулятору |
| mov dword ptr [ebp+8],eax | Устанавливаем равенство |
| mov eax,dword ptr [ebp+8] | Устанавливаем равенство |
| mov esp,ebp | Устанавливаем размер стека на исходное значение |
| pop ebp | Возвращение указателя размера стека на исходное значение |
| ret | Возврат |
| push ebp | Сохранение содержимого ebp в стеке |
| mov ebp,esp | Создание «рамки» стека для текущей функции |
| push 64h | Запись значения 64h на вершину стека |
| call 01A10E6h | Вызов первой функции с аргументом на вершине стека |
| add esp,4 | Удаление первого аргумента |
| mov dword ptr [a],eax | Запись результата вызова по адресу a |
| push 10h | Запись значения 10h на вершину стека |
| push 8 | Запись значения 8 на вершину стека |
| call 01A11D1h | Вызов второй функции с аргументами из вершины стека |

Таблица 1 – Продолжение

|  |  |
| --- | --- |
| add esp,8 | Удаление аргументов |
| mov dword ptr [b],eax | Устанавливаем равенство |
| mov esp,ebp | Устанавливаем размер стека на исходное значение |
| pop ebp | Возвращение указателя размера стека на исходное значение |
| ret | Возврат |

## 4 Выводы по заданию

В ходе выполнения задания были изучены методы и получены навыки восстановления исходного кода программы по части дизассемблированного кода.