УO «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа

**«Использование языка Ассемблер»**

Выполнил студент

3 курса, 4 группы

Чаевский А. В.

Проверила

Герман Ю. О.

Минск 2019

# **1 Постановка задачи**

Цель: Изучить основы языка Ассемблера как псевдомашинного языка программирования**.**

**2 Краткие сведения**

Рассмотрим следующую программу на языке с++, в которую добавлен код на ассемблере:

#include "stdafx.h"

#include <iostream>

void sum(int ,int );

int main()

{

int a=12;

int b=34;

\_asm

{

push a

push b

call sum

pop b

pop a

}

getchar();

return 0;

}

void sum(int a,int b)

{ printf("%d\n",a+b); }

Код на Ассемблере вставляется в фрагменте \_asm{…} В нашем примере этот код работает со стеком: заносит в стек операнды a , b и вызывает функцию sum. Тот же пример можно было бы реализовать следующим образом:

#include "stdafx.h"

#include <iostream>

void sum(int );

int main()

{

short a=12;

short b=5;

\_asm

{

mov ax,a

mov bx,b

add ax,bx

mov a,ax

}

sum(a);

getchar();

return 0; }

void sum(int a)

{ printf("%d\n",a); }

Второй пример работает без стека. Замечания по первой программе:

push - заносит операнд в стек; pop – выгружает из стека. Call – вызывает программу (процедуру). При этом данные передаются через стек. Команда mov пересылает правый операнд на место левого. Теперь познакомимся с командами сравнения. Следующий код находит наименьшее число из двух коротких целых чисел:

#include "stdafx.h"

#include <iostream>

int main()

{ short a=12;

short b=5;

\_asm

{ mov ax,a

mov bx,b

cmp ax,bx

jge bbb

push a

jmp eee

bbb: push b

eee: pop a}

printf("%d\n", a);

getchar();

return 0; }

Команда cmp сравнивает два операнда. Команда jge выполняет переход на метку, если первый операнд больше или равен второму. Команда jmp выполняет безусловный переход на метку.

Наконец, рассмотрим двоичные операции.

#include "stdafx.h"

#include <iostream>

void sum(int );

int main()

{

short a=2;

\_asm

{

mov ax,a

OR ax,00000001B

mov a,ax

}

printf("%d\n", a);

return 0; }

Команда OR находит логическое ИЛИ. Команда AND – логическое И. С помощью этих команд можно уже проверить, стоит ли, скажем, в третьем разряде 1. Оставляем этот вопрос на самостоятельное задание.

Интерес представляет использование арифметического сопроцессора. Если таковой физически отсутствует, то используется его программный эмулятор. Сопроцессор позволяет выполнять различные арифметические и алгебраические команды на регистрах разрядности 64. Все регистры организованы в стеке. Обозначается ST(0) – вершина стека (начальный регистр), ST(1) – второй регистр стека и т.д. Поясним команды, используя следующий пример.

#include “stdafx.h”

#include <stdio.h>

int main()

{

float aa=10.0;

\_\_asm{

FINIT

FLD aa

FLD aa

FMUL

FLDPI

FMUL

FST aa

FWAIT

}

printf (“%f”,aa);

getchar();

return 0;

}

Команда **инициализации** сопроцессора - FINIT .

**Загрузка в ST(0)** операнда с именем aa - FLD aa .

**Умножение** ST(0)xST(1) с занесением результата в ST(0) - FMUL

Загрузка в ST(0) числа π - FLDPI

**Запись результата в** ячейку (переменную) памяти aa - FST aa.

**Ожидание** завершения работы сопроцессора – FWAIT.

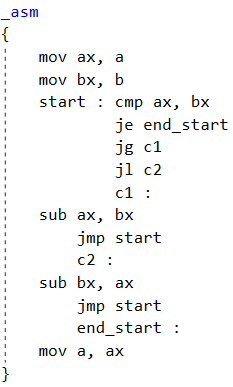
Остальные команды, которые могут пригодиться в выполнении лабораторной работы взять из конспекта лекций.



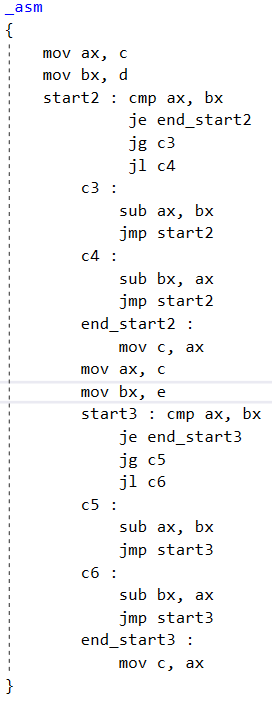
# **3 Задание**

1. Реализовать код на ассемблере для нахождения наибольшего общего делителя двух целых коротких положительных чисел. Алгоритм Евклида состоит в следующем.

1. Ввести А и В.
2. Если А=В, то конец. Вывести А.
3. Если А> В , то принять А=А-В. Переход на шаг 1.
4. Если В> А , то принять В=В-А. Переход на шаг 1



2. Реализовать код на языке Ассемблера для отыскания наибольшего общего делителя трех коротких положительных чисел.



3. Проверить, стоит ли в третьем разряде справа двоичного числа 1 или 0.

Копирование строк:

#include "stdafx.h"

#include <iostream>

void sum(int );

const char \*a = "Hello world!";

char\* b;

void my\_strcpy();

int main(void)

{

my\_strcpy ();

printf("Original string: '%s'\n", a);

printf("Copied string: '%s'\n", b);

getchar();

return 0;

}

void my\_strcpy()

{

\_\_asm

{

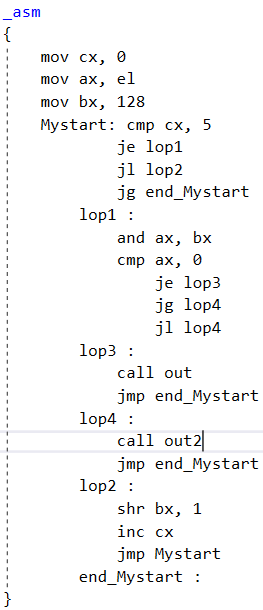
mov eax,a

mov b,eax

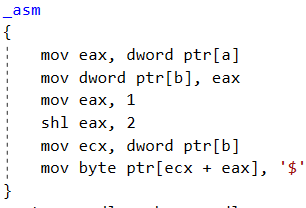
}

return;

}



1. Скопировать на базе приведенного примера одну строку в другую с заменой последней буквы первой строки на символ $.



**5**. Посчитать на сопроцессоре формулу:

Если a>b, то вернуть a2 иначе

вернуть b2.

