Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Реферат**:

«Элементы программирования на языке Ассемблер»

Выполнил:

студент 3 курса 4 группы

специальности ПОИТ

Карленок Ю.А.

Минск 2020

Ассемблер работает с ячейками памяти и регистрами. Регистрами в языке ассемблер называют ячейки памяти, расположенные непосредственно на кристалле с АЛУ. Особенностью этого типа памяти является скорость обращения к ней, которая значительно быстрее оперативной памяти ЭВМ.

Существуют следующие виды регистров:

* Регистры общего назначения (РОН).
* Флаги.
* Указатель команд.
* Регистры сегментов.

Регистры общего назначения размерностью в 32 бита: EAX, ECX, EDX, EBX.

Регистры общего назначения размерностью в 16 бит: AX, BX, CX, DX.

Регистры общего назначения размерностью в 8 бит: AH и AL, BH и BL и т. д.

Регистры, имеющие префикс **E**, являются расширенными.

Accumulator register (**AX**) – для арифметических операций.

Counter register (**CX**) – для сдвигов и циклов.

Data register (**DX**) – для арифметических операций и операций ввода/вывода. Base register (**BX**) – для указателя на данные.

Stack Pointer register (**SP**) – для указателя вершины стека.

Stack Base Pointer register (**BP**) – для индикатора основания стека.

Source Index register (**SI**) – для указателя отправителя (источника).

Destination Index register (**DI**) – для получателя.

Работать на языке C++ с языком ассемблера можно двумя способами: подключить файл инструкций с расширением .asm, либо же напрямую в C++ коде с помощью блока \_asm{ }.

Основные команды ассеблера:

* Pop – очистка стека.
* Push – поместить в стек.
* Mov – передача данных.
* Call – вызов функций.
* Sub – вычитание.
* Add – сложение.
* Xor – логическое и/или.
* Not – инверсия битов.

Познакомимся теперь с работой математического сопроцессора. Он позволяет обрабатывать вещественные числа и вычислять тригонометрические, гиперболические и логарифмические функции.

Команда **инициализации** сопроцессора - FINIT.

**Загрузка** в ST(0) операнда с именем [name] - FLD [name].

**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 [name] - FST [name].

**Ожидание** завершения работы сопроцессора – FWAIT.