**Лабораторная работа № 1**

**Тема:** Команды безусловного и условного переходов в языке Ассемблер. Реализация конструкции if-else на языке ассемблера.

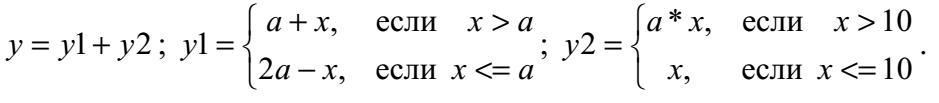
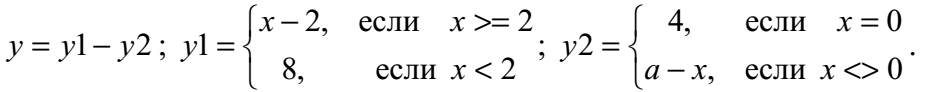
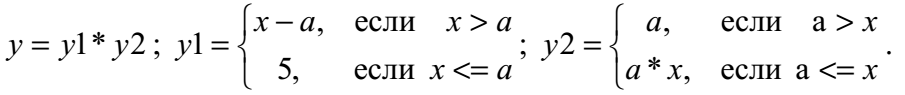
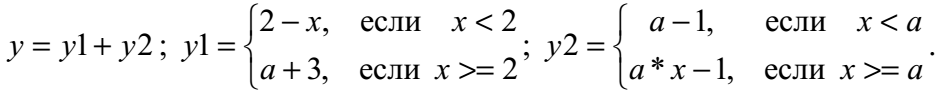
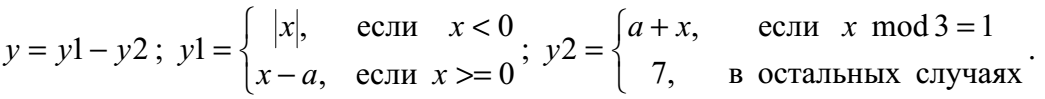
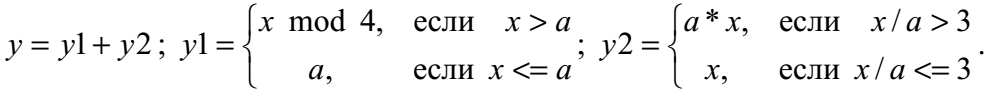
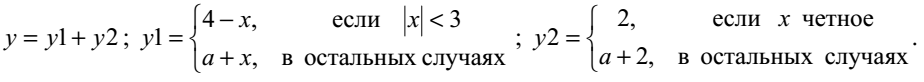
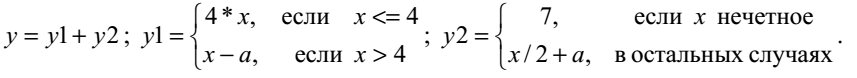
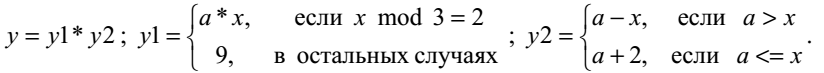
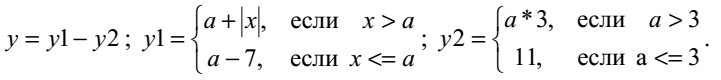
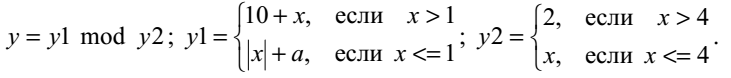
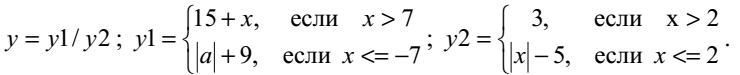
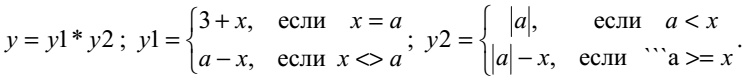
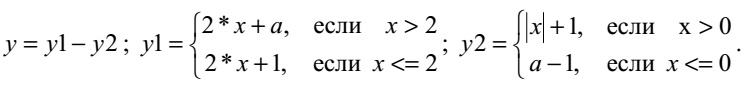
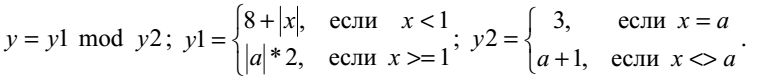
**Цель работы:** Приобретение навыков использования команд переходов при программировании на языке ассемблера. Реализация конструкции if-else на языке ассемблера.

**Оборудование:** персональный компьютер.

**Программное обеспечение:** Windows 7, Ms Visual Studio

**Задания к лабораторной работе:**

В соответствии с вариантом напишите программу на языке ассемблера (с помощью ассемблерных вставок). Сравните полученный результат.

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. 
15. 

**Краткие теоретические сведения:**

**1. Основные понятия и определения.** В языках программирования высокого уровня (таких как С++) конструкция выбора известна как оператор IF-THEN. Эта конструкция позволяет выбрать следующее действие из нескольких возможных вариантов в зависимости от выполнения определенного условия. В языке ассемблера механизм выбора реализован посредством команд сравнения, условного и безусловного переходов.

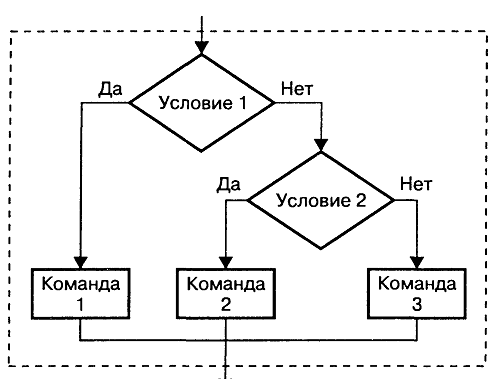


Рис. 1 Выбор следующей команды.

**2. Команда СМР.** Команда СМР используется для сравнения двух операндов. Операндами могут быть как регистры, так и адреса памяти, размер операнда — 8, 16 6или 32 бита.

СМР *Операнд1, Операнд2*

Команда СМР — это сокращение от «compare», «сравнить». Она работает подобно SUB: *Операнд2* вычитается из *Операнд1*. Результат нигде не сохраняется, команда просто изменяет регистр признаков. Команда СМР может использоваться как для сравнения целых беззнаковых чисел, так и для сравнения чисел со знаком.

Примеры:

cmр ах, 4 ;сравниваем АХ со значением 4

сmр dl, ah ;сравниваем DL с АН

сmр diameterl, ах ;сравниваем переменную "diameterl" с АХ

cmp еах, есх ;сравниваем регистры ЕАХ и ЕСХ

**3. Команда безусловного перехода — JMP.** Самый простой способ изменить последовательность выполнения команд заключается в использовании команды jmp — так называемой команды безусловного перехода. Она перезаписывает указатель команд (регистр IP или CS), что заставляет процессор «переключиться» на выполнение команды по указанному адресу. Формат команды таков:

JMP [тип\_перехода] операнд

Команда JMP — аналог конструкции GOTO, которая используется высокоуровневых языках программирования. Название команды объясняет ее действие, а именно «jump», «переход». Команде нужно передать один обязательный операнд — адрес в памяти, с которого процессор должен продолжить выполнение программы. Операнд может быть указан явно (непосредственное значение адреса) или быть регистром общего назначения, в который загружен требуемый адрес.

Пример:

mov ах, 4 ;АХ = 4

new\_step: ;метка new\_loop

mov bx, ах ;копируем АХ в ВХ

Чтобы перейти к метке new\_step из другого места программы, используйте команду:

Jmp new\_step ;переходим к new\_step

**4. Условные переходы — Jxx.** Другой способ изменения последовательности выполнения команд заключается в использовании команды условного перехода. В языке ассемблера имеется множество команд условного перехода. Имена этих команд различаются в зависимости от условия перехода. Условие состоит из значений одного или нескольких флагов в регистре признаков. Работают эти команды одинаково: если условие истинно, выполняется переход на указанную метку, если нет, то процессор продолжит выполнять программу со следующей команды.

Общий формат команд условного перехода следующий:

Jxx *метка\_назначения*

Пример:

Jz is\_true ;переходит к is\_true, если флаг ZF = 1

Jc is\_true ;переходит к is\_true, если флаг CF = 1

Js is\_true ;переходит к is\_true, если флаг SF = 1

Jo is\_true ;переходит к is\_true, если флаг переполнения 0F=1

Перечень инструкций условных переходов приводится в таблице 1.

Таблица 1 - Инструкции условных переходов

|  |  |  |
| --- | --- | --- |
| **Название** | **Значение** | **Проверяемые**  **флаги** |
| JB/JNAE | Перейти, если меньше/ перейти, если не больше или равно | CF = 1 |
| JAE/JNB | Перейти, если больше или равно / перейти, если не меньше | CF = 0 |
| JBE/JNA | Перейти, если меньше или равно / перейти, если не больше | CF = 1 или ZF = 1 |
| JA/JNBE | Перейти, если больше/ перейти, если не меньше или равно | CF = 0 и ZF = 0 |
| JE/JZ | Перейти, если равно | ZF = 1 |
| JNE/JNZ | Перейти, если не равно | ZF = 0 |
| JL/JNGE | Перейти, если меньше чем/ перейти, если не больше чем или равно | SF = OF |
| JGE/JNL | Перейти, если больше чем или равно /перейти, если не меньше чем | SF = OF |
| JLE/JNLE | Перейти, если меньше чем или равно/ перейти, если не больше, чем | ZF = 1 или SF = OF |
| JG/JNLE | Перейти, если больше чем/ перейти, если не меньше чем или равно | ZF = 0 или SF = OF |
| JP/JPE | Перейти по четности | PF = 1 |
| JNP/JPO | Перейти по нечетности | PF = 0 |
| JS | Перейти по знаку | SF = 1 |
| JNS | Перейти, если знак не установлен | SF = 0 |
| JC | Перейти при наличии переноса | CF = 1 |
| JNC | Перейти при отсутствии переноса | CF = 0 |
| JO | Перейти по переполнению | OF = 1 |
| JNO | Перейти при отсутствии переполнения | OF = 0 |

**CF** - флаг переноса, **SF** - флаг знака, **OF** - флаг переполнения, **ZF** - флаг нуля, **PF** - флаг четности.

**5. Пример реализации конструкции if-then-else на языке ассемблера.** Для примера преобразуем код С++:

if( a == b )

c = d;

else

b = b + 1;

в код на языке ассемблера:

mov eax, a

cmp eax, b

jne ElsePart

mov c, d

jmp EndOfIf

ElsePart:

inc b;

EndOfIf:

**Порядок проведения лабораторной работы:**

1. Изучить краткие теоретические сведения.
2. Выполнить задания согласно варианту.
3. Оформить отчет по лабораторной работе.

**Оформление отчета по лабораторной работе:**

Отчет оформляется согласно методическим указаниям по оформлению отчетов.

**Вопросы для зачета по работе:**

1. Перечислите флаги, используемые при выполнении условных переходов.
2. Охарактеризуйте группу команд сравнения операндов.
3. Охарактеризуйте группу команд условных переходов, ориентированных на сравнение беззнаковых чисел.
4. Охарактеризуйте группу команд условных переходов, ориентированных на сравнение знаковых чисел.

**Рекомендуемая литература:**

1. Зубков С.В. Assembler для DOS, Windows и UNIX. 3-е изд., стер. – М.: ДМК Пресс ; СПб.: Питер, 2004. – 608 с. : ил. – (Серия «Для программистов»).
2. Крупник А.Б. Изучаем Ассемблер – СПб.: Питер, 2005. – 249 с.: ил.
3. Жуков А.В., Авдюхин А.А. Ассемблер. – СПб.: БХВ-Петербург, 2002. – 448 с.: ил.