

Институт (факультет) _____ Институт информационных технологий
Кафедра _____ Математического и Программного Обеспечения ЭВМ

по дисциплине	Программирование на ассемблере
на тему	Программирование на языке низкого уровня

Оценка _____, _____
количество баллов

Подпись преподавателя _____

Череповец, 2024 год

Аннотация

Курсовую работу по дисциплине «Программирование на ассемблере» на тему «Программирование на языке низкого уровня» выполнил студент группы 1ПИБ-02-2оп-22 ИИТ Овчинников М.В.

Тема: Программирование на языке низкого уровня.

Курсовая работа была создана на основе Технического задания, описанного в Приложении 1.

Курсовая работа посвящена программированию на языке низкого уровня Assembler. Продуктом курсовой работы будет являться программа, подсчитывающая количество структур, подходящих по условию.

Курсовая включает в себя разделы: аннотация; оглавление; введение; основная часть; заключение; список литературы. Также вместе с курсовой работой идут приложения: техническое задание; руководство пользователя; текст программы.

Оглавление

Введение	4
1. Изучение и описание предметной области	5
2. Постановка задачи	7
2.1 Исходные данные	7
2.2 Набор выполняемых функций	7
3. Выбор структур данных для решения поставленной задачи	7
4. Логическое проектирование	8
5. Физическое проектирование	10
6. Проектирование интерфейса.....	10
7. Кодирование	11
8. Тестирование	12
Заключение	15
Список литературы	16
Техническое задание. Приложение 1	17
Руководство пользователя. Приложение 2	24
Программный код. Приложение 3	29

Введение

Данная курсовая работа фокусируется на создании программы для подсчета количества структур, подходящих под условие, что является актуальным в наше время.

Основной целью работы является определение количества книг с более чем 200 страницами и передача этой информации в регистр DI. Для достижения этой цели необходимо создать эффективный алгоритм и его программную реализацию.

В рамках данной работы будет проведен анализ языка программирования низкого уровня и его инструментов, применяемых для обработки структур и массивов. Особое внимание будет уделено алгоритмам подсчета количества структур, удовлетворяющих условиям.

1. Изучение и описание предметной области

Предметной областью является набор команд для ассемблера x86, а именно команды для работы с массивами и структурами.

В этом языке программирования используются различные директивы для определения данных: DB (байт – 8 бит), DW (слово – 16 бит), DD (двойное слово – 32 бита), DP (слово в 6 байт – 48 бит), DQ (четверное слово – 64 бита) и DT (слово в 10 байт – 80 бит) [1].

Для перемещения данных из одной области в другую применяется инструкция MOV. Её синтаксис: MOV <пункт назначения>, <источник> [2].

Для управления набором данных различных типов можно использовать структуры. Структура – это тип данных, состоящий из фиксированного числа элементов разных типов. Для объявления структуры используется ключевое слово STRUC. Сначала необходимо описать шаблон будущей структуры:

имя_структуры STRUC

<описание полей>

имя_структуры ENDS.

Здесь <описание полей> представляет собой последовательность директив описания данных db, dw, dd, dp, dq и dt. Их операнды определяют размеры полей и, при необходимости, начальные значения. Эти значения будут использованы для инициализации соответствующих полей при определении структуры. Чтобы обратиться к полю структуры, используется конструкция: <имя переменной структуры> <имя поля> [3].

Для записи нескольких элементов одного типа используются массивы. Массив – это структурированный тип данных, состоящий из некоторого числа элементов одного типа [4]. В языке ассемблера отсутствуют специальные средства описания массивов. Если нужно использовать массив в программе, его следует моделировать одним из следующих способов:

1. Перечисление элементов массива в поле операндов одной из директив описания данных. Элементы перечисляются через запятую. Например: массив из 5 элементов, каждый размером 4 байта:

Array1 dd <1,2,3,4,5>

2. Использование оператора повторения dup. Например: массив из 5 элементов, каждый равен нулю, размером 2 байта:

Array2 dw 5 dup (0)

3. Использование цикла для инициализации значений области памяти, которую можно трактовать как массив.

Для проверки условий применяются команды условных и безусловных переходов. Команды условных переходов позволяют передать управление процессору к указанной команде в зависимости от выполнения некоторых условий, определяемых состоянием флагов. Команда безусловного перехода передаёт управление всегда (без учёта условий). Синтаксис команд условного и безусловного переходов: <команда_перехода> <адресное_выражение> [5].

Данные команды представлены в таблице 1.

Таблица 1

Команды условных и безусловных переходов

Мнемоническое обозначение	Условие перехода	Проверяемые значения флагов
JMP	безусловный переход	Нет
JA / JNBE	больше (для беззнаковых чисел)	CF=0 и ZF=0
JAЕ / JNB	больше или равно (для беззнаковых чисел)	CF=0
JB / JNAE	меньше (для беззнаковых чисел)	CF=1
JBE / JNA	меньше или равно (для беззнаковых чисел)	CF=1 или AF=1
JC	перенос	CF=1
JCXZ	CX равно нулю	нет
JE / JZ	равно	ZF=1
JG / JNLE	больше (для знаковых чисел)	ZF=0 и SF и OF одинаковы (оба 0 или 1)

JGE / JNL	больше или равно (для знаковых чисел)	SF и OF одинаковы (оба 0 или 1)
JL / JNGE	меньше (для знаковых чисел)	SF и OF различны
JLE / JNG	меньше или равно (для знаковых чисел)	ZF=1 и SF и OF различны
JO	переполнение	OF=1
JNO	нет переполнения	OF=0
JP / JPE	паритет четный	PF=1
JNP / JPO	паритет нечетный	PF=0
JS	знак	SF=1
JNS	нет знака	SF=0
JNZ	не ноль	ZF=0

2. Постановка задачи

Необходимо написать программу, которая должна верно обрабатывать массив структур В и определять количество книг с количеством страниц более 200 и запишет это число в регистр DI.

2.1 Исходные данные

В качестве исходных данных будет представлен массив структур В, который включает в себя информацию о книгах: фамилия автора, название книги, количество страниц и цена.

2.2 Набор выполняемых функций

Программа будет иметь две основные задачи: выбор структур, соответствующих определенному условию, и сохранение количества выбранных структур в регистр.

3. Выбор структур данных для решения поставленной задачи

Для решения задачи будут использоваться структуры данных, представленные в табл. 2.

Структуры данных

Наименование	Обозначение	Тип данных
Books	Book STRUC	Структура
B	B Book <>	Массив структур
author_name	db 50 DUP (?)	Массив байтов
book_title	db 50 DUP (?)	Массив слов
page_count	dw ?	Двойное слово
price	dw ?	Двойное слово

Переменная Books представляет структуру с данными о книге.

B – это массив, предназначенный для хранения структур.

Переменные author_name, book_title, page_count и price хранят в себе информацию о книгах: фамилия автора, название книги, количество страниц и цена.

4. Логическое проектирование

Дана структура с полями: фамилия автора, название книги, количество страниц, цена и массив типа структуры, где собрана информация о 10 книгах. Найти и вывести количество книг с объемом более 200 страниц.

1. Инициализация массива структур:

- Объявление структуры, содержащей информацию о книгах: фамилия автора, название книги, количество страниц и цена;
- задание характеристик каждой книги в соответствии с этой структурой и помещение их в массив.

2. Итерация по массиву книг:

- установка счетчика цикла таким образом, чтобы он содержал количество элементов в массиве, которое в данном случае равно 10;
- установка счётчика структур на начало массива для доступа к элементам массива книг.

3. Проверка на наличие более 200 страниц:

- чтение значения количества страниц из текущей книги;
- если книга содержит более 200 страниц, увеличение счётчика книг на 1.

4. Инкрементация счетчика структур:

- переход к следующему элементу массива структур.

5. Завершение программы:

- выход из цикла, когда все элементы массива книг проверены.

Блок-схема программы представлена на рис. 1.

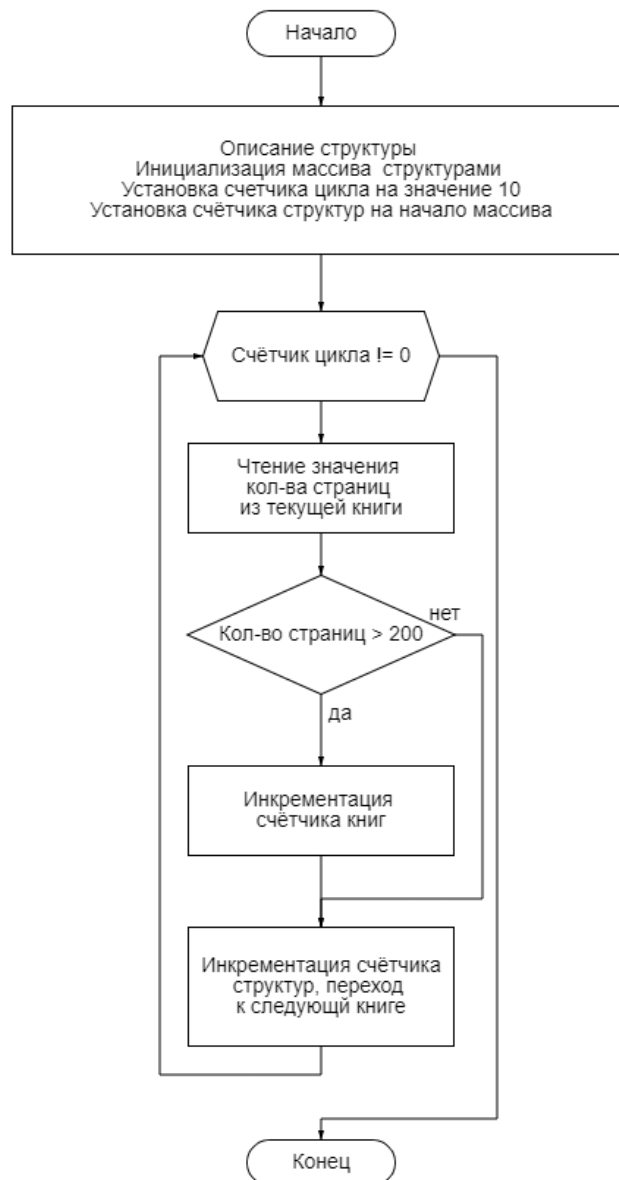


Рис. 1. Блок-схема программы

5. Физическое проектирование

Шаблон структуры состоит из последовательности следующего набора команд:

```
Books STRUC
author_name db 50 DUP (?)
book_title db 50 DUP (?)
page_count dw ?
price dw ?
Books ENDS
```

Где Books – название структуры, author_name – поле, которое хранит фамилию автора, book_title – поле, которое хранит название книги, page_count – поле, которое хранит количество страниц, price – поле, которое хранит цену книги.

Массив В представлен следующим видом: В Books <...>. Где В – название массива, Books – тип данных в массиве, а именно структура, <...> – инициализация элементов массива В, в котором записываются значения полей структуры через запятую.

Регистр CX используется как счётчик для цикла.

Регистр SI используется для навигации по массиву путем перехода к следующей структуре. Это достигается путем увеличения его значения на размер структуры с помощью операции ADD.

6. Проектирование интерфейса

Пользователь взаимодействует с GUI TurboAssembler, предоставляя необходимые данные в кодовом файле. После компиляции программа обрабатывает эти данные, и результаты доступны в скомпилированной программе, конкретно в регистре DI.

Формой для ввода данных является код, где пользователь может изменять информацию о книгах по своему усмотрению.

Результаты выводятся в регистр DI, который содержит количество структур, соответствующих условиям, заданным пользователем.

7. Кодирование

В данном разделе приведена последовательность команд, которые использовались для определения числа книг с количеством страниц более 200.

1. Объявление структуры Books и инициализация массива B:

```
.model small
.stack 100h
.data
Books STRUC
author_name db 50 DUP (?)
book_title db 50 DUP (?)
page_count dw ?
price dw ?
Books ENDS
```

```
B Books <" Lee", "To Kill a Mockingbird", 200, 345>, <"Orwell", "1984", 89, 1100>, <"Palanik", "Fight Club",
123, 250>, <"Tolkien", "The Hobbit", 300, 150>, <"Brown", "Da Vinci Code", 253, 300>, <"Rowling", "Harry Potter
and the Philosopher's Stone", 112, 250>, <" Tolkien", "The Lord of the Rings", 543, 369>, <"Fitzgerald", "The Great
Gatsby", 321, 281>, <"Salinger", "The Catcher in the Rye", 124, 1120>, <"Austen", "Pride and Prejudice", 300, 500>
```

2. Инициализация регистров для корректного доступа программы к данным, особенно к массиву структур B, содержащему информацию о книгах:

```
.code
ORG 100h
start:
mov ax, @data
mov DS, AX
xor DI, DI
mov CX, 10
mov SI, OFFSET B
```

3. Этот участок кода проходит по массиву B. Для каждой структуры извлекается значение page_count, которое проверяется на превышение 200, и в случае соответствия этому условию, счетчик DI увеличивается. Затем

происходит переход к следующей структуре в массиве, и этот процесс повторяется, пока все структуры не будут просмотрены:

```
iterate:
mov AX, [SI + Books.page_count]
cmp AX, 200
jbe not_more_than_200
inc DI
not_more_than_200:
add SI, TYPE Books
loop iterate
mov ah, 4Ch
INT 21h
END start
```

8. Тестирование

Для проверки работоспособности программы и нахождения возможных ошибок было проведено тестирование, в результате было подтверждено, что программа работает исправно, и не было обнаружено никаких проблем.

Тестовые данные представлены в табл. 3.

Таблица 3

Тестовые данные

Исходные данные	Тестируемый модуль	Ожидаемые результаты
Объявление шаблона структуры	Books.asm	Корректное объявление шаблона структуры
Объявление и инициализация массива структур	Books.asm	Корректная инициализация массива структур.
Обработка массива, запись значения из структуры в регистр	Books.asm	Корректная обработка массива, корректная запись значения из структуры в регистр

Переход между структурами в массиве	Books.asm	Корректный переход между структурами в массиве
Установка меток для цикла	Books.asm	Корректный переход между метками
Подсчёт количества верных структур	Books.asm	Корректный подсчёт количества верных структур
Запись конечного числа в необходимый регистр	Books.asm	Корректная работа записи ответа в необходимый регистр

Результаты тестирования представлены в табл. 4.

Таблица 4

Результаты тестирования

Дата тестирования	Тестируемый модуль	Кто проводил тестирование	Описание теста	Результаты тестирования
23.12.2023	Books.asm	Овчинников М.В.	Корректное объявление шаблона структуры	Успех
23.12.2023	Books.asm	Овчинников М.В.	Корректная инициализация массива структур	Успех
23.12.2023	Books.asm	Овчинников М.В.	Корректная обработка массива, корректная запись значения из структуры в регистр	Успех
23.12.2023	Books.asm	Овчинников М.В.	Корректный переход между структурами в массиве	Успех
23.12.2023	Books.asm	Овчинников М.В.	Корректный переход между метками	Успех

23.12.2023	Books.asm	Овчинников М.В.	Корректный подсчёт при сравнении	Успех
23.12.2023	Books.asm	Овчинников М.В.	Корректный подсчёт количества верных структур	Успех
23.12.2023	Books.asm	Овчинников М.В.	Корректная работа записи ответа в необходимый регистр	Успех

Заключение

В результате выполнения курсовой работы была создана программа, предназначенная для подсчета количества книг с более чем 200 страницами. Эта программа соответствует требованиям технического задания: она анализирует структуры и записывает окончательный результат в указанный регистр.

В процессе выполнения проекта были освоены навыки программирования на языке низкого уровня - *Assembler*. Эти навыки представляют ценность для дальнейшего обучения в области программирования, так как дают понимание работы высокоуровневых языков программирования и процессов программирования на машинном уровне.

Список литературы

1. Директивы определения данных [Электронный ресурс]. Дата доступа: 25.12.2023. Ссылка: <https://studfile.net/preview/16566533/page:7/>
2. Руководство по ассемблеру x86 для начинающих [Электронный ресурс]. Дата доступа: 25.12.2023. Ссылка: <https://habr.com/ru/articles/423077/>
3. Ассемблер Intel x86-64 Структуры [Электронный ресурс]. Дата доступа: 25.12.2023. Ссылка: <https://metanit.com/assembler/tutorial/3.11.php>
4. Ассемблер Intel x86-64 массивы [Электронный ресурс]. Дата доступа: 25.12.2023. Ссылка: <https://metanit.com/assembler/tutorial/3.9.php>
5. Учебный курс. Часть 16. Условные и безусловные переходы [Электронный ресурс]. Дата доступа 25.12.2023. Ссылка: <https://fasmworld.ru/uchebnyj-kurs/016-uslovnye-i-bezuslovnye-perexody/>
6. Turbo Assembler - Wikipedia [Электронный ресурс]. Дата доступа 25.12.2023. Ссылка: https://en.m.wikipedia.org/wiki/Turbo_Assembler
7. DOSBox — Википедия [Электронный ресурс]. Дата доступа 25.12.2023. Ссылка: <https://ru.wikipedia.org/wiki/DOSBox>
8. Е.В. Ершов, Л.Н. Виноградова, В.В. Селивановских, О.Л. Селяничев «Методика и организация самостоятельной работы студентов» [Файл].

Техническое задание. Приложение 1

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

(наименование структурного подразделения)

Кафедра математического и программного обеспечения ЭВМ

(наименование кафедры)

Программирование на ассемблере

(наименование дисциплины в соответствии с учебным планом)

УТВЕРЖДАЮ
Зав. кафедрой МПО ЭВМ,
д.т.н., профессор Ершов Е.В.
«___» октября 2023 г.

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ НИЗКОГО УРОВНЯ

Техническое задание на курсовую работу

Листов 7

Руководитель: Виноградова Л. Н.
Исполнитель: студент гр. 1ПИБ-02-
2оп-22
Овчинников М.В.

2024 год

Введение

Цель данной курсовой работы заключается в изучении и применении концепции структур в языке программирования ассемблер. Программа будет исследовать создание и управление структурами, включая описание типов структур, создание переменных структурного типа и формирование массивов структур для оптимального использования памяти.

1. Основания для разработки

Разработка ведётся на основании задания на курсовую работу по дисциплине «Программирование на ассемблере», выданное на кафедре МПО ЭВМ ИИТ ЧГУ.

Дата утверждения: 10 ноября 2023 года

Тема разработки: «Программирование на языке низкого уровня»

2. Назначение разработки

Основной задачей данной работы является практическое применение знаний, полученных при изучении дисциплины "Программирование на ассемблере", а также при освоении работы с программой-эмулятором процессора Intel 8086 "emu8086".

3. Требования к программе

3.1. Требования к функциональным характеристикам

Программа должна быть способна работать с структурой Books, которая включает в себя информацию о книгах: фамилию автора, название книги, количество страниц, цену и массив В с данными о 10 книгах. Основная задача программы - определить количество книг в структуре Books, у которых

количество страниц превышает 200, и передать это число в регистр DI для последующего использования.

3.2. Требования к надёжности

Программа должна обеспечивать надежную обработку структуры данных, сохраняя её целостность и создавая новые структуры с обработанными результатами. Необходимо гарантировать, что данные в структуре остаются неповрежденными и не подвергаются случайному удалению или изменению в процессе обработки. Важно, чтобы программа демонстрировала стабильную работу и способность корректно завершать свою работу даже при возможных сбоях или непредвиденных обстоятельствах.

3.3. Условия эксплуатации

Эксплуатационные условия программы должны включать в себя работу на совместимых средах, поддерживающих функциональность ассемблера. Программа должна корректно функционировать на подходящих для ассемблерной среды операционных системах. Также важно, чтобы входные данные соответствовали ожидаемому формату и были представлены для обработки в соответствии с требованиями программы.

3.4. Требования к составу и параметрам технических средств

Технические минимальные требования:

- оперативная память: не менее 128 Кб;
- процессор: архитектура x86-16 или совместимая;
- свободное место на диске: не менее 1 Мбайта;
- монитор для отображения программы;
- мышь и клавиатура для ввода данных и взаимодействия с программой.

3.5. Требования к информационной и программной совместимости

Для обеспечения совместимости программы требуется наличие операционной системы Windows 7, 8, 10, 11. Также необходимо установить ПО, такое как emu8086 или другой компилятор Assembler, чтобы разрабатывать и запускать программу на языке ассемблера.

3.6. Требования к маркировке и упаковке

Обычно требования к маркировке и упаковке не применимы к программе, поскольку она является цифровым продуктом, распространяемым в электронном формате.

3.7. Требования к транспортированию и хранению

Для правильной работы программы необходимо расположить соответствующие файлы на флеш-накопителе или в памяти компьютера. Рекомендуется сохранить программу на внешнем носителе, чтобы предотвратить потерю информации.

3.8. Специальные требования

Отсутствуют.

4. Требования к программной документации

4.1. Содержание расчётно-пояснительной записки:

- титульный лист;
- оглавление;
- введение;
- описание предметной области;
- описание разработки;
- описание созданной программы;
- заключение;

- источники;
- приложения.

4.2. Требования к оформлению

Требования к оформлению, установленные ГОСТ, должны быть выполнены на протяжении всей работы без каких-либо изменений (в табл. П1.1).

Таблица П1.1

Требования к оформлению

Документ	Печать на отдельных листах формата А4 (210х297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б. Файлы предъявляются на компакт-диске: РПЗ с ТЗ; программный код. Листы и диск в конверте вложены в пластиковую папку скоросшивателя.
Страницы	Печать на отдельных листах формата А4 (210х297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б. Файлы предъявляются на компакт-диске: РПЗ с ТЗ; программный код. Листы и диск в конверте вложены в пластиковую папку скоросшивателя.
Абзацы	Ориентация – книжная; отдельные страницы, при необходимости, альбомная. Поля: верхнее, нижнее – по 2 см, левое – 3 см, правое – 1 см.
Шрифты	Межстрочный интервал – 1.5, перед и после абзаца – 0. Кегль – 14. В таблицах шрифт 12. Шрифт листинга – 10 (возможно в 2 колонки).
Рисунки	Подписывается под ним по центру: Рис.Х Название В приложениях: Рис.П1.3. Название
Таблицы	Подписывается: над таблицей, выравнивание по правому: «Таблица Х». В следующей строке по центру Название Надписи в «шапке» (имена столбцов, полей) – по центру. В теле таблицы (записи) текстовые значения – выровнены по левому краю, числа, даты – по правому.

5. Стадии и этапы разработки

Стадии и этапы разработки представлены в таблице П1.2.

Таблица П1.2

Стадии и этапы разработки

Наименование этапа разработки ПО	Сроки разработки	Результат выполнения	Отметка о выполнении
Определение темы курсовой работы	10.11.23	Утверждена тема для разработки	
Оформление технического задания	17.11.23	Выполнение технического задания	
Разработка алгоритма	01.12.2023	Готовый алгоритм	
Написание программы	20.12.2023	Написанная программа	
Тестирование программы	25.12.2023	Проверенная и отлаженная программа	
Оформление РПЗ	25.12.2023	Написание РПЗ	

6. Порядок контроля и приёмки

Порядок контроля и приёма представлены в таблице П1.3

Таблица П1.3

Порядок контроля и приёма

Наименование контрольного этапа выполнения курсовой работы	Сроки контроля	Результат выполнения	Отметка о приёмке результата контрольного этапа
Техническое задание	17.11.23	Оформленное техническое задание	
Теоретическая часть курсовой работы	06.12.2023	Оформленная теоретическая часть	
Практическая часть курсовой работы	07.12.2023	Программа	
Расчётно- пояснительная записка	23.12.2023	Оформленная РПЗ	
Защита курсовой работы	26.12.2023	Итоговая оценка за курсовую работу	

1. Общие сведения о программе

Эта программа анализирует информацию о книгах, выявляя их количество, объём страниц которых превышает 200. Он использует массив структур, где каждая структура представляет данные об одной книге.

2. Описание установки

Для запуска программы необходимо иметь доступ к файлу программы и использовать программу GUI Turbo Assembler или любую другую, которая может использовать язык программирования Assembler. Также необходимо иметь программу DOSBox для компиляции кода.

Установка программы не требуется.

3. Описание запуска

Необходимо извлечь программу из носителя на компьютер, используя дисковод. Для просмотра и редактирования кода необходимо щёлкнуть по файлу Books.asm два раза или воспользоваться контекстным меню с помощью нажатия правой кнопкой мыши по файлу.

Для того чтобы отредактировать параметры книг, необходимо поменять значения в элементах массива <”фамилия автора“, “ название книги”, количество страниц, цена> в 14 строке (см. рис. П.2.1).

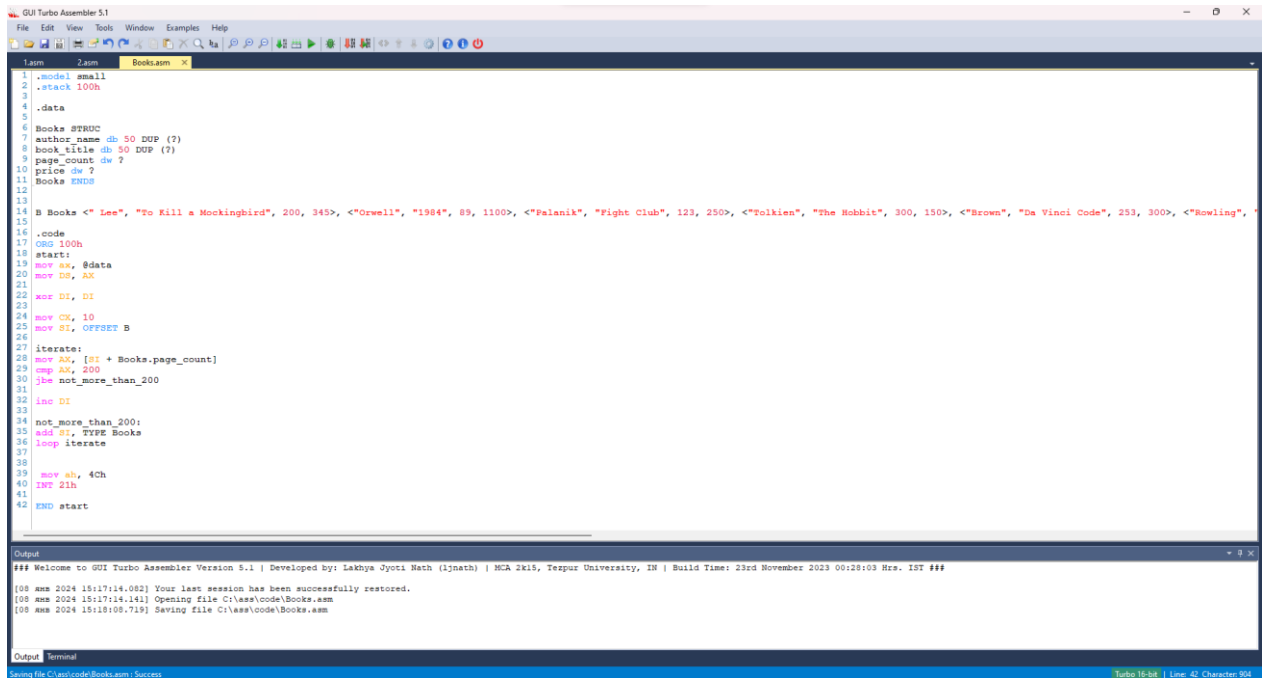


Рис. П.2.1. Программа, запущенная в GUI Turbo Assembler

Для того чтобы запустить и скомпилировать код необходимо воспользоваться программой DOSBox. Для начала необходимо запустить DOSBox и перейти в директорию code с помощью команды `cd code` (см. рис. П.2.2).

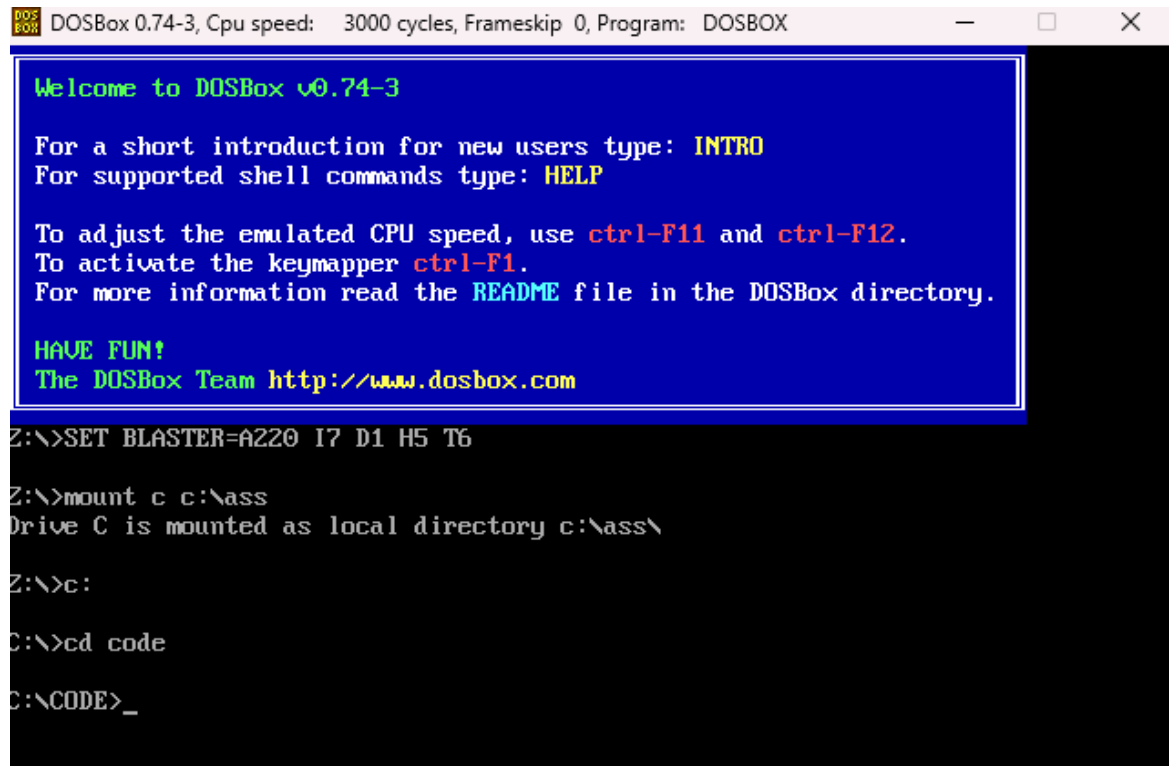


Рис. П.2.2. Запуск DOSBox и переход в директорию code

Затем нам необходимо ввести команду `c:\tasm\bin\tasm.exe Books.asm`. Через данную команду мы вызываем транслятор Turbo Assembler и он генерирует файл `Books.obj`, если в файле нет синтаксических ошибок (см. рис. П.2.3).

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\ass
Drive C is mounted as local directory c:\ass\

Z:\>c:

C:\>cd code

C:\CODE>c:\tasm\bin\tasm.exe Books.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:   Books.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  464k

C:\CODE>

```

Рис. П.2.3. Ввод команды `c:\tasm\bin\tasm.exe Books.asm`

Далее мы вводим команду `c:\tasm\bin\tlink.exe Books.obj`. Данная команда вызывает компоновщик, который компоует `Books.obj` файл в `Books.exe` (см. рис. П.2.4).

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\ass
Drive C is mounted as local directory c:\ass\

Z:\>c:

C:\>cd code

C:\CODE>c:\tasm\bin\tasm.exe Books.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:   Books.asm
Error messages:   None
Warning messages: None
Passes:           1
Remaining memory: 464k

C:\CODE>c:\tasm\bin\tlink.exe Books.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\CODE>

```

Рис. П.2.4. Ввод команды c:\tasm\bin\tlink.exe Books.obj

В конце мы вводим команду c:\tasm\bin\tl.exe Books.exe. Данная команда вызывает отладчик в DOSBox (см. рис. П.2.5).

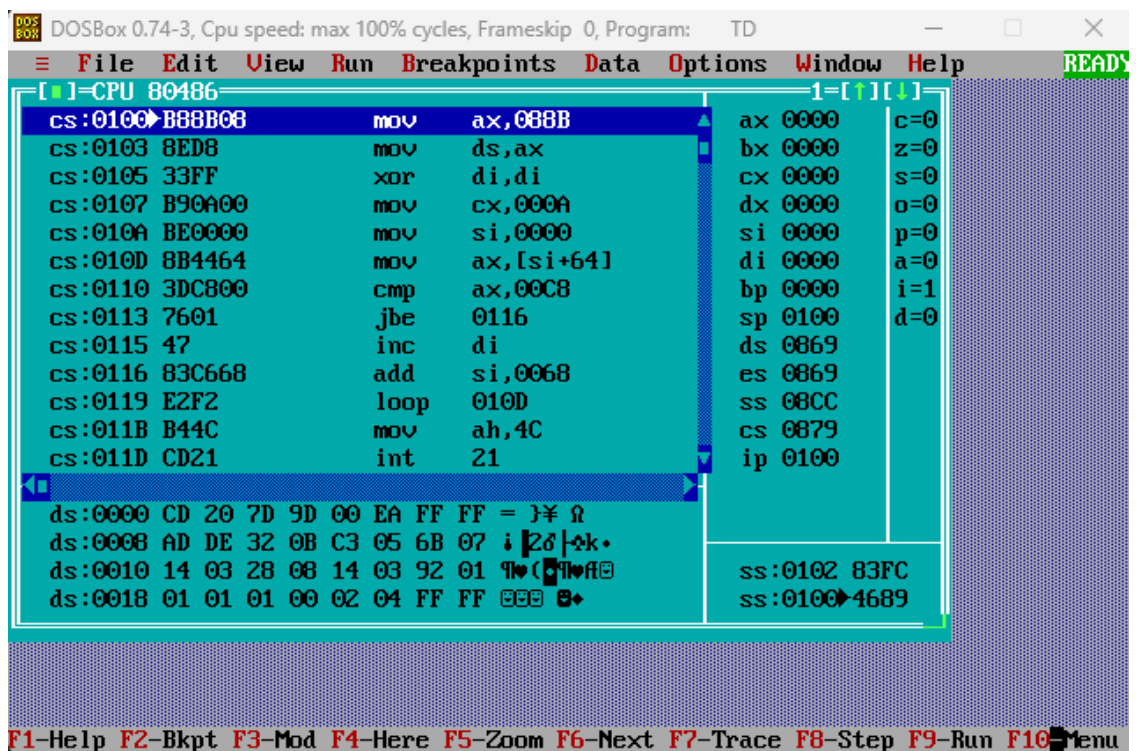


Рис. П.2.5. Вызов отладчика в DOSBox

При нажатии клавиши F8 будет выполнен шаг программы, при нажатии F9 будет выполнена вся программа. Результат можно увидеть в регистре DI (см. рис. П.2.6).

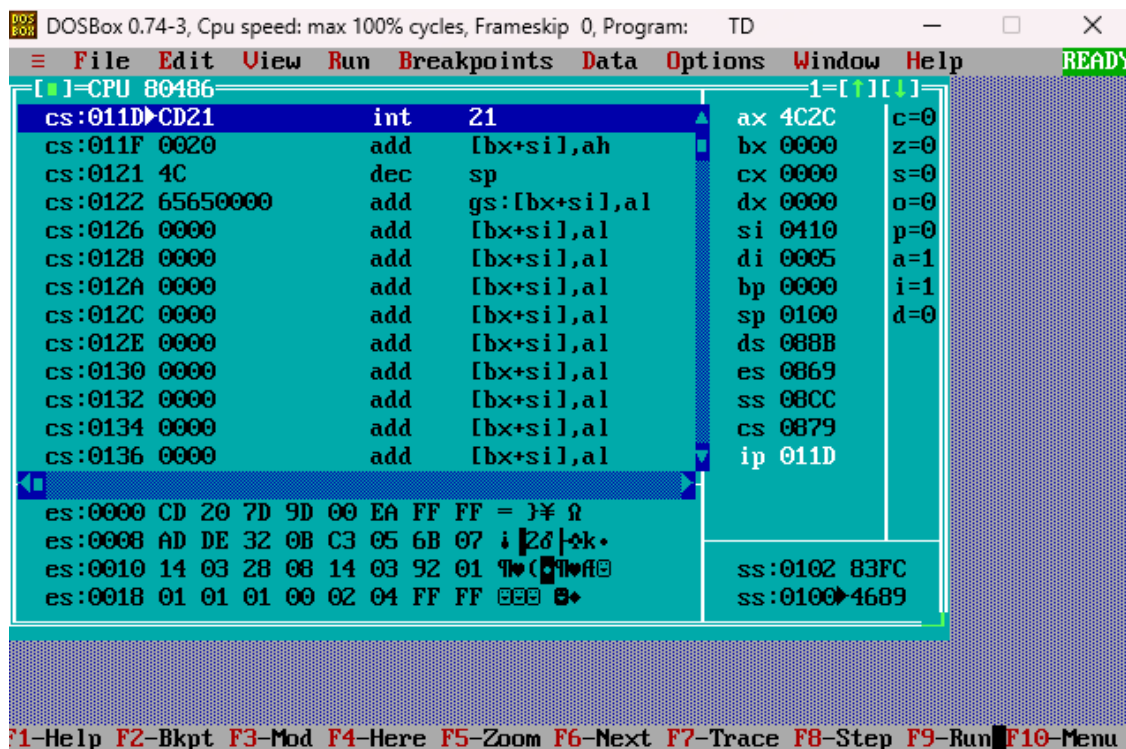


Рис. П.2.6. Результат выполнения программы

Файл Books.asm:

```

.model small
.stack 100h
.data
Books STRUC
author_name db 50 DUP (?)
book_title db 50 DUP (?)
page_count dw ?
price dw ?
Books ENDS
B Books <" Lee", "To Kill a Mockingbird", 200, 345>, <"Orwell", "1984", 89, 1100>, <"Palanik", "Fight Club", 123,
250>, <"Tolkien", "The Hobbit", 300, 150>, <"Brown", "Da Vinci Code", 253, 300>, <"Rowling", "Harry Potter and the
Philosopher's Stone", 112, 250>, <" Tolkien", "The Lord of the Rings", 543, 369>, <"Fitzgerald", "The Great Gatsby",
321, 281>, <"Salinger", "The Catcher in the Rye", 124, 1120>, <"Austen", "Pride and Prejudice", 300, 500>
.code
ORG 100h
start:
mov ax, @data
mov DS, AX
xor DI, DI
mov CX, 10
mov SI, OFFSET B
iterate:
mov AX, [SI + Books.page_count]
cmp AX, 200
jbe not_more_than_200
inc DI
not_more_than_200:
add SI, TYPE Books
loop iterate
mov ah, 4Ch
INT 21h
END start

```