МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Методические указания
к выполнению курсовой работы
«Проектирование транслятора для языка АССЕМБЛЕР»
для обучающихся очной формы обучения
направлений подготовки
09.03.01 «Информатика и вычислительная техника»,
09.03.04 «Программная инженерия»

Севастополь
СевГУ
2025

УДК [004.45:004.431.4]:001.891(076.5)
ББК 32.973.2в6я73
С409

Р е ц е н з е н т:
В. Ю. Карлусов – канд. техн. наук, доцент,
доцент кафедры «Информационные системы»

Составитель: Шалимова Е. М.

С409 Системное программирование: методические указания к выполнению курсовой работы «Проектирование транслятора для языка АССЕМБЛЕР» для обучающихся очной формы обучения направлений 09.03.01 «Информатика и вычислительная техника», 09.03.04 «Программная инженерия» / Севастопольский государственный университет, Институт информационных технологий; сост. Е. М. Шалимова. – Севастополь : СевГУ, 2025. – 27 с. – Текст : электронный.

Дисциплина «Системное программирования» включена в блок обязательных дисциплин учебного плана. Целью методических указаний к выполнению курсовой работы является изложение основных теоретических положений разработки системного программного обеспечения. В методических указаниях представлены требования к транслятору. Кроме того, приведены форматы одно- и двухадресных команд, а также порядок выполнения курсовой работы, варианты заданий и содержание пояснительной записки.

УДК [004.45:004.431.4]:001.891(076.5)
ББК 32.973.2в6я73

Рассмотрены и утверждены на заседании кафедры «Информационные технологии и компьютерные системы», протокол № 8 от 11 марта 2025 г.

СОДЕРЖАНИЕ

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Программное обеспечение (ПО) современных ЭВМ можно разделить на прикладное и специальное или системное. Прикладное ПО– это программы пользователей, системное – это операционные системы, компиляторы, трансляторы, отладчики, драйверы и т.д.

В рамках данной курсовой работы рассматривается задача разработки транслятора для подмножества команд языка Ассемблера IBM PC (процессор i8086/8088).

Любая программа, переводящая программу на исходном языке в выходную или объектную программу, называется транслятором. Если исходным языком является Ассемблер, то транслятор тоже часто называют ассемблером.

Язык Ассемблер является машинно-ориентированным языком, другими словами, для разработки ассемблерных программ от программиста требуется знание архитектуры компьютера, организации памяти, способов адресации и т.д. [1–4]. Упрощенно структурную организация компьютера можно представить абстрактной моделью, изображенной на рисунке 1.

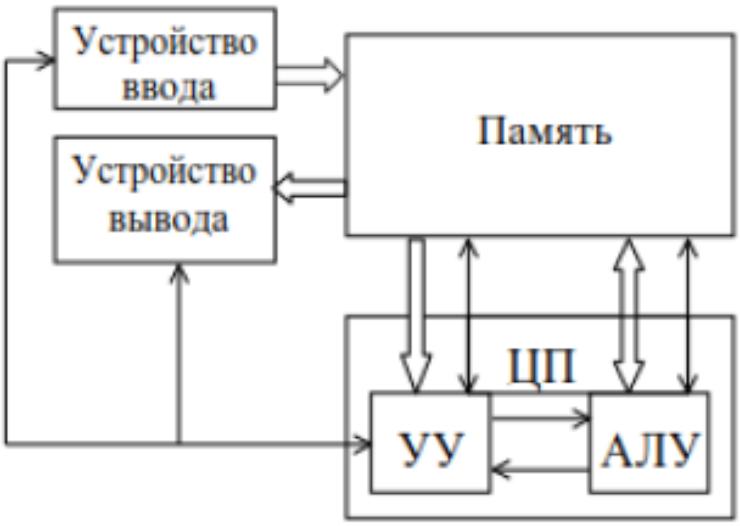


Рисунок 1 – Абстрактная модель ЭВМ

На рисунке 1 приняты следующие обозначения:
ЦП – центральный процессор;
УУ – устройство управления;
АЛУ – арифметико-логическое устройство;
⇒ – передача данных;
→ – передача управляющих сигналов.

Процессор *i8086/8088* имеет 14 регистров, каждый из которых характеризуется длиной в одно слово. Длина машинного слова – 2 байта или 16 бит. Всем регистрам присвоены имена, которые используются для обращения к регистрам. На рисунке 2 приведена классификация регистров.

Рисунок 2 – Классификация регистров процессора *i8086/8088*

Регистры общего назначения состоят из двух однобайтовых регистров, к которым можно обращаться независимо: AH, BH, CH, DH (старшие байты) и AL, BL, CL, DL (младшие байты).

Память компьютера представляется в виде линейной пронумерованной последовательности байтов. Каждый байт имеет адрес (номер), таким образом, байт – это минимальная адресуемая единица информации. Процессор может обращаться как к байтам, так и к словам памяти. Слово содержит два смежных байта памяти и может начинаться как с четного, так и нечетного адреса. При размещении слова в памяти байт с адресом, соответствующим адресу слова, содержит его младшую часть, а следующий байт – его старшую часть.

Логически память разбивается на сегменты размером по 64 Кбайт. Все пространство памяти, начиная с нулевого адреса, делится на параграфы – области из 16 смежных байт. Очевидно, что любой сегмент может начинаться только на границе параграфа (это означает, что четыре младших бита адреса – нулевые).

Физический или абсолютный адрес памяти представляется 20-битным числом и получается из двух составляющих: 20-битного адреса начала сегмента и 16-битного смещения внутри сегмента. Физический адрес любого байта памяти формируется сложением указанных составляющих так, как это представлено на рисунке 3.

Значение сегментного регистра (16 бит)	0 0 0 0

+

0 0 0 0	Смещение внутри сегмента (16 бит)

Физический или абсолютный адрес памяти (20 бит)

Рисунок 3 – Схема формирования абсолютного адреса

При таком способе формирования адреса сегмент всегда начинается с адреса, кратного 16.

Длина машинной команды составляет от 1 до 6 байт. Код команды занимает первый байт, способ адресации задается во втором байте, в остальных байтах команды указываются непосредственные данные или адрес памяти, где эти данные находятся.

Структура второго (адресного) байта представлена на рисунке 4.

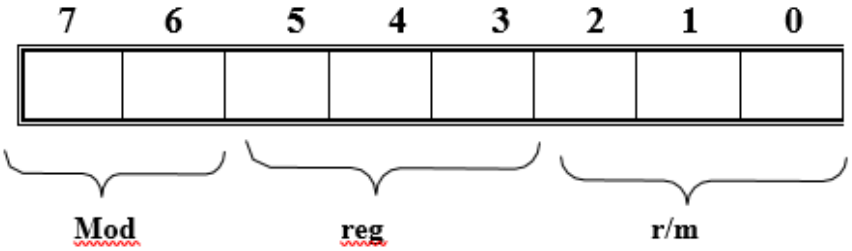


Рисунок 4 – Структура адресного байта

Байт кода команды содержит бит **w**, определяющий длину операндов команды:

w=0 – размер операндов байт;

w=1 – размер операндов слово.

Однако, положение бита **w** для разных команд может быть различным (рисунок 5).

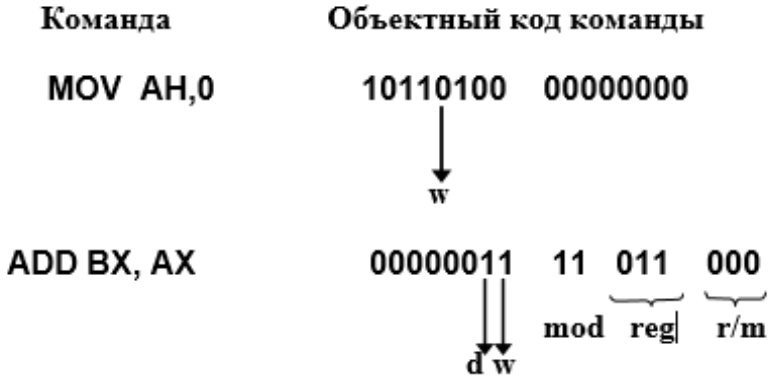


Рисунок 5 – Примеры положения бита **w** для разных команд

Кроме того, первый байт машинного кода может содержать бит **d**, который указывает направление потока данных между операндами. Если **d=0**, то поле **reg** определяет второй операнд, а поля **mod** и **r/m** – первый, иначе поле **reg** определяет первый операнд, а поля **mod** и **r/m**– второй. На рисунке 5, в последнем примере, **d=1.** Это означает, что поле **reg** задает первый операнд, а поля **mod** и **r/m** – второй.

Операнды команды кодируются полями **reg** или **mod, r/m**. Поле **reg** предназначено для указания операнда с регистровым режимом адресации и содержит код регистра. Поле **r/m** предназначено для указания операнда с регистровым режимом адресации или с режимом адресации оперативной памяти. Если биты поля **mod** имеют значение **11**, это значит, что операнд находится в регистре, т. е. биты **r/m** вместе с битом **w** определяют конкретный регистр. Кодирование регистров приведено в таблице 1.

Таблица 1– Таблица кодирования регистров

reg *или* ***r/m***		***000***	***001***	***010***	***011***	***100***	***101***	***110***	***111***
Регистр	***w =0***	***AL***	***CL***	***DL***	***BL***	***AH***	***CH***	***DH***	***BH***
	w=1	***AX***	***CX***	***DX***	***BX***	***SP***	***BP***	***SI***	***DI***

Если биты поля ***mod*** имеют значение ***00, 01 ,10*** то операнд находится в оперативной памяти, т. е. поле **r/m** содержит код способа адресации оперативной памяти. Кодирование типов адресаций приведено в таблице 2. Значения битов ***mod*** имеют следующую интерпретацию:

mod=00 – биты ***r/m*** задают абсолютный адрес, байт смещения отсутствует;

mod=01 – биты ***r/m*** задают абсолютный адрес памяти и имеется один байт смещения;

mod=10 – биты ***r/m*** задают абсолютный адрес памяти и имеется два байта смещения.

.

Таблица 2– Таблица кодов способов адресации

mem \ mod	***00***	***01***	***10***
000	***[BX]+[SI]***	***[BX]+[SI] +a8***	***[BX]+[SI] +a16***
001	***[BX]+[DI]***	***[BX]+[DI] +a8***	***[BX]+[DI] +a16***
010	***[BP]+[SI]***	***[BP]+[SI] +a8***	***[BP]+[SI] +a16***
011	***[BP]+[DI]***	***[BP]+[DI] +a8***	***[BP]+[DI] +a16***
100	***[SI]***	***[SI] +a8***	***[SI] +a16***
101	***[DI]***	***[DI] +a8***	***[DI] +a16***
110	***a16***	***[BP] +a8***	***[BP] +a16***
111	***[BX]***	***[BX] +a8***	***[BX] +a16***

В объектном коде команды, если используется непосредственный операнд длиной в слово, сначала указывается младший байт, а затем – старший (рисунок 6). В командах с одним операндом или в командах, когда второй операнд является непосредственным значением, а первый определяет память, первый операнд кодируется полями ***mod*** и ***r/m***, а поле ***reg*** содержит дополнительные 3 бита кода операции (рисунок 7).

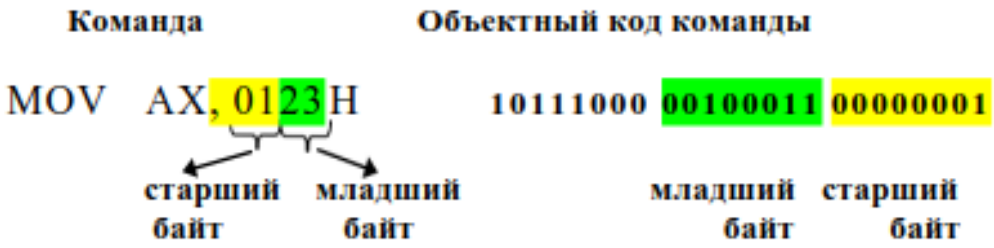


Рисунок 6 – Объектный код команды ***MOVAX,0123H***

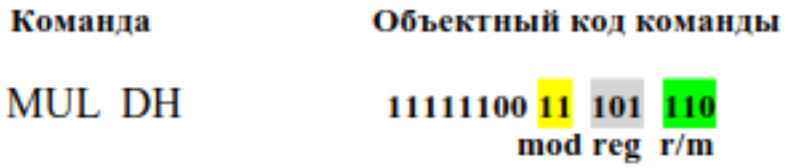


Рисунок 7 – Объектный код команды ***MUL DH***

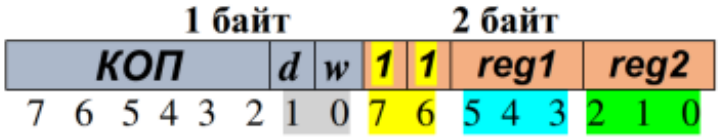
Описание основных форматов команд приведено в п.2 настоящих методических указаний. Полное описание команд процессора *i8086/8088* можно найти в [1,2,4].

2. ФОРМИРОВАНИЕ ОБЪЕКТНОГО КОДА КОМАНД

Рассмотрим основные форматы двухадресных и одноадресных команд.

2. 1. Форматы двухадресных команд

1) Формат ***"регистр-регистр"*** (2байта):



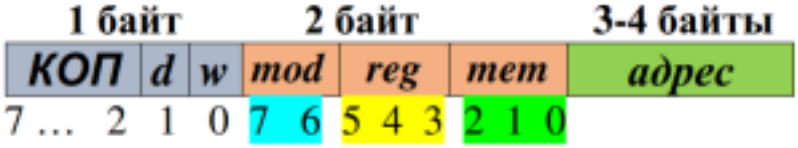
Команды этого формата задают действие:

reg1:=reg1 ОП reg2 *или*
reg2:=reg2 ОП reg1.

Поле ***КОП*** (код операции) первого байта указывает операцию (**ОП**).
Бит ***w*** определяет размер операндов:
1 , слово ; 0 , w = байт.
Бит ***d*** указывает, в какой из регистров записывается результат:
d =1 , reg1:=reg1 **ОП** ***reg2***
d =0 , reg2:=reg2 **ОП** ***reg1***

Во втором байте два левых бита фиксированы ***для данного формата (11)***, а поля ***reg1 и reg2*** указывают на регистры, коды которых приведены в таблице 1.

2) Формат "***регистр-память***" (2-4 байта):



Команды этого формата задают действие:

reg:=reg ОП mem *или*
mem:= mem ОП reg.

Биты ***w*** и ***d*** имеют такой же смысл, как и в предыдущем случае.
Поле ***reg*** *(3 бита)* определяет операнд-регистр.
Поле ***mod*** *(2 бита)* определяет, сколько байтов в команде занимает ***операнд-смещение***: ***00*** - **0** байтов, ***01*** - **1** байт, ***10*** - **2** байта.
Поле ***mem*** *(3 бита)* указывает способ модификации адреса памяти (таблица 2).

3) Формат "***регистр - непосредственный операнд*** " (3-4 байта):

Команды этого формата задают действие:

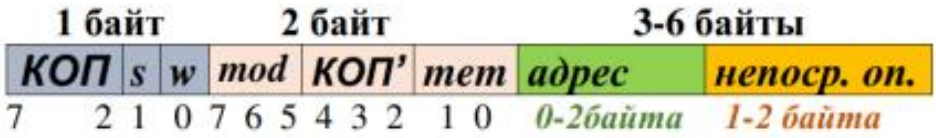
reg:=reg ОП immed .

Бит ***w*** задает размер операндов, а поле ***reg*** - регистр-операнд.

Поле ***КОП*** (*байт 1***)** определяет ***класс операции*** (например, класс сложения), уточняет же операцию поле ***КОП'*** (*байт 2***)**.

Непосредственный операнд может занимать **1** или **2** байта, операнд-слово записывается в команде в "***перевернутом***" виде. В ***операции над словами*** непосредственный операнд может быть задан ***байтом*** (***s=1*** *при* ***w=1***), и тогда перед выполнением операции байт автоматически расширяется (со знаком) до слова.

4) Формат "***память - непосредственный операнд*** " (3-6 байтов):



Команды этого формата задают действие:

mem:=mem ОП immed .

Смысл всех одноименных полей такой же, как и в предыдущих форматах.

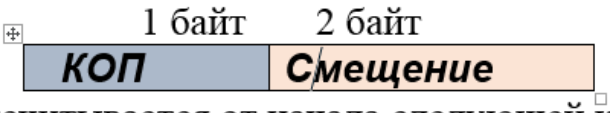
Для команд сдвига на месте бита ***s*** находится бит ***c***. Значение бита ***c*** определяется по следующему правилу:

равен ***0,*** если выполняется сдвиг на 1 разряд,

равен ***1***, если выполняется сдвиг на несколько разрядов (количество определяется значением в регистре ***CL*****)** .

2. 2. Форматы одноадресных команд

1) Команды перехода



Смещение отсчитывается от начала следующей команды

Команды перехода изменяют значение счетчика адреса (регистр ***IP***):
IP:= IP+Смещение .

2) Команды умножения, деления, отрицания, изменение знака числа

Команды умножения и деления задают действие:
Аккумулятор:= Аккумулятор ОП операнд .

Команды отрицания, изменение знака числа задают действие
reg:= ОП reg или ***mem:= ОП mem.***

3. ТРЕБОВАНИЯ К ТРАНСЛЯТОРУ

Транслятор с языка Ассемблера или ассемблер относится к классу системных программ. Транслятор с языка Ассемблера должен:

- выявить ошибки в исходной программе;
- распределить память;
- перевести на машинный язык команды мнемокода и константы;
- сформировать объектный код программы;
- сформировать протокол трансляции (листинг).

Эти задачи трудно решить за один просмотр [2–4]. Поэтому, как правило, используются двухпросмотровые или двухпроходные ассемблеры. При первом просмотре ассемблер анализирует исходную программу и строит таблицу символических имен или идентификаторов (ТСИ), куда заносятся имена полей данных и меток, используемых в программе, и их относительные адреса.

Кроме того, при первом проходе подсчитывается длина объектного кода, но сам объектный код не генерируется. При втором просмотре, ассемблер, используя ТСИ, генерирует объектный код для каждой команды и объектный код программы заданного формата.

В ходе трансляции ассемблер использует постоянные и временные таблицы. Постоянные таблицы составляются при разработке ассемблера, а временные создаются в ходе трансляции. Примером постоянной таблицы может служить таблица команд, в которой перечислены все мнемонические коды команд, типы операндов и их машинные эквиваленты, или таблица сообщений об ошибках, в которой фиксируется тип ошибки и соответствующий текст для вывода в файл-листинг. Множество и структура таблиц зависят от особенностей машины, языка Ассемблер и от конкретной реализации транслятора.

Обязательной временной таблицей в любом ассемблере является ТСИ, содержащая все имена, определенные в исходной программе и их характеристики: тип, значение (адрес), длину и т.д. Значением любого имени в ассемблере является адрес.

Результатом работы транслятора с языка Ассемблера являются листинг и объектный код. Объектный код формируется только при отсутствии ошибок в тексте исходной программы, а листинг создается в любом случае. Листинг содержит не только исходный текст, но и, при отсутствии ошибок, машинный код команды в шестнадцатиричном формате, а также шестнадцатиричный адрес команды. Если же в исходном тексте команды есть ошибки, то объектный код команды не может быть сформирован, и листинг должен содержать подробную информацию об ошибках. Общая схема обработки данных транслятором приведена на рисунке 8. Стрелками на рисунке показаны информационные и логические связи между блоками транслятора.

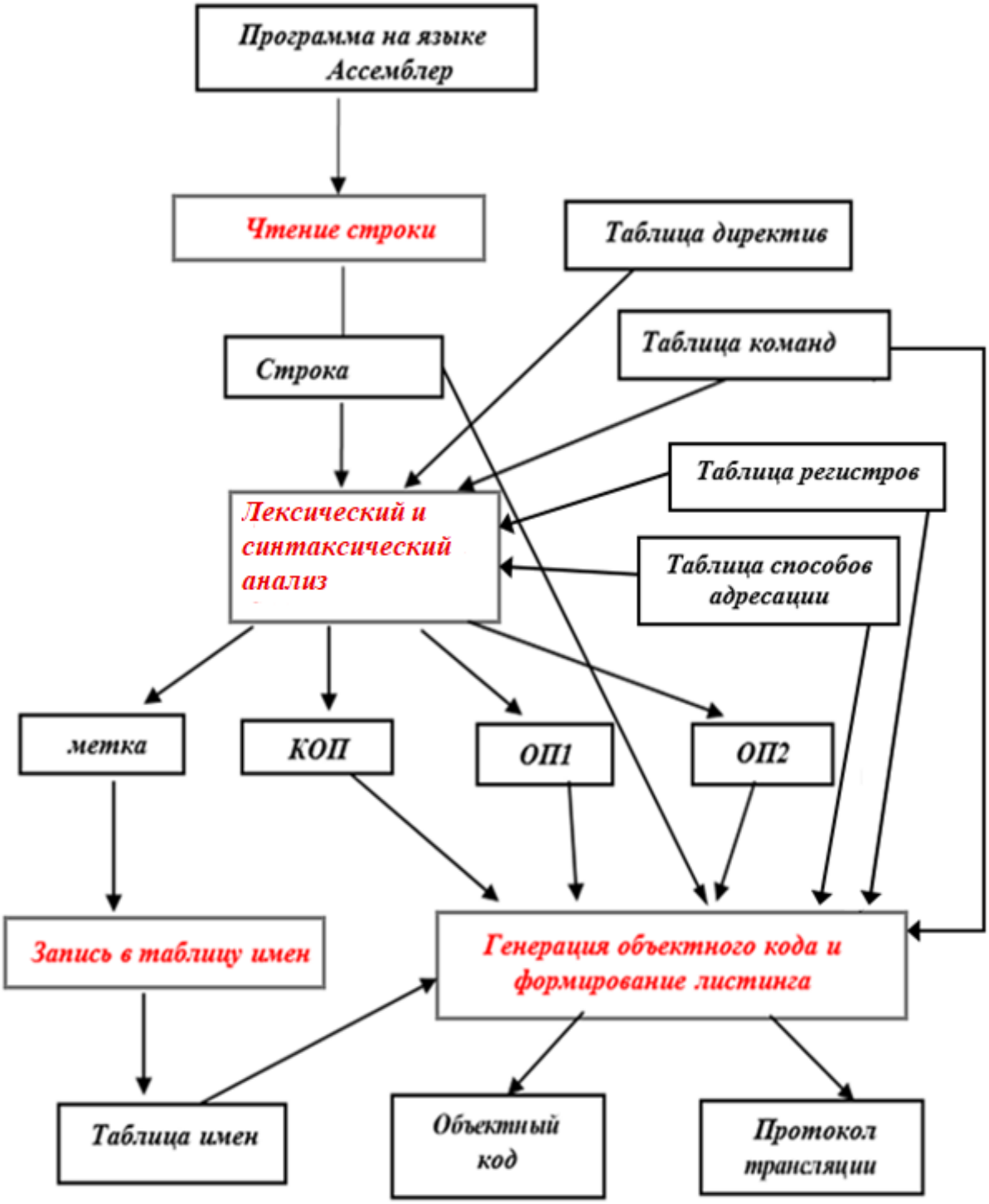


Рисунок 8 – Схема обработки данных транслятором

На рисунке используются следующие обозначения:
КОП – мнемонический код операции;
ОП1 и ***ОП2*** – операнды первый и второй соответственно.

4. ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Разработать программу транслятора для подмножества команд языка Ассемблера процессора *i8086/8088*. Транслируемая программа должна иметь следующий формат:

```
NAMESEG     SEGMENT
ORG         100H
...
область команд
...
INT 21H
NAMEDATA    DB
            DW
                ...
        область данных
                ...
NAMESEG     ENDS
END
```

В качестве признака завершения вычислений используется команда ***INT 21H***.

В результате работы транслятора должны быть сформированы протокол трансляции и объектный код программы.

Протокол трансляции должен представляться файлом-листингом, в котором содержится объектный код и мнемонический код команд, сообщения об ошибках, информация о результатах трансляции и т. д.

Объектный код программы должен быть записан в файл объектного кода и иметь следующий формат:

H запись-заголовок
T тело
E запись-конец

Здесь
запись-заголовок имеет формат:
H<Имя_сегмента><Начальный_адрес_загрузки><Длина_кода>
запись-тело:
T<Адрес_кода><Длина_кода><Код>
запись-конец:
E<Точка_входа>

Коды команд и данных следует записать в виде строки, представляющей шестнадцатеричную константу

Во всех вариантах реализовать обработку команды ***MOV*** и команд, определенных номером варианта (п. 6). Кроме того, во всех вариантах предусмотреть обработку директив:

1) ***SEGMENT, ENDS, END;***
2) ***ORG;***
3) ***DB, DW***.

5. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1) Получить вариант задания у преподавателя.

2) Разработать техническое задание (ТЗ) на проектирование в соответствии с вариантом. В ТЗ необходимо указать назначение программы, реализуемые ею функции, входные и выходные данные и их форматы, особые ситуации и реакцию разрабатываемой программы на них.

3) Изучить форматы представления команд микропроцессора *i8086 /8088*, определенных вариантом.

4) Разработать схему обработки данных. За основу можно взять схему, приведенную на рисунке 8.

5) Выбрать структуры данных. Указать типы, размеры, возможные значения для переменных, записей, таблиц и т.д.

6) Разработать алгоритм решения задачи. Допустимы любые способы представления алгоритма (словесное описание, представление в виде блок-схемы и т.д.)

7) Разработать и протестировать программу. Для тестирования необходимо определить перечень контролируемых ситуаций. В соответствии с этим перечнем, разработать тестовые примеры. В пояснительной записке привести все тесты с описанием проверяемых ими контрольных ситуаций и реакцию программы на них.

8) Оформить пояснительную записку. Правила оформления приведены в приложении А.

Содержание пояснительной записки

Введение

1. Постановка задачи.
2. Определение форматов команд.
3. Выбор и обоснование структур данных.
4. Описание алгоритма.
5. Описание программы.
   - 5.1. Общие сведения.
   - 5.2. Требования к программному и техническому обеспечению.
   - 5.3. Логическая структура программы.
   - 5.4. Используемые переменные и типы.
   - 5.5. Спецификация подпрограмм.
   - 5.6. Сообщения, выдаваемые программой.
   - 5.7. Вызов и загрузка программы.
6. Тестирование программы.

Заключение.

Библиографический список.

Приложения.

В пункте «Логическая структура программы» следует указать структуру программы с описанием функций составных частей (модулей, подпрограмм) и связи между ними.

Защита курсовой работы осуществляется в соответствии с графиком, который объявляется за неделю до начала защиты. ВНИМАНИЕ! За 3 дня до защиты работа должна быть зарегистрирована на кафедре и представлена на проверку преподавателю.

6. ВАРИАНТЫ ЗАДАНИЙ

Разрабатываемая программа (транслятор) должна обрабатывать следующие команды:

1) ***MOV***;

2) по одной команде из трех групп (согласно варианту).

Варианты заданий приведены в таблице 3, а группы команд −в таблице 4.
Режим адресации данных в оперативной памяти задан буквами:
П – прямая, ***К*** – косвенная, ***И*** – индексная, ***Б*** – базовая,
С – базово-индексная без смещения.

Описания и примеры указанных способов адресации памяти, которые должны быть реализованы, приведены в таблице 5.

Команда ***MOV*** и команда первой группы должны иметь четыре формата:

КОП Р, Р
КОП Р, НО
КОП Р, ОП
КОП ОП, Р
КОП ОП, НО

В описании команд используются следующие обозначения:
КОП − мнемонический код команды, ***Р*** − регистр общего назначения, ***НО*** − непосредственный операнд, ***ОП*** − оперативная память.

В команде второй группы операндом может быть регистр общего назначения (***Р***) или адрес оперативной памяти (***ОП***). Режим обращения к оперативной памяти такой же, как для команды первой группы.

Например, в варианте 2 задано следующее множество команд:

MOV Р, Р
MOV Р, НО
MOV Р, ОП(индексная)
MOV ОП(индексная), Р
MOV ОП(индексная), НО
ADD Р, Р
ADD Р, НО
ADD Р, ОП(индексная)
ADD ОП(индексная), Р
ADD ОП(индексная), НО
DEC ОП(индексная)
JP метка
INT 21H

Таблица 3 – Варианты заданий

№ п/п	Группа 1	Группа 2	Группа 3	№ п/п	Группа 1	Группа 2	Группа 3
1	1п	1р	2	41	11к	6р	13
2	2и	2оп	1	42	12с	7оп	14
3	3б	3р	3	43	13п	1р	15
4	4к	4оп	4	44	14и	2оп	1
5	5с	5р	5	45	15б	3р	2
6	6п	6оп	6	46	1б	4оп	4
7	7и	7р	7	47	2к	5р	5
8	8б	1оп	8	48	3с	6оп	6
9	9к	2р	9	49	4п	7р	7
10	10с	3оп	10	50	5и	1оп	8
11	11п	4р	11	51	6б	2р	9
12	12и	5оп	12	52	7к	3оп	10
13	13б	6р	13	53	8с	4р	11
14	14к	7оп	14	54	9п	5оп	12
15	15с	1р	15	55	10и	6р	13
16	1с	2оп	2	56	11б	7оп	14
17	2п	3р	3	57	12к	1р	15
18	3и	4оп	4	58	13с	2оп	1
19	4б	5р	5	59	14п	3р	2
20	5к	6оп	6	60	15и	4оп	3
21	6с	7р	7	61	1и	5р	5
22	7п	1оп	8	62	2б	6оп	6
23	8и	2р	9	63	3к	7р	7
24	9б	3оп	10	64	4с	1оп	8
25	10к	4р	11	65	5п	2р	9
26	11с	5оп	12	66	6и	3оп	10
27	12п	6р	13	67	7б	4р	11
28	13и	7оп	14	68	8к	5оп	12
29	14б	1р	15	69	9с	6р	13
30	15к	2оп	1	70	10п	7оп	14
31	1к	3р	3	71	11и	1р	15
32	2с	4оп	4	72	12б	2оп	1
33	3п	5р	5	73	13к	3р	2
34	4и	6оп	6	74	14с	4оп	3
35	5б	7р	7	75	15п	5р	4
36	6к	1оп	8	76	1п	6оп	10
37	7с	2р	9	77	2и	7р	11
38	8п	3оп	10	78	3б	1оп	12
39	9и	4р	11	79	4к	2р	13
40	10б	5оп	12	80	5с	3оп	14

Таблица 4 – Группы команд

№ п/п	Группа 1	Группа 2	Группа 3
1	*SUB*	*INC*	*JP*
2	*ADD*	*DEC*	*JNZ*
3	*ADC*	*NOT*	*LOOP*
4	*SBB*	*IMUL*	*JZ*
5	*CMP*	*MUL*	*JG*
6	*OR*	*IDIV*	*JGE*
7	*TEST*	*DIV*	*JE*
8	*XOR*		*JLE*
9	*XCHG*		*JA*
10	*AND*		*JAE*
11	*SHL*		*JB*
12	*SHR*		*JBE*
13	*SAR*		*JS*
14	*RCR*		*JC*
15	*RCL*		*JNE*

Таблица 5 – Примеры способов адресации памяти

Режим адресации	**Значения битов *mod, mem* в соответствии с таблицей 2**	**Способ выборки данных, формат записи операнда**
Прямая адресация	*mod=00, mem=110*	*EA* указан в команде. Примеры: [20H], [30], A
Косвенная адресация	*1) mod=00, mem=100* *2) mod=00, mem=101* *3) mod=01, mem=110* *4) mod=00, mem=111* *В 3-ем случае значение смещения равно 0*	*EA* указан в регистре. Примеры: [SI], [DI], [BP], [BX]
Базовая адресация	*1) mod=01, mem=110* *2) mod=01, mem=111* *3) mod=10, mem=110* *4) mod=10, mem=111*	*EA* формируется как сумма содержимого базового регистра и смещения, заданного в команде. Примеры: [Bx+20H] [Bx + A] [Bx]+20H A [Bx] [BP] + A [BP]20
Индексная адресация	*1) mod=01, mem=100* *2) mod=01, mem=101* *3) mod=10, mem=100* *4) mod=10, mem=101*	То же, что в пункте выше, только индексные регистры. Примеры: [SI + 20H] [DI] + A, [DI] + 20 [SI]+ 20H [DI]A, 20 [DI]
Базово-индексная адресация без смещения	*1) mod=00, mem=000* *2) mod=00, mem=001* *3) mod=00, mem=010* *4) mod=00, mem=011*	*EA* получается как сумма: (базовый регистр) + (индексный регистр) Примеры: [BX + SI], [BX + DI] [BP][SI] [BP][DI]

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Абель, П. Ассемблер. Язык и программирование для IBM PC / П. Абель. – Минск: КОРОНА-Век, 2009. – 736с.
2. Дао, Л. Программирование микропроцессора 8088 / Л. Дао – М.: Мир, 1988.– 355с.
3. Бек Л. Введение в системное программирование / Л. Бек – М.: Мир, 1988. – 448с.
4. Пильщиков, В.Н. Программирование на языке Ассемблера IBM PC/ В.Н. Пильщиков. – М.: Диалог – МИФИ, 2019. – 288 с.

ПРИЛОЖЕНИЕ А
ПРАВИЛА ОФОРМЛЕНИЯ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

А.1. Общие требования к оформлению

Пояснительную записку (ПЗ) оформляют машинным (при помощи компьютерной техники) способом на одной стороне листа формата А4 (210х297мм). При оформлении следует руководствоваться ГОСТ Р 7.0.11-2011, ГОСТ 7.32-2017 и ГОСТ Р 2.105-2019.

Допустимые параметры: ориентация страницы – книжная; поля: левое –20 мм, правое – 10 мм, нижнее – 20 мм, верхнее – 20 мм; шрифт Times New Roman, 14 пунктов (межстрочный интервал – полуторный); абзацный отступ должен быть одинаковым по всему тексту и равен 1,25 см, интервал перед абзацем и после абзаца – 0; выравнивание – по ширине.

А.2. Изображение структурных элементов

Все разделы и подразделы должны иметь заголовки, которые нумеруют в установленном ГОСТом порядке. Исключение составляют некоторые структурные элементы, такие как "СОДЕРЖАНИЕ", "ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ, СОКРАЩЕНИЙ И ТЕРМИНОВ", "ВВЕДЕНИЕ", "ЗАКЛЮЧЕНИЕ", "ПРИЛОЖЕНИЯ", "БИБЛИОГРАФИЧЕСКИЙ СПИСОК". Они не нумеруются, а их наименования служат заголовками структурных элементов.

Заголовки структурных элементов ПЗ и заголовки разделов выравниваются по центру. Их необходимо печатать прописными буквами без точки в конце, не подчеркивая. Если заголовок состоит из двух или более предложений, их разделяют точкой. Переносы слов в заголовке не допускаются. Основной текст должен быть выровнен по ширине листа.

Заголовки подразделов, пунктов и подпунктов ПЗ следует начинать с абзацного отступа и печатать строчными буквами, кроме первой прописной, не подчеркивая, без точки в конце.

Расстояние между заголовком текстом должно быть при машинном способе 1–2 строки.

Не допускается размещать наименование раздела, подраздела, а также пункта и подпункта в нижней части страницы, если после него расположена только одна строка текста.

А.3. Нумерация

Страницы ПЗ следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту отчета. Номер страницы проставляют в правом верхнем углу или в центре страницы без точки в конце. Титульный лист включают в общую нумерацию страниц, однако номер страницы на титульном листе не проставляется.

Разделы, подразделы, пункты, подпункты ПЗ следует нумеровать арабскими цифрами. Разделы должны иметь порядковую нумерацию в пределах изложения сути материала и обозначаться арабскими цифрами, например, 1, 2, 3 и т.д. Подразделы должны иметь порядковую нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и порядкового номера подраздела, разделенных точкой (например, 2.1). Пункты должны иметь порядковую нумерацию в пределах каждого раздела или подраздела. Номер пункта состоит из номера раздела и порядкового пункта, или из номера раздела, порядкового номера подраздела и порядкового номера пункта разделенных точкой, например, 1.1, 1.2 или 1.1.1 или 1.1.2 и т.д.

Иллюстрации и таблицы следует нумеровать арабскими цифрами порядковой нумерации в пределах раздела или использовать сквозную нумерацию, за исключением иллюстраций и таблиц, приводимых в приложениях. Если используется нумерация по разделам, то номер иллюстрации или таблицы состоит из номера раздела и порядкового номера иллюстрации или таблицы, разделенных точкой. Кроме того, каждый рисунок должен иметь подпись. Подпись к рисунку должна располагаться под рисунком и включать слово "Рисунок", его номер и название, разделенные символом тире, например, Рисунок 1.2 – Структура программной системы.

Каждая таблица должна иметь заголовок. Заголовок таблицы должен располагаться над таблицей и включать слово "Таблица", ее номер и название, разделенные символом тире, например, Таблица 2.2 – Результаты измерений.

Перечисления, при необходимости, могут быть приведены внутри пунктов или подпунктов. Перед перечислением ставят двоеточие. Перед каждой позицией перечисления следует ставить строчную букву русского алфавита со скобкой, или, не нумеруя, дефис (первый уровень детализации). Для дальнейшей детализации перечисления необходимо использовать арабские цифры со скобкой (второй уровень детализации). Перечисления первого уровня детализации печатают строчными буквами с абзацного отступа, второго уровня – с отступом относительно месторасположения перечислений первого уровня.

Все формулы и уравнения в ПЗ должны быть пронумерованы сквозной или порядковой нумерацией в пределах раздела. Формула и уравнение должны располагаться в отдельной строке и выравниваться по центру. Номер формулы или уравнения состоит из номера раздела и

порядкового номера формулы или уравнения, разделенных точкой и заключенных в круглые скобки, например, (2.3). Номер формулы или уравнения указывают на уровне формулы или уравнения в крайнем правом положении в строке.

Ссылки на используемые литературные источники следует указывать порядковым номером в библиографическом списке, заключенным в квадратные скобки, например, [1] или [2, 4 – 8] и т.д.

При ссылках на разделы, подразделы, пункты, подпункты, иллюстрации, таблицы, формулы, уравнения, приложения указывают их номера. Например, ". в разделе 4...", "... в соответствии с 2.1...", "... на рисунке 1.3...", "... в таблице 3.2...", "... по формуле (2).", "... в приложении Б ..." и т.д.

А.4. Оформление приложений

В приложениях рекомендуется размещать материал, не имеющий прямого отношения к теме. Приложения следует оформлять как продолжение отчёта на его последующих страницах, располагая приложения в порядке появления ссылок на них в тексте ПЗ. Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц. Каждое приложение должно начинаться с новой странницы и иметь обозначение. Обозначение включает слово "ПРИЛОЖЕНИЕ", прописную букву, обозначающую приложение и название (заголовок). Заголовок выполняется строчными буквами с первой прописной во второй строке страницы.

Приложения следует обозначать последовательно прописными буквами русского алфавита, начиная с А, за исключением Ё, З, Й, О, Ч, Ь, Ы, Ъ. Если в ПЗ имеется только одно приложение, то оно обозначается как приложение А.

При необходимости текст приложения может быть разделён на разделы, подразделы, пункты и подпункты, которые следует нумеровать в пределах каждого приложения в соответствии с ранее изложенными требованиями. При этом перед каждым номером проставляют обозначение приложения (букву) и точку, например, А.2.

Имеющиеся в тексте приложения иллюстрации, таблицы, формулы и уравнения следует нумеровать в пределах каждого приложения, например, рисунок Б.3 или таблица А.2; формула (А.4) и т.д.

Источники, цитируемые только в приложениях, должны рассматриваться независимо от цитируемых в основной части ПЗ, и должны быть перечислены в конце каждого приложения в перечне ссылок.

А.5. Оформление библиографического списка

Библиографический список должен быть оформлен в соответствии с требованием национального стандарта ГОСТ Р 7.0.100-2018 «Библиографическая запись. Библиографическое описание» (введен в действие с 1 июля 2019 года). Предшествующий ему межгосударственный стандарт ГОСТ 7.1-2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления» в настоящее время отменён в России, но остаётся действующим на территории СНГ.

В качестве заголовка раздела применяется название "Библиографический список". Библиографический список нумеруется и составляется, как правило, в порядке ссылок на источники по тексту документа.

В библиографическом описании источников допускаются общепринятые сокращения слов и места издания: Москва – М., Ленинград – Л.; Санкт-Петербург – С-Пб.; Киев – К.

Для книг, брошюр и подобных изданий общее число страниц указывается строчной буквой "с, " а для статей и докладов с прописной буквой "С" указываются первая и последняя страница данной работы в сборнике, например: " 235с. ; С.87-93."

Библиографическое описание содержит следующие элементы:

– сведения об авторе (авторах);

– заглавие, причем параллельное заглавие (например, на иностранном языке) отделяется от основного знаком « = », а дополнительные сведения (например, название серии) приводятся после двоеточия;

– сведения об ответственности отделяются от заглавия знаком « / » и приводятся, если авторов более трех или имеется ответственное лицо (составитель, редактор, переводчик и т.п.);

– групповой источник отделяется знаком « // » и приводится, если описываемый источник является составной частью (например, статьей из журнала);

– выходные данные отделяются знаком « .– » и содержат сведения о месте издания, издательстве, годе издания и числе страниц.

Подробное изложение правил можно найти по ссылке https://internet-law.ru/gosts/gost/70535/

Ниже представлены примеры библиографических описаний.

Книги одного автора:

Павловская, Т.А. С/С++. Программирование на языке высокого уровня/ Т.А. Павловская. — СПб.:Питер, 2002.—464с.— Текст: непосредственный.

Книги нескольких авторов:

Касьянов, В. Н. Графы в программировании: обработка, визуализация и применение/ В. Н.Касьянов , В. А. Евстигнеев. —СПб.:БХВ-Петербург, 2003.—1004с. – Текст: непосредственный.

Шапцев, В. А. Теория информации. Теоретические основы создания информационного общества : учебное пособие / В. А. Шапцов, Ю. В. Бидуля. — Москва : Юрайт, 2019. — 177 с. — Текст: непосредственный (Университеты России). — ISBN 978-5-534-02989-5. — Текст: непосредственный.

Словари:

Варламова, Л. Н. Управление документацией: англо-русский аннотированный словарь / Л. Н. Варламова, Л. С. Баюн, К. А. Бастрикова.— Москва : Спутник+, 2017. — 398 с. - ISBN 978-5-9973-4489-4. — Текст: непосредственный.

Книги из ЭБС:

Павлов, Л. А. Структуры и алгоритмы обработки данных / Л. А. Павлов, Н. В. Первова. — 2-е изд., стер.—Санкт-Петербург: Лань, 2022. — 256 с. — URL: https://e.lanbook.com/book/207563 (дата обращения: 10.01.2025). — Режим доступа: для авторизованных пользователей. — ISBN 978-5-507-44105-1. — Текст: электронный.

Материалы конференции:

Проблемы развития предприятий: теория и практика: материалы 16-й Международной научно-практической конференции, Самара, 16-17 ноября 2017 г.: в 3 ч. Ч. 2. Региональное развитие в условиях глобализации. Развитие теории и практики менеджмента предприятий в условиях перехода к инновационной экономике / отв. ред. С. И. Ашмарина. — Самара : Изд-во Самар. гос. экон. ун-та, 2017— 276 с. — ISBN 978-5-94622-775-9. — Текст: непосредственный.

Электронные ресурсы:

Правительство Российской Федерации: официальный сайт. — Москва. — Обновляется в течение суток. — URL: http://government.ru (дата обращения: 10.05.2025). — Текст: электронный.

Национальная стратегия развития ИИ на период до 2030 года [Электронный ресурс]. — URL: http://www.kremlin.ru/acts/bank/44731 (дата обращения 03.05.2025) — Режим доступа: свободный. —Текст: электронный.

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Методические указания
к выполнению курсовой работы

Составитель: **Шалимова** Елена Михайловна

В авторской редакции

Изд. № 104/2025. Объем 1,75 п.л. Усл. печ. л. 1,63. Уч.-изд. л. 1,715.
Издательский центр ФГАОУ ВО «Севастопольский государственный университет»