

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
Московский государственный университет имени М.В. Ломоносова
филиал МГУ в г. Севастополе
факультет компьютерной математики
кафедра программирования

УТВЕРЖДЕНО
на 20 22 - 20 23 учебный год
Методическим советом Филиала

Протокол № 8 от 28.06.2022 г.

Заместитель директора по учебной работе

Заведующий кафедрой

УТВЕРЖДАЮ

Директор
Филиала МГУ в г. Севастополе
О.А. Шпырко

«03» августа 20 21 г.

УТВЕРЖДЕНО
на 20 23 - 20 24 учебный год
Методическим советом Филиала

Протокол № 9 от 28.06.2023 г.

Заместитель директора по учебной работе

Заведующий кафедрой

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
Наименование дисциплины (модуля):

Б-ОН АРХИТЕКТУРА ЭВМ И ЯЗЫК АССЕМБЛЕРА

код и наименование дисциплины (модуля)

Уровень высшего образования:
бакалавриат

Направление подготовки:

01.03.02 Прикладная математика и информатика

(код и название направления/специальности)

Направленность (профиль) ОПОП:
общий

(если дисциплина (модуль) относится к вариативной части программы)

Форма обучения

очная

Рабочая программа рассмотрена
на заседании кафедры программирования
протокол № 1 от «10» июня 2021 г.
Руководитель ОП 01.03.02 «Прикладная
математика и информатика»

(подпись) (Н. В. Лактионова)

Рабочая программа одобрена
Методическим советом
Филиала МГУ в г. Севастополе
Протокол № 8 от «31» августа 2021 г.

(подпись) (С. А. Наличаева)

Севастополь, 2021

Рабочая учебная программа составлена на основе

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.03.02 «Прикладная математика и информатика» в редакции приказа МГУ от 30 августа 2019 г. (в редакции приказа МГУ от 11 сентября 2019г №1109)

курс – I

семестры – 2

зачетных единиц – 3

академических часов – 108, в т.ч.:

лекций – 54 часа

практических занятий – нет

Формы промежуточной аттестации:

зачёты в семестрах – нет

Форма итоговой аттестации:

экзамены во 2 семестре.

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплин модуля

Целью освоения дисциплины «Архитектура ЭВМ и язык ассемблера» является:

- создание у студентов на основе современного состояния компьютерной индустрии целостного представления об этапах развития вычислительной техники, архитектуре ЭВМ и способах прохождения задач через вычислительные системы.

Основные задачи дисциплины:

- дать основные сведения об архитектуре и принципах работы ЭВМ;
- на примере модельных ЭВМ разобрать достоинства и недостатки различных систем команд;
- изучить конкретный язык ассемблера. Рассмотреть методы реализации на этом языке управляющих конструкций и различных структур данных из языков программирования высокого уровня;
- дать понятие о модульном программировании и системе программирования, схеме работы ассемблера, редактора внешних связей, загрузчика.
- рассмотреть особенности современных компьютеров, дать представление об аппаратных способах повышения их быстродействия.

2. Место модуля дисциплин в структуре ООП

Дисциплина «Архитектура ЭВМ и язык ассемблера» входит в базовую часть профессионального цикла ОС МГУ по направлению подготовки 010400.62 «Прикладная математика и информатика». Логически и содержательно-методически данная дисциплина связана базовыми курсами: «Алгоритмы и алгоритмические языки» и «Операционные системы».

Модуль дисциплин «Информатика» поддерживается дисциплиной «Практикум на ЭВМ» входящей в базовую часть профессионального цикла ОС МГУ по направлению подготовки 010400.62 «Прикладная математика и информатика». «Практикум на ЭВМ» включает в себя как семинарские занятия, так и практическое выполнение заданий на ЭВМ.

Для успешного освоения дисциплин модуля «Информатика» студент должен обладать основами знаний по информатике и математике в рамках школьной программы.

3. Требования к результатам освоения дисциплин модуля

В результате освоения дисциплин модуля «Информатика» обучающийся должен:

Знать:

«Архитектура ЭВМ и язык Ассемблер»

- основные архитектурные особенности ЭВМ;
- способы работы ЭВМ;
- синтаксис и семантику языка низкого уровня (Ассемблера);
- способы разработки программного обеспечения на Ассемблере;

– способы связи языков высокого уровня и языка Ассемблера.

Уметь:

«Архитектура ЭВМ и язык Ассемблер»

- разрабатывать алгоритмы для исполнителя низкого уровня – компьютера;
- разрабатывать и реализовывать простые алгоритмы на Ассемблере;

Владеть:

«Архитектура ЭВМ и язык Ассемблер»

- профессиональными знаниями из области архитектуры ЭВМ и способов выполнения программ на компьютере.

1. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИН МОДУЛЯ

а) Общая трудоемкость дисциплин

«Архитектура ЭВМ и язык ассемблера»

- 34 зачетных единицы,
- 144 академических часа. В том числе: 54 часа лекций, 90 часов самостоятельной работы

б) Тематический план

Таблица 1.

ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Название темы	Количество часов			Формы текущего контроля успеваемости (по темам) / Форма промежуточной аттестации (по семестрам)
		Л	С(П,Лб)	СРС	
1	2	3	4	5	6
Семестр II «Архитектура ЭВМ и язык ассемблера»					
Раздел 1. Архитектура ЭВМ					
1	Понятие об архитектуре ЭВМ	2		3	Письменная самостоятельная работа (одна задача или пример)
2	Машина фон Неймана	2		3	Письменная самостоятельная работа (одна задача или пример)
3	Учебные ЭВМ	4		6	Письменная самостоятельная работа (одна задача или пример)

4	Форматы команд и данных	2		5	Письменная самостоятельная работа (одна задача или пример)
---	-------------------------	---	--	---	--

1	2	3	4	5	6
5	Архитектура младшей модели семейства Intel	4		8	Письменная самостоятельная работа (одна задача или пример)

Раздел 2. Язык ассемблера

6	Особенности архитектуры ПК	2		3	Письменная самостоятельная работа (одна задача или пример)
7	Основные конструкции языка MASM	2		3	Письменная самостоятельная работа (одна задача или пример)
8	Команды перехода	2		3	Письменная самостоятельная работа (одна задача или пример)
9	Модификация адресов в ПК	2		3	Письменная самостоятельная работа (одна задача или пример)
10	Битовые команды	2		3	Письменная самостоятельная работа (одна задача или пример)
11	Сегментирование памяти в ПК	2		3	Письменная самостоятельная работа (одна задача или пример)
12	Стек	2		3	Письменная самостоятельная работа (одна задача или пример)
13	Процедуры	2		4	Письменная самостоятельная работа (одна задача или пример)
14	Макросредства	4		8	Письменная самостоятельная работа (одна задача или пример)
15	Многомодульные программы	2		4	Письменная самостоятельная работа (одна задача или пример)

Раздел 3. Элементы систем программирования

16	Понятие о системе программирования	2		3	Письменная самостоятельная работа (одна задача или пример)
17	Задачи и принцип работы двухпроходного ассемблера	2		3	Письменная самостоятельная работа (одна задача или пример)
18	Задачи и схемы работы редактора внешних связей	3		4	Письменная самостоятельная работа (одна задача или пример)

19	Статический и динамический за- грузчики и схема их работы	3		4	Письменная самостоя- тельная работа (одна задача или пример)
----	--	---	--	---	--

1	2	3	4	5	6
Раздел 4. Особенности архитектуры современных компьютеров					
20	Понятие прерываний	2		3	Письменная самостоятельная работа (одна задача или пример)
21	Мультипрограммный режим работы ЭВМ	2		3	Письменная самостоятельная работа (одна задача или пример)
22	Архитектура с общей шиной и каналами ввода-вывода	4		8	Письменная самостоятельная работа (одна задача или пример)
Всего, часов		54		90	
Итоговая аттестация					Экзамен - 36 часов

где: Л – лекции, С – семинарские занятия, П – практические занятия, Лб – лабораторные занятия, СРС – самостоятельная работа студентов.

2. СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИН МОДУЛЯ

Планы лекций

«Архитектура ЭВМ и язык ассемблера»

Раздел 1. Архитектура ЭВМ

Тема 1. Понятие об архитектуре ЭВМ. Уровни видения архитектуры. Рассмотрение архитектуры на внешнем, концептуальном и внутренних уровнях.

Тема 2. Машина фон Неймана. Типовая схема ЭВМ: центральный процессор, оперативная память, внешние устройства. Принципы фон Неймана. Организация оперативной памяти: ячейка, адрес ячейки, машинное слово, объем оперативной памяти.

Тема 3. Учебные ЭВМ. Трех-, двух-, одно- и нульадресные ЭВМ.

Тема 4. Способы адресации. Многообразие форматов команд и данных. Принципы базирования и сегментирования памяти.

Тема 5. Архитектура младшей модели семейства Intel Способы представления данных в памяти ЭВМ: целые числа без знака и со знаком; использование дополнительного кода для представления чисел со знаком. Сложение и вычитание целых чисел. Особенности представления вещественных чисел. Устройство процессора: машинная команда, такт работы процессора. Понятие системы команд процессора; выполнение процессором арифметических команд и команд перехода. Регистры, преимущества использования регистров. Механизм модификации адресов, понятие исполнительного адреса. Базирование.

Раздел 2. Язык ассемблера

Тема 1. Особенности архитектуры персонального компьютера: структура оперативной памяти, регистры, арифметические флаги.

Тема 2. Основные конструкции языка MASM: лексемы: идентификаторы, числа строки. Типы предложений языка MASM: комментарии, директивы, команды. Директивы определения данных. Константные и адресные выражения. Правила записи команд на языке MASM.

Тема 3. Команды перехода: действие команд перехода. Виды переходов: внутрисегментные, межсегментные, прямые и косвенные. Условные переходы. Команда LOOP.

Тема 4. Модификация адресов в ПК: массивы, их описание, доступ к элементам. Правила записи выражений. Операторы TYPE, LENGTH, SIZE, OFFSET.

Тема 5. Битовые команды: логические команды, сдвиги. Быстрое умножение и деление на степени двойки. Упакованные данные и операции над ними.

Тема 6. Сегментирование памяти в ПК: абсолютные адреса. Сегментные регистры. Выбор сегментных регистров при вычислении абсолютного адреса. Директива ASSUME.

Тема 7. Стек: описание сегмента стека. Загрузка регистров SS и SP. Команды PUSH и POP. Работа со стеком через регистр BP.

Тема 8. Процедуры: описание процедур, близкие и дальние процедуры. Команды вызова процедуры и возврата из нее. Передача параметров через регистры и через стек. Стандартные соглашения о связях.

Тема 9. Макросредства: различия между директивами EQU и =. Условное ассемблирование. Блоки повторения. Макросы, их описание и использование. Директива LOCAL. Отличие макросов от процедур.

Тема 10. Многомодульные программы: понятие модуля, независимая трансляция модулей. Описание модуля на языке MASM. Межмодульные связи: внешние и общие имена.

Раздел 3. Элементы системы программирования

Тема 1. Понятие о системе программирования

Тема 2. Задачи и принципы работы двухпроходного ассемблера: таблицы ассемблера. Структура объектного модуля.

Тема 3. Задачи и схема работы редактора внешних связей: структура загрузочного модуля.

Тема 4. Статический загрузчик и схема его работы. Понятие о динамической загрузке модулей, схема работы динамического загрузчика. Явное и неявное связывание.

Раздел 4. Особенности архитектуры современных компьютеров

Тема 1. Понятие прерываний: внутренние и внешние прерывания. Аппаратная и программная реакция на прерывание. Организация ввода-вывода.

Тема 2. Мультипрограммный режим работы ЭВМ: аппаратные средства поддержки мультипрограммного режима: прерывания, защита памяти, привилегированные команды, таймер.

Тема 3. Архитектура компьютеров с общей шиной. Порты ввода-вывода. Протокол работы шины. Понятие блокирующего ввода-вывода. Архитектура компьютеров с каналами ввода-вывода. Параллельная работа каналов и центрального процессора. Не блокирующий ввод-вывод.

3. РЕКОМЕНДУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Работа в аудитории: лекции; консультации перед экзаменом.

Внеаудиторная работа: изучение пройденных на лекциях тем, самостоятельное изучение литературы по алгоритмам и алгоритмическим языкам и по архитектурам ЭВМ и языкам программирования низкого уровня (Ассемблерам).

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

а) Оценочные средства для текущего контроля успеваемости

– небольшие самостоятельные работы (одна задача или пример) по каждой пройденной на лекциях теме.

Самостоятельная работа по изучению данной дисциплины включает:

- проработку теоретических основ лекционного материала;
- систематизацию изученного материала по курсу;
- научно-исследовательская работа учащегося в библиотеках;
- подготовка к устному экзамену.

б) Система итогового контроля знаний

По итогам освоения дисциплин модуля проводится **письменные экзамены.**

«Архитектура ЭВМ и язык ассемблера»

Фамилия И.О.

<u>1</u>	2	3	4	5	6	7

1. Перечислить аппаратные действия, выполняемые центральным процессором ЭВМ при поступлении сигнала прерывания.

Ответ:

2. Что будет напечатано в результате выполнения последовательности команд

```
mov     cx, -254
mov     ax, cx
imul   ah
shl     ax, cl
outint  ax
```

Ответ:

3. Написать полную программу на Ассемблере, которая вводит одно целое знаковое число (по макрокоманде `inint`) и, если это число чётное, то выводит частное от деления введённого числа на константу 13, иначе выводит значение ноль. (Ответ на обороте.)

4. Вычеркнуть синтаксически неверные команды

```
MUL AL    CMP CF,1    ADD 5000[SI],0    JMP AX    MOV BX,'A'    MOV AX[BX],5
```


ют, а в других случаях дает пустое макрорасширение. Минимум считать самим.

7. Вычеркните команды, записанные с ошибкой:

а) `div 5` б) `lea BX, [BX]` в) `mov ES, DS` г) `add 800[SI], 15` д) `jmp AX`

8. Передавая параметры через стек, описать процедуру `AVG(X, N)`, где `X` – массив беззнаковых байт, `N` – количество элементов в массиве ($N < 256$). Процедура вычисляет среднее арифметическое элементов массива. Результат вернуть в `AL`, регистры не портить. Предусмотреть возможное переполнение.

9. Записать значение `-13` в двойное слово по абсолютному адресу `512133`.

10. Написать фрагмент программы с использованием строковых команд для решения следующей задачи.

`S DB 500 DUP (?)`; массив литер

В массиве `S` задана последовательность из 500 литер. Литера `' , '` - запятая - разделяет последовательность на слова. Записать в `AX` количество слов в последовательности. Учесть, что слов нулевой длины быть не может.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Любимский Э.З., Мартынюк В.В., Трифонов Н.П. Программирование. – М., Наука, 1980.
2. Пильщиков В.Н. Программирование на языке Ассемблера IBM PC. – М., Диалог-МИФИ, 1994.
3. Баула В.Г. Введение в архитектуру ЭВМ и системы программирования. – М., Изд. Отдел ф-та ВМиК МГУ, 2007.
4. Баула В.Г., Томилин А.Н., Волканов Д.Ю. Архитектура ЭВМ и операционные среды. – М., Академия, 2012.
5. Таненбаум Э. Архитектура компьютера. – СПб., Питер, 2005.

б) дополнительная литература

1. Королев Л. Н. Архитектура ЭВМ. М., – Научный мир, 2005.
2. Хамахер К., Враншевич З., Заки С. Организация ЭВМ. СПб., Питер, 2003.
3. Корнеев В.В., Киселев А.В. Современные микропроцессоры. – СПб., БХВ-Петербург, 2003.
4. Ю-Чжень Лю, Гибсон Г. Микропроцессоры семейства 8086/8088. М., – Радио и связь, 1987.
5. Юров В.И. Assembler. Специальный справочник. – СПб., Питер, 2005.
6. Бордаченкова Е.А., Метвеева Т.К.. Архитектура ЭВМ и язык ассемблера. Часть 1. – М., Изд. Отдел ф-та ВМиК МГУ, 2003.

в) Интернет-ресурсы

- видеозаписи лекций доцента факультета ВМК Баулы В.Г.

<http://video.yandex.ua/#search?text=%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B8%20%D0%91%D0%B0%D1%83%D0%BB%D1%8B>

- электронная учебно-методическая система «Ownlibragi» кафедры программирования Филиала МГУ в г. Севастополе

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

- библиотека Филиала МГУ в г. Севастополе;
- библиотека кафедры программирования Филиала МГУ в г. Севастополе;
- лекционные аудитории;
- специализированные компьютерные классы с доступом к Интернет-ресурсам с любого компьютера.

ПРИЛОЖЕНИЕ А

ПОЛОЖЕНИЕ
о проведении пересдач задолженностей студентов по итоговым аттестациям
на кафедре программирования Филиала МГУ в г. Севастополе
прикомандированными преподавателями с факультета ВМК МГУ
протокол №5 заседания кафедры от 10 апреля 2012 г.

Настоящее положение регулирует порядок проведения пересдач задолженностей студентами факультета Компьютерной математики на кафедре программирования Филиала МГУ в г. Севастополе прикомандированными преподавателями с факультета ВМК МГУ.

1. Пересдача задолженностей принимается по графику пересдач задолженностей, установленному кафедрой программирования в принятые ректоратом МГУ имени М.В.Ломоносова сроки, в отдельных случаях - в сроки установленные комиссией по студенческим делам.
2. График пересдач задолженностей сообщается в учебный отдел в виде служебной записки и преподавателю факультета ВМК МГУ, по дисциплине которого проводится пересдача.
3. Пересдачи экзаменов (зачётов) в отсутствие преподавателей факультета ВМК в Филиале МГУ в г. Севастополе проводятся в письменной форме, независимо от того в какой форме проводился основной экзамен (зачёт).
4. Вариант письменной экзаменационной (зачётной) работы для пересдачи составляет преподаватель, проводивший основной экзамен. Также преподаватель указывает требования к проведению экзамена (сколько времени даётся на написание работы, какими материалами разрешается пользоваться студенту при написании работы и др.)
5. Кафедрой программирования назначаются местные преподаватели кафедры, которые проводят пересдачу в установленные даты и время в соответствии с графиком пересдач по присланному варианту с выполнением всех требований к проведению.
6. Написанные студентами при пересдаче экзаменационные (зачётные) работы сканируются по окончании пересдачи и пересылаются по электронной почте на проверку преподавателю факультета ВМК МГУ.
7. Преподавателю ВМК МГУ пересылается скан ведомости пересдачи для выставления отметок по результатам пересдачи.
8. Преподаватель ВМК присылает в Филиал скан заполненной им ведомости пересдачи и сканы проверенных им работ.
9. Полученный скан ведомости подписывается зам. зав. кафедрой.
10. Скан ведомости пересдачи сдаётся в учебный отдел Филиала.
11. Кафедра знакомит студентов, писавших работу на пересдаче, с результатами пересдачи.
12. При возникновении у студентов, писавших работу на пересдаче, вопросов по результатам проверки, студент может обратиться к преподавателю ВМК лично по электронной почте (скайпу). Свои координаты для консультаций с ним преподаватель сообщает студентам на первой лекции курса.