

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики



УТВЕРЖДАЮ

Декан факультета ВМК МГУ

И.А.Соколов/

2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Наименование дисциплины (модуля):

Архитектура ЭВМ и язык ассемблера

Уровень высшего образования:

бакалавриат

Направление подготовки (специальность):

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) ОПОП:

дисциплина относится к базовой части программы

Форма обучения:

очная

Москва 2023

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ бакалавриата, магистратуры, реализуемых последовательно по схеме интегрированной подготовки по направлениям 02.03.02, 02.04.02 «Фундаментальная информатика и информационные технологии» в редакции приказа МГУ от 30 декабря 2016 г.

1. Дисциплина относится к базовой части ОПОП ВО.

2. Входные требования для освоения дисциплины (модуля): отсутствуют.

3. Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.

Компетенции выпускников, частично формируемые при реализации дисциплины (модуля):

- **ОПК-2.** Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий;
- **ОПК-3.** Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям;
- **ПК-3.** Способность использовать современные инструментальные и вычислительные средства;
- **ПК-8.** Способность применять на практике международные и профессиональные стандарты информационных технологий, современные парадигмы и методологии, инструментальные и вычислительные средства.

Планируемые результаты обучения по дисциплине (модулю):

Знать:

1. основные принципы устройства ЭВМ, основные компоненты, их качественные и количественные характеристики;
2. особенности архитектуры IA-32, влияние архитектуры набора команд на реализацию языков высокого уровня и систему программирования;
3. язык ассемблера архитектуры IA-32;
4. особенности реализации различных конструкций языка Си на языке ассемблера и причины, влияющие на выбор механизма реализации;
5. свойства аппаратуры, влияющие на производительность и безопасность работы программ.

Уметь:

1. составлять и отлаживать многомодульные программы на языках Си и ассемблера;
2. использовать в программах на языке ассемблера стандартную библиотеку языка Си;
3. обосновано выбирать механизмы реализации конструкций языка Си и корректно их кодировать на языке ассемблера;
4. оценивать производительность программ на языках Си и ассемблера, используя знания об особенностях устройства аппаратуры ЭВМ.

Владеть:

1. современной технологией разработки и отладки многомодульных программ на языках Си и ассемблера.

4. Формат обучения: лекции проводятся с использованием компьютера и мультимедийного проектора для демонстрации слайдов. Для самостоятельной работы студенты используют компьютеры в машинных залах и/или личные компьютеры.
5. Объем дисциплины (модуля) составляет 3 з.е., в том числе 54 академических часов, отведенных на контактную работу обучающихся с преподавателем, 54 академических часов на самостоятельную работу обучающихся.
6. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий.

Наименование и краткое содержание разделов и тем дисциплины (модуля), Форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе			
		Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, часы		Самостоятельная работа обучающегося, часы (виды самостоятельной работы – эссе, реферат, контрольная работа и пр.)	
		Занятия лекционного типа*	Занятия семинарского типа*	Всего	
1. Введение в архитектуру ЭВМ. Основные компоненты компьютера.	5	4	0	4	1
2. Организация ассемблерной программы	5	4	0	4	1
3. Арифметические и битовые операции над целыми числами	8	6	0	6	2
4. Текущий контроль успеваемости: контрольная работа № 1	2	0	0	0	2
5. Условная и безусловная передача управления. Реализация ветвлений и циклов	5	4	0	4	1
6. Массивы. Адресная арифметика.	5	4	0	4	1
7. Текущий контроль успеваемости: коллоквиум №1	2	0	0	0	2
8. Организация вызова функции. Соглашения о вызовах. Организация рекурсии. Работа со стандартными функциями языка Си	10	8	0	8	2
9. Текущий контроль успеваемости: контрольная работа № 2	2	0	0	0	2
10. Представление вещественных чисел. Стандарт E-754.	5	4	0	4	1

Арифметический сопроцессор FPU x87. Программирование арифметических выражений и условий.									
11. Макросредства языка ассемблера namn. Условная компиляция. Сравнение функций и макросов.	5	4	0	4	0	1			
12. Текущий контроль успеваемости: контрольная работа № 3	2	0	0	0	0	2			
13. Многомодульная программа. Связывание имен. Сборка многомодульной программы.	5	4	0	4	0	1			
14. Система программирования языка Си и ее связь с архитектурой ЭВМ	5	4	0	4	0	1			
15. Особенности архитектуры современных ЦПУ. Аппаратная поддержка многозадачной работы компьютера.	5	4	0	4	0	1			
16. Архитектурные особенности современных компьютеров. Устройство оперативной памяти. Кэширование. Конвейерная обработка данных.	5	4	0	4	0	1			
17. Текущий контроль успеваемости: коллоквиум №2	2	0	0	0	0	2			
Промежуточная аттестация: письменный экзамен	30	0	0	0	0	30			
Итого	108	54	0	54	0	54			

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю)

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

Типовая контрольная работа № 1

1. Рассматривается представление чисел в формате байта. Заполните все пустые ячейки таблицы:

Двоичное представление (байт)	16-ричное представление (байт)	Число десятичное со знаком	Число десятичное без знака
11010111		-95	
	50h		
			100

2. Выпишите результат операций в десятичном виде и значения флагов. Рассматриваются байты.

операция	без знака	со знаком	SF	ZF	CF	OF
200 + (-70)						
30 - 150						

3. Укажите значение регистра `eax` (как десятичное число со знаком) после выполнения следующего фрагмента кода:

```
mov cx, -253
mov ax, cx
imul ah
shl ax, cl
movsx eax, ax
```

3. Зачеркните команды, содержащие ошибки.

```
section .bss
x resd 1
y resd 1
add byte [x], edx      movzx eax, ah      div dword [x]
test 15, eax          mov dword[x], dword[y]  cmp ax, 0
```

4. Напишите код на ассемблере для реализации указанного присваивания.

```
section .bss
a resw 1; число без знака
b resd 1; число со знаком
b := (b + a) mod 350 - a * 10.
```

Типовая контрольная работа № 2

```
1. section .bss
x resd 100
```

Напишите команды (не больше двух) для реализации следующего действия:

```
x[2] := x[99]
```

```
2. section .bss
x resb 100
```

Напишите фрагмент кода для заполнения всех элементов массива `x` по следующему правилу :

```
x[i] := 100 + i
```

3. Зачеркните команды, содержащие ошибки.

```
section .bss
x resd 1
movsx ebx, byte [eax]      pop x      sub eax, dword [x + ecx * 10]
shr word [x], 15          imul 10      cmp dword [eax + ebx * 4], 2
```

4. Напишите на ассемблере функцию, прототип которой на языке Си имеет вид:

```
void aver_max(int x, int y, int *z);
```

Функция находит среднее арифметическое модулей чисел x и y и записывает его по указателю z .

Функция должна удовлетворять соглашениям cdecl.

Напишите пример вызова функции aver_max из главной программы.

5. Напишите код на ассемблере для решения указанной задачи. Пусть в регистре eax содержится целое неотрицательное число. Поместить в регистр esx 1, если в десятичной записи этого числа содержится четное количество цифр 3, и 0 в противном случае.

6. Укажите основные отличия соглашения о вызовахfastcall от соглашения cdecl.

Типовая контрольная работа № 3 (проводится на компьютерах в машинном зале в системе автоматизированного приема задач e-judge)

1. На стандартном потоке ввода задается натуральное число $2 \leq N < 10^6$, после которого следует последовательность из N 32-битных целых чисел.
На стандартный поток вывода напечатать 1, если эта последовательность является строго монотонной, и 0 в противном случае.
2. На стандартном потоке ввода задается положительное 32-битное число. На стандартный поток вывода напечатать максимальное возможное число, полученное из десятичных цифр исходного числа.
Примеры: 53095 → 95530, 10317 → 73110, 78554 → 78554.
3. На стандартном потоке ввода задается непустая последовательность из положительных 32-битных чисел, оканчивающаяся нулем. На стандартный поток вывода напечатать максимальное количество делителей, имеющих у чисел данной последовательности. При решении задачи необходимо описать функцию $F(x)$ для подсчета количества всех делителей числа x (1 и само число x тоже входят в число делителей). Функция должна удовлетворять соглашениям о вызовах cdecl.

Типовой вариант коллоквиума

1. Определить значение регистра AL в виде знакового десятичного числа, а также значения флагов CF, OF, SF, ZF после выполнения следующих команд:
MOV AL, 120
ADD AL, 6Eh
2. Дайте определение следующим понятиям:
a) ячейка памяти
b) машинная команда
c) регистр команд.
3. Как в компьютере определяется, какое число – со знаком или без знака - находится в ячейке?

4. Пусть размер машинной ячейки 9 двоичных разрядов и для представления чисел со знаком используется дополнительный код. Указать в десятичной системе счисления наибольшее и наименьшее числа со знаком и числа без знака, представимые в такой ячейке. Указать в 16-ричном виде представление числа (-71) в такой ячейке.

	Наименьшее число	Наибольшее число
Число со знаком		
Число без знака		
Представление числа (-71)		

5. section .bss
 X resb 100 ; числа со знаком
 Y resw 100
 Считая массив X уже введенным, написать фрагмент кода для реализации присваивания $Y[i] := X[i]$ для всех $i = 0, \dots, 99$.

6. Из указанных команд вычеркнуть те, что записаны с ошибкой:

```

CMP CF, 1      ADD AX, BYTE [EBP + 8] MUL 5
MOV AL, 'C'   DEC DWORD [ESP]      LEA ECX, [EBX-a']

```

7. Укажите, на какую метку будет осуществлен переход при выполнении следующего фрагмента команд. Ответ обосновать:

```

mov al, -80
cmp al, 150
jb .m1
jl .m2
jmp .m3

```

8. Напишите код на ассемблере для реализации указанного присваивания.

```

section .bss
a resw 1; число без знака
b resd 1; число со знаком
b := (b + a) mod 350 - a * 10.

```

9. Укажите значение регистра eax (как десятичное число со знаком) после выполнения следующего фрагмента кода:

```

mov cx, -253
mov ax, cx
imul ah
shl ax, cl
movsx eax, ax

```


7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

Типовая зачетная работа

1. Зачеркните команды, содержащие ошибки:

```
shr edx, al      cmp CF, 0      neg dword[edx + esi * 4]
jmp edx         imul ax, cx      xor byte[edi], byte[edi]
```

2. Указать значение регистра CL (в виде десятичного числа со знаком и без знака) и флагов CF, OF, SF, ZF после выполнения команд:

```
mov cl, 0xD1
add cl, -74
```

3. Напишите на ассемблере код для реализации указанного присваивания.

```
section .bss
a resb 1; число без знака
b resd 1; число со знаком
c resd 1
c : = ((b - a) div 113) * ((b - a) mod 113).
```

4. section .data

```
a dd 0xAAFE0203
```

Укажите значение регистра eax (как десятичное число со знаком) после выполнения следующего фрагмента кода:

```
mov cx, word [a + 1]
mov ax, cx
imul ah
shl ax, cl
movsx eax, ax
```

5. Дано описание массива:

```
section .bss
x resd 100; числа без знака
```

Считая, что массив уже введен в память, напишите фрагмент программы для решения следующей задачи: в регистр eax поместить количество элементов массива, двоичное представление которых содержит не менее 5 единиц.

6. Напишите на ассемблере функцию `zero_less(x,y)`, параметрами которой являются адреса двух знаковых переменных в формате двойного слова. Функция обнуляет переменную с меньшим значением. Функция должна удовлетворять соглашению `sdecl`. Напишите пример вызова функции.
7. Описать макрос `is_letter`, единственным параметром которого является имя байтовой переменной, где записан некоторый символ. Макрос записывает в регистр `eax` 1, если переменная содержит строчную латинскую букву, 0 – в противном случае. Выпишите макрорасширение для макровызова `is_letter A`

Вопросы к экзамену.

1. Архитектурные принципы фон Неймана.
2. Ключевые термины архитектуры компьютера: память, ячейка памяти, ЦП, регистр.
3. Такт работы ЦП.
4. Понятие семейства ЭВМ, основные требования.
5. Представление целых чисел в IA-32, особенности хранения чисел в основной и регистровой памяти, прямой и дополнительный код.
6. Организация ассемблерной программы, секции кода и данных. Директивы определения данных.
7. Команды пересылки данных и арифметические команды. Регистр флагов.
8. Арифметические операции над 64-разрядными числами.
9. Команды передачи управления. Косвенные и прямые переходы. Команды безусловного и условных переходов.
10. Реализация ветвлений и циклов.
11. Побитовые операции, сдвиги, быстрое умножение и деление на степень двойки.
12. Одномерные массивы: размещение в памяти, доступ к элементам массива, обработка массивов.
13. Двумерные массивы: размещение в памяти, доступ к элементам массива, обработка массивов.
14. Поддержка аппаратного стека IA-32. Команды доступа к стеку (чтение/запись).
15. Функции. Команды вызова функции и возврата из нее. Передача параметров в функцию. Понятие стекового фрейма функции.
16. Соглашение о вызовах `sdecl`, основные требования соглашения.
17. Реализация рекурсивных функций.
18. Использование функций стандартной библиотеки в ассемблерном коде. Внешние имена. Выравнивание фреймов в стеке.
19. Другие соглашения о вызовах: `fastcall`, `stdcall`. Их отличия от соглашения `sdecl`.
20. Представление вещественных чисел в стандарте IEEE – 754. Нормализованные и ненормализованные числа, особые значения. Округление чисел.
21. Сопроцессор x87, структура сопроцессора.
22. Сопроцессор x87, обработка вещественных чисел. Основные команды: обмен с памятью, арифметические операции, сравнение. Дополнения к соглашению `sdecl`.
23. Макросредства языка ассемблера, основные понятия: макроопределение, макрокоманда, макроподстановка, макрорасширение.
24. Однострочные и многострочные макросы в `asm`. Локальные метки в макросах.

25. Условная компиляция. Разновидности директив условной компиляции.
26. Сравнение макросов и функций.
27. Понятие модульного программирования. Преимущества и недостатки.
28. Сборка многомодульной программы, основные этапы.
29. Поддержка модулей в `asm`. Глобальные метки и внешние имена. Понятие связывания.
30. Статический компоновщик: назначение, схема работы.
31. Понятие объектного модуля. Перемещаемый и исполняемый объектный код.
32. Понятие системы программирования. Основные программные инструменты в составе системы.
33. Загрузчик: назначение, виды загрузчиков. Отличия статического и динамического загрузчиков.
34. Понятие мультипрограммного режима работы. Пакетный режим, режим разделения времени, режим реального времени. Основные отличия.
35. Требования к аппаратуре компьютера для поддержки многопрограммного режима.
36. Понятие прерывания. Обработка прерываний: аппаратная и программная реакция на прерывание.
37. Системные вызовы: назначение, реализация в IA-32.
38. Иерархическая организация памяти, понятие кэша. Попадание и промахи в кэше. Кэш прямого отображения.
39. Понятие конвейера. Общие принципы конвейерной обработки.

Типовой вариант письменного экзамена.

1. Определите значение регистра EAX (в виде 16-ричного кода), а также значения флагов CF, OF, SF, ZF после выполнения следующих команд:

<pre>section .data a dw 0x2019, 0xbabe</pre>	<pre>section .text movsx eax, word [a + 1] add al, 0x80</pre>
--	---
2. Пусть в центральном процессоре используется конвейерный способ выполнения команд, при котором параллельно выполняются следующие 6 этапов: 1) чтение команды, 2) дешифровка кода операции, 3) вычисление исполнительных адресов, 4) чтение операндов, 5) выполнение операции, 6) запись результата. Считая, что на каждый из этих этапов тратится 1 единица времени, и что вначале конвейер был свободен, определите, сколько времени будет затрачено на выполнение следующего фрагмента программы:


```
sub eax, 3
add ebx, eax
and dword [x + ebx], eax
```
3. Перечислите этапы сборки программы.
4. Пусть размер машинной ячейки 11 двоичных разрядов и для представления чисел со знаком используется дополнительный код. Укажите в десятичной системе счисления наибольшее и наименьшее числа со знаком и числа без знака, представимые в такой ячейке. Укажите в 16-ричном виде представление числа (-14) в такой ячейке.

	Наименьшее число	Наибольшее число
Число со знаком		
Число без знака		
Представление числа (-14)		

5. section .bss
x resd 100
y resw 100; числа со знаком
Считая массив y уже введенным, напишите фрагмент кода для реализации присваивания $x[1] = y[97]$ (не более двух команд).
6. А) Для представления вещественных чисел используется 8-битный формат, удовлетворяющий требованиям стандарта IEEE 754: знаковый бит, 4 бита – порядок, 3 бита – мантисса. Округление выполняется к ближайшему четному. Выпишите битовое представление для чисел:

Наименьшее ненормализованное
4.75

- Б) Перепишите приведенный ниже код, не используя команды сопроцессора x87.
- ```
fld0
fstp qword [esp + 4]
```
7. Напишите на ассемблере функцию, определение которой на языке Си имеет вид:
- ```
int f (int *a, short b) {
    return (*a + 2019) % b;
};
```
- Функция должна удовлетворять соглашению cdecl.
8. Опишите присваивание $C = \max(A, B)$ в виде макроса `mmax A, B, C` при условии, что A, B, C обозначают имена знаковых переменных в формате двойного слова. (Используемые регистры необходимо сохранить). Напишите макрорасширение для макроккоманды `mmax X, Y, Z`.
9. Перечислите виды режимов мультипрограммирования.
- Задания, вынесенные на контрольные, коллоквиумы и итоговый экзамен оцениваются единым образом. Максимальное количество баллов за задание – 6.
- Набранные за решение заданий баллы суммируются, после чего определяются процент набранных баллов от максимального значения. Полученный процент определяет результирующую оценку.

0-39 %	40-59 %	60-79%	80-100 %
неудовлетворительно	удовлетворительно	хорошо	отлично

Итоговая оценка за лекционный курс определяется результатом письменного экзамена.

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)				
Оценка РО и соответствующие виды оценочных средств	2	3	4	5
Знания <i>Коллоквиумы, Экзамен</i>	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения <i>Контрольные работы</i>	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности непринципиального характера)	Успешное и систематическое умение
Навыки (владения, опыт деятельности) <i>Экзамен</i>	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

Соответствие результатов обучения и компетенций, в развитии которых участвует дисциплина (модуль)	
Результаты обучения	Компетенция, с частичным формированием которой связано достижение результата обучения
Знать: 1. основные принципы устройства ЭВМ, основные компоненты, их качественные и количественные характеристики; 2. язык ассемблера архитектуры IA-32; 3. свойства аппаратуры, влияющие на производительность и безопасность работы программ.	ОПК-2

<p>Уметь:</p> <ol style="list-style-type: none"> 1. составлять и отлаживать многомодульные программы на языках Си и ассемблера; 2. использовать в программах на языке ассемблера стандартную библиотеку языка Си. 	
<p>Знать:</p> <ol style="list-style-type: none"> 1. особенности архитектуры IA-32, влияние архитектуры набора команд на реализацию языков высокого уровня и систему программирования; 2. особенности реализации различных конструкций языка Си на языке ассемблера и причины, влияющие на выбор механизма реализации; 3. свойства аппаратуры, влияющие на производительность и безопасность работы программ. <p>Уметь:</p> <ol style="list-style-type: none"> 1. обосновано выбирать механизмы реализации конструкций языка Си и корректно их кодировать на языке ассемблера; 2. оценивать производительность программ на языках Си и ассемблера, используя знания об особенностях устройства аппаратуры ЭВМ. 	ОПК-3
<p>Владеть:</p> <p>Современной технологией разработки и отладки многомодульных программ на языках Си и ассемблера.</p>	ПК-3 ПК-8

8. Ресурсное обеспечение:

Основная литература:

1. Рэндал Э. Брайант, Дэвид Р. О'Халларон. Компьютерные системы: архитектура и программирование. Изд-во: БХВ-Петербург, 2005 г. - 1186 стр.
2. Е.А. Кузьменкова, В.С. Махнычев, В.А. Падарян. Семинары по курсу «Архитектура ЭВМ и язык ассемблера»: учебно-методическое пособие. Часть 1. Изд-во: МАКС ПРЕСС, 2014. - 78 стр.
3. Е.А. Кузьменкова, В.А. Падарян, М.А. Соловьев. Семинары по курсу «Архитектура ЭВМ и язык ассемблера»: учебно-методическое пособие. Часть 2. Изд-во: МАКС ПРЕСС, 2014. - 100 стр.

Дополнительная литература:

1. Э. Таненбаум, Т. Остин. Архитектура компьютера. 6 - е изд. Изд-во: Питер, 2017. - 816 стр.
2. А.В. Столяров. Программирование: введение в профессию. II: Низкоуровневое программирование. Изд-во: МАКС ПРЕСС, 2016. – 496 стр.

Материально-техническое обеспечение: компьютер и мультимедийный проектор для демонстрации слайдов в ходе лекций.

Программное обеспечение и Интернет-ресурсы:

- свободно распространяемый ассемблер nasm. (<http://www.nasm.us/>)
- сервер автоматического приема заданий <http://earth.ispras.ru/>.

9. Язык преподавания - русский.

10. Преподаватели:

доценты факультета ВМК МГУ Л.С. Корухова, Е.А. Кузьменкова.

11. Авторы программы: доцент факультета ВМК МГУ Е.А. Кузьменкова.