**Материалы для конкурса методического мастерства.**

**Четыре электронные лекции «Вхождение в программирование на Ассемблере»**

**Организация** – Финансовый Университет при правительстве Российской Федерации, Колледж информатики и программирования, г.Москва

**Автор –** преподавательСибирев Иван Валерьевич

**Лекции предназначены** для обучающихся Колледжа информатики и программирования

– специальности 10.02.05 Обеспечение информационной безопасности автоматизированных систем, относится к профессиональному модулю ПМ.02 "Защита информации в автоматизированных системах программными и программно-аппаратными средствами", междисциплинарному курсу МДК.02.03 "Машинно-ориентированное программирование для решения задач защиты информации".

**Форма проведения занятий:**проблемные лекции.

**Форма представления методического материала на конкурс:** электронные лекции.

**Тема лекций. Вхождение в программирование на Ассемблере**

**Цели лекций**

***Образовательные цели:*** ознакомить учащихся с языком машинно-ориентированного программирования; выработать у учащихся умение применять ассемблирование критически важных узких мест программных кодов на практике, развитие профессиональных навыков программиста.

***Развивающие цели:*** обеспечить условия для развития логики, алгоритмического машинно-ориентированного мышления; для формирования умения производить поиск информации, необходимой программисту для профессиональной работы, анализировать ее, наблюдать, сравнивать, применять эту информацию для решения профессиональных задач, в том числе в вопросах защиты информации и оптимизации программных кодов.

***Воспитательные цели:*** обеспечить условия для профориентации обучающихся, их вхождения в профессию программиста, воспитания дисциплинированности, аккуратности, добросовестности и для формирования коммуникативной культуры и бережного отношения к своему и чужому программному коду.

**Задачи**

– познакомить учащихся с введением в машинно-ориентированное программирование на языках высокого уровня С++.

– познакомить учащихся с введением в машинно-ориентированное программирование на примере группы языков аппаратного уровня, BF, LamPanel, FlatAssembler.

– разобраться с шаблонами и логикой написания программ управления оперативной памятью на вершине стека вызова функций на языке BF на примере казалось-бы несложно задачи вывода текста в консоль. Подходы к решению вопроса у каждого ученика вполне могут в корне отличаться…

– познакомится с командами работы с внешними портами на примере программы LamPanel в игровом задании нарисовать рисунок в лампочках портов и сделать для него анимацию.

- отработать написание программ на основе указателей, массивов и команд пересылки данных на учебных заданиях на языке С++ с использованием дизассемблера.

- отработать написание программ на основе указателей, массивов, функций и команд пересылки данных на учебных заданиях на языке FlatAssembler под DosBox и Windows10x64.

**Образовательные результаты**

***Личностные:*** развитие интереса к программированию, к решению трудных программистских задач; осознание важности изучения данной темы для расширения инструментария и возможностей программиста, для развития вариативности мышления; готовность к общению и сотрудничеству с преподавателем и сверстниками, уважительное отношение к ответу одногруппников и их мнению.

***Метапредметные:*** умение производить поиск и выбор материала на заданную тему, умение критически его переосмысливать; выбирать способы достижения результата, оценивать правильность выполнения учебной задачи; умение применять полученные знания при решении задач; умение четко формулировать собственное мнение, слушать и понимать товарищей и преподавателя.

***Предметные:*** знание основных принципов и команд в машинно-ориентированного программирования, умение оперировать понятиями регистров микропроцессора, условных и безусловных переходов, арифметических команд, указателей, процедур, функций, макросов и \*.bat файлами; умение применять их на практике при программировании на Ассемблере.

**Совершенствуются следующие универсальные учебные действия:** критическое мышление; технологическая и информационная грамотность; навыки сотрудничества.

**Краткая характеристика методических материалов**

**Актуальность содержания материала лекций**

Ассемблер – машинно-ориентированный язык программирования, предназначенный для управления битами в регистрах и оперативной памяти. Осво­ение Ассем­блера позволит обучающимся:

– понимать причины и следствия поведения программных кодов как высокого, так и машинного уровня.

– познакомиться с понятием стековой машины и с принципами организации исполняемых кодов компь­ютер­ных прог­рамм.

– получить мольный инструмент оптимизации программных и аппаратных кодов.

– отла­живать или защищать элементы компь­ютер­ные сис­темы на самом низ­ком уров­не, в том числе за счет нестандартных ассемблерных программных кодов. Только 10 % программистов занимаются вопросами реверс инжиниринга, и менее 1% пишут код на ассемблере. Эти цифры позволяют оценить, сколько придется предпринять усилий по взлому и изменению ассемблерного программного кода. В вопросах защиты информации иногда программное обеспечение пишут как этажерку из нескольких языков, для взлома требуется найти специалиста который мало-того обладал бы знанием всех использованных технологий, так ещё и на уровне тех 10% что позволяет реверс-инжениренг. Использование ассемблера резко сужает круг людей, которые за это возьмутся, что сильно повышает взломо-устойчивость системы.

– мно­гие изъ­яны безопас­ности трудно распознаваемы и про­явля­ются толь­ко на уров­не машин­ного кода, но при этом относительно легко устранимы. Например “ Windows 10 RDP несколько пользователей” до версии 1909. Можно было открыть бинарный файл и изменить несколько бит. После этого обычный Win – можно было использовать, как сетевую операционную систему, а это другая лицензия… Проблема безопасности устранилась в последующих версиях, скорее всего, простым убиранием этого ветвления. На уровне машинно-ориентированных языков почти панацеей является перемешивание команд параллельно работающих ветвей кода в пределах одной последовательности ассемблерных команд. По уже готовой программе сделать это не сложно, но после удаления всех комментариев, не факт что кто-то разберет, что было в этом мешке мелкодисперсных ассемблерных команд после перемешивания. Тем более что особенности языка это позволяют.

К тому же, некоторые команды, как не странно, пишут программное обеспечение для себя именно на Flat Assemblere. В узко специализированных аппаратных задачах и задачах защиты информации ассемблер не имеет себе равных. Языки программирования можно разделить на профессиональные и любительские по принципу возможности написания ассемблерных вставок. Именно они и определяют качество и глубину оптимизации написанных программных кодов. Если возникает необходимость переписать что-то на другом языке, это означает либо вы не изучили возможность написания ASM вставок, либо изначально был выбран не тот язык.

Это свидетельствует об актуальности программирования на Ассемблере. Именно он является перво-языком для всего остального программистского мира.

Возникновение Ассемблера датируется 1947 годом, язык содержит много диалектов, которые являются частными случаями реализации идеи машинно-ориентированного языка. Примеры из литературы часто не работают или требуют адаптации под современные реалии. В Интернет лишь каждый 10-20 раз будут попадаться примеры работающих программных кодов. Периодически встречаются посты с заведомо ложной информацией, которые новичку не под силу распознать. Программные коды, взятые из литературы, обладают высокой скоростью роста сложности и поэтому не пригодны для начинающих. Все это создает высокий входной порог сложности материала данной темы и множество проблем для начинающего программиста.

**Проблемное изложение.** При чтении лекций традиционным является повествовательно-лекционный стиль изложения, при этом обучающимся результаты программистского поиска, а также способы преодоления ряда программистских проблем и решения задач не предлагаются как устоявшиеся догмы. Большая часть обучающихся, привыкнув принимать предлагаемый материал как постулаты, не пытаются самостоятельно мыслить, творчески подходить к решению программистских задач. Ассемблер же требует противоположного. Безудержного желания и возможности изобрести заново весь материал, пройдённый в рамках основ алгоритмизации. Именно отсутствие возможности сделать это по неработающим примерам из интернета и запыленной бумажной литературы останавливает учащегося.

Нами же предлагаются учебные задачи и примеры работающих программных кодов, на которых можно себя почувствовать первооткрывателем всего остального программистского мира, как это было в эпоху изобретения и разработки первых языков программирования. Все задачи решаемы, все примеры работают, что мотивирует к обучению.

В то же время исторический путь развития программирования представляет дидактическую ценность. Этот путь хорошо можно проиллюстрировать на примере развития языков ассемблерной группы. Проблемы развития данного направления, а также проблемы, возникающие у начинающего программиста при вхождении в Ассемблер, предлагаемые обучающимся в проблемной постановке, будут способствовать развитию мышления, логики, позволят почувствовать радость открытий.

Данный материал относится к проблемным лекциям. В стиле изложения, подразумевающем диалог с аудиторией, изложены проблемы вхождения в Ассемблер. В диалоге намечаются пути решения этих проблем, названы книги и Интернет-источники, которые могут служить «дорожной картой» для самостоятельного решения названных проблем студентами. Автор лекций ведет спор с авторами некоторых Интернет-источников и с аудиторией, провоцируя слушателей к диалогу.

**Оригинальные методические решения** при освоении Ассемблера воплощены в **авторские программистские задания**, выстроенные систематически, с последовательным введением нового программистского инструментария и нарастанием сложности. Они приближают слушателя шаг за шагом к программированию на Ассемблере. Задачи дифференцированы по уровню сложности (вплоть до олимпиадных) и сопровождаются комментариями.

**Оформление электронных лекций произведено с использованием** современного программного инструментария: в «Jupiter Notebook» (2015 г.), в качестве ядра используется язык Julia (2018 г. появления). Этот язык предназначен для постановки и проведения вычислительных экспериментов, заточен под высокую производительность, имеет возможность для написания ассемблерных вставок. Julia – это язык высокого уровня, являющийся при этом «ассемблероблизким» языком. В дальнейшем это даст возможность перейти к изучению описанного технологического стека. Выбранный инструментарий позволяет запускать программные коды «прямо из лекций».

**План лекций**

**Лекция 1.**

1. Актуальность изучения Ассемблера. История языков ассемблерной группы. Проблемы вхождения в тему.

2. Эзотерические языки программирования. BrainF – младший брат Ассемблера (всего 8 команд, возможность работы со стеком вызова функции, прост в схемотехнической реализации, неожиданно упрощает понимание Ассемблера).

3. Освоение команд BrainF. Решение задач.

**Лекция 2.**

1. Обзор актуальной литературы, сайтов на тему «Программирование на Ассемблере».

2. Начинаем писать. Вывод на экран состояния регистров. ЛамПанель. 3. Рисунки с использованием ЛамПанели, анимация.

**Лекция 3.**

1.Команды пересылки данных в Ассемблере.

2. Ассемблерные команды.

3. Работа с массивами.

**Лекция 4.**

1. Изучение работы со строками. Изучение циклов.

2. Создание переменных и вывод на экран.

3. Механика условных переходов.

4.Создание циклов.

5. Сбор воедино моноблока ассемблерных кодов.

**Намечены перспективы для дальнейшей работы.**

*Материалы и программы есть, осталось описать:*

– Вызов call и 3-4 способа передачи параметров внутрь функций.  
– Макросы как средство повышения быстродействия (полный курс макросов).  
– Стек вызова функций и размещение всех локальных переменных в нем.  
– Компьютерная графика под DosBox.  
– Подключение и вызов системных библиотек с++ Windows.  
– Написание тестового стенда для изучения работы сдвигов под FASM Windows. -Ловим ошибки разработчиков FASM.  
– Компьютерная графика под WinOpenGL.

*Ближайшие перспективы:*

­– Ассемблерные команды MMX - аппаратное сложение массивов, используется обычно в криптографии и компьютерной графике.  
– Работа с fword 48 бит. Аппаратная работа с дробными числами.  
– Перебор материала [Мануал программера.flat assembler 1.71](https://www.cyberforum.ru/fasm/thread1240599.html). Выясняем какие аппаратные процедуры работают под макроассемблер Windows FASM.

**Методические указания для обучающихся по работе с лекционным материалом**

Для успешного вхождения обучающихся в программирование на Ассемблере необходимо посещение лекций, ознакомление с основной и дополнительной литературой, предлагаемой в процессе лекций, поиск актуальных Интернет-источников, разбор и анализ предлагаемых там программ, активная работа на лабораторных занятиях, выполнение учебных заданий, а также практика написания программных кодов.

Запись **лекции** – одна из форм активной самостоятельной работы обучающихся, требующая навыков и умения кратко, схематично, последовательно и логично фиксировать основные положения, выводы, обобщения, формулировки. В конце лекции преподаватель оставляет время (5 минут) для того, чтобы обучающиеся имели возможность задать уточняющие вопросы по изучаемому материалу.

Лекции прочитаны в очной форме, но учащимся предоставляется также электронный вариант лекций для дальнейшей работы с этим материалом. Для лучшего освоения материала и систематизации знаний по дисциплине, необходимо самостоятельно разбирать материалы лекций, т.е. прочитанная лекция – это материал для дальнейшей работы. В случае необходимости обращаться к преподавателю за консультацией.

Преподаватель некоторые вопросы выносит на самостоятельную работу студентов, рекомендуя ту или иную литературу. Лекции снабжены актуальными и современными ссылками на Интернет-источники. В лекциях производится анализ литературы и Интернет-источников, предлагается «дорожная карта», литература и задания для углубленного изучения предмета. В целях дифференциации обучения предложены различные траектории изучения темы: от поверхностного ознакомления с принципами работы Ассемблера и умения писать простейшие программы ­ – до профессионального владения этим языком в будущем.

Электронные лекции снабжены работающим программным кодом, что делает работу с этим материалом – интерактивной. Наблюдая за работой программ, можно глубже понять принципы, на которых она построена, что улучшит освоение материала. Кроме того, работающий код можно использовать для решения других поставленных задач.

В тексте лекций приведено множество задач. Некоторые из них решены здесь же, как иллюстрирующий пример, в режиме диалога с аудиторией. Приведены основные теоретические положения, необходимые для решения, программный код и комментарии к решению задач. Здесь же предложены аналогичные задачи для самостоятельного решения (среди которых есть олимпиадные). Задачи дифференцированы по уровню сложности и сопровождаются комментариями.

Параллельно с лекциями в курсе предусмотрены лабораторные работы. В лекциях к краткой форме приводятся задания к лабораторным работам и комментарии к их выполнению. Некоторые задания рекомендуется выполнять парами в целях развития навыков профессиональной коммуникации.

**Апробация методических материалов** произведена при ведении курса лекций в 2019-20 и 2020-21 уч. г. в группах:  
3ОИБАС-618, 3ОИБАС-718, 3ОИБАС-818, 2ИСИП-118, 2ИСИП-218  
3ОИБАС-517,3ОИБАС-617, 3ОИБАС-717,3ОИБАС-1018.