**Сочинение на тему: «ЧТО ТАКОЕ ХОРОШО И ЧТО ТАКОЕ ПЛОХО В ТОМ, ЧТО НАЗЫВАЕТСЯ БИС»**

**Р. Д. Доскоч**

Кафедра технологий программирования,

Факультет прикладной математики и информатики

Белорусского государственного университета

г. Минск, Республика Беларусь

е-mail: [fpm.doskoch@bsu.by](mailto:fpm.doskoch@bsu.by)

***Хорошо или Плохо?***

В контексте безопасности информационных систем существует несколько ключевых понятий, которые могут помочь определить, что такое "хорошо" и что такое "плохо".:

**Конфиденциальность**: Хорошие методы обеспечения безопасности информационных систем уделяют приоритетное внимание конфиденциальности конфиденциальных данных, гарантируя, что к ним имеют доступ только авторизованные пользователи. С другой стороны, неправильная практика может привести к несанкционированному доступу к конфиденциальным данным.

**Целостность**: Хорошие методы обеспечения безопасности информационных систем обеспечивают целостность данных, что означает, что они не могут быть изменены или повреждены без надлежащего разрешения. Недобросовестная практика может привести к несанкционированным изменениям данных, что может привести к неточностям или ошибкам.

**Доступность**: Хорошие методы обеспечения безопасности информационных систем гарантируют, что авторизованные пользователи могут получить доступ к необходимым им данным и системам, когда им это нужно. Неправильная практика может привести к тому, что системы или данные будут недоступны авторизованным пользователям.

**Соответствие требованиям**: Надлежащая практика обеспечения безопасности информационных систем гарантирует соответствие организации соответствующим законам, нормативным актам и отраслевым стандартам. Недобросовестная практика может привести к тому, что организация не выполнит эти требования, что может привести к юридическим и финансовым последствиям.

В целом, хорошие методы обеспечения безопасности информационных систем ставят во главу угла конфиденциальность, целостность, доступность и соответствие требованиям данных и систем организации, в то время как плохие методы могут подвергать эти вещи риску.

***Язык Ассемблера в контексте БИС***

Вот несколько примеров того, как ассемблерный код может быть использован в контексте безопасности информационных систем:

**Шифрование**: Ассемблерный код может быть использован для реализации алгоритмов шифрования, которые могут помочь защитить конфиденциальность данных, сделав их нечитаемыми для тех, у кого нет надлежащего ключа дешифрования.

**Проверка данных**: Ассемблерный код может использоваться для реализации контрольных сумм или других форм проверки данных, которые могут помочь обеспечить целостность данных путем обнаружения любых несанкционированных изменений.

**Защита системы**: Ассемблерный код может использоваться для реализации мер безопасности, таких как контроль доступа или брандмауэры, которые могут помочь защитить системы от несанкционированного доступа или вредоносной активности.

**Исправления уязвимостей**: Ассемблерный код может использоваться для исправления уязвимостей в программном обеспечении или системах, помогая повысить их общую безопасность и предотвратить эксплойты или атаки.

В целом, ассемблерный код может использоваться различными способами для повышения безопасности информационных систем и защиты от различных угроз.

***То, чему я научился в курсе БИС***

Я познакомился с прекрасным языком записи алгоритмов ­­– ассемблер. Мы познакомились с реализацией ассемблера в Dos-Box, где царствует такой вид ассемблера как **TASM,** но есть и другие немного расскажу о каждом.

TASM, NASM и MASM — все это ассемблеры, которые представляют собой программы, преобразующие код языка ассемблера в машинный код, который может быть выполнен компьютером.

TASM (Turbo Assembler) — это ассемблер, разработанный Borland для архитектуры x86. Первоначально он был выпущен в 1980-х годах и до сих пор широко используется сегодня, особенно для программирования в 16-разрядных и 32-разрядных версиях архитектуры x86.

NASM (Netwide Assembler) — это бесплатный ассемблер с открытым исходным кодом, который поддерживает широкий спектр платформ, включая x86, x86-64 и ARM. Он широко используется в Unix-подобных операционных системах и известен своей переносимостью и поддержкой различных форматов вывода.

MASM (Microsoft Macro Assembler) — это ассемблер, разработанный Microsoft для архитектуры x86. Он используется в основном на платформе Windows и входит в состав среды разработки Microsoft Visual Studio.

В целом, все эти ассемблеры представляют собой инструменты, которые позволяют разработчикам писать и собирать код на языке ассемблера для различных платформ и архитектур.

В любой программе обучения не обойтись без своих определений понятий и догм, вот несколько которые я выучил на курсе БИС:

***ПРОГРАММА – это последовательность команд, которая ПОСЛЕ ДЕКОДИРОВАНИЯ ИХ САМОЙ МАШИНОЙ(компиляции линковки) может заставить эту машину выполнить строго определенную последовательность действий***

Под **ОПЕРАЦИОННОЙ СИСТЕМОЙ** (НЕ ОБОЛОЧКОЙ) понимается совокупность аппаратных и программных средств, предназначенных для управления внешними устройствами и предоставления ПОЛЬЗОВАТЕЛЯМ (разного рода прикладным программам) услуг по управлению этими устройствами.

Ниже перечислены постулаты из области БИС:

**Постулат 1.** Не существует абсолютных методов защиты данных и программ.  
**Постулат 2.** Построение системы для защиты данных оправдано тогда и только тогда, когда стоимость разработки и внедрения данной системы меньше стоимости возможного вреда, который может быть нанесён заказчику системы и  из-за ее отсутствия.

**Постулат 3.** Невозможно управлять тем, что нельзя измерить.  
**Постулат 4.** Чем ближе к аппаратному уровень реализации системы защиты данных, тем более надёжна сама система защиты данных при прочих равных условиях.

Не могу обойти стороной понятие прерываний, которое сильно засело у меня в голове после курса. На языке ассемблера прерывания — это специальные сигналы, которые могут быть отправлены на компьютер для запроса определенной услуги или действия. Прерывания могут быть вызваны аппаратными устройствами, такими как клавиатура или мышь, или программными программами.

В ассемблере TASM вы можете использовать инструкцию INT для запуска прерывания. Команда INT принимает номер прерывания в качестве аргумента, который определяет запрашиваемую услугу или действие. Например, следующий код вызовет прерывание номер 21h:

INT 21h

Существует множество различных доступных прерываний, каждое из которых выполняет определенную функцию. Некоторые распространенные прерывания включают:

Прерывание 21h: это прерывание используется для операций ввода/вывода файлов (I/O), таких как чтение из файла или запись в файл.

Прерывание 10h: это прерывание используется для операций ввода-вывода видео, таких как настройка видеорежима или отображение текста на экране.

Прерывание 13h: это прерывание используется для операций ввода-вывода с диска, таких как чтение с диска или запись на диск.

Чтобы обработать прерывание в TASM, вы можете использовать инструкцию INT вместе со службой или функцией, которые вы хотите выполнить. Например, следующий код отобразит строку "Привет, мир!" на экране с использованием прерывания 10h:

MOV AH, 09h ; функция 09h выводит строку

MOV DX, OFFSET MESSAGE ; адрес строки для печати

INT 21h

В целом, прерывания являются полезным способом запроса определенных служб или действий в программах на языке ассемблера, а ассемблер TASM обеспечивает поддержку запуска и обработки прерываний с помощью инструкции INT.

**В заключении хочу сказать**. Существует множество различных угроз безопасности информационных систем, включая вирусы, вредоносные программы, фишинговые атаки и кибершпионаж. Для защиты от этих угроз организации и отдельные лица могут внедрять различные меры безопасности, такие как антивирусное программное обеспечение, брандмауэры, средства контроля доступа и протоколы безопасности.

Эффективные методы обеспечения безопасности информационных систем включают регулярное обновление и исправление систем, обучение пользователей передовым методам обеспечения безопасности и внедрение надежных паролей и протоколов аутентификации. Также важно иметь план реагирования на нарушения безопасности и инциденты.

В целом, безопасность информационных систем имеет решающее значение для защиты от широкого спектра угроз и обеспечения конфиденциальности, целостности и доступности информационных систем.

**ЧУТЬ НЕ ЗАБЫЛ**

вы просили показать

обязательна демонстрация каких-никаких знаний по этому курсу на примере решения задач с использованием ТОЛЬКО DosBox, языка записи алгоритмов ассемблер, соответствующих трансляторов, линковщиков

Приведу пример решения моей любимой задачи, (она кстати была первой)

**.Model Small ;** Используем модель small

**.CODE ;**Сегмент кода

**UU:**

;Адрес области памяти, относящейся к информации об экране ;располагается, начиная с адреса 40h. А в свою очередь внутри ;этой области памяти, есть байт, который содержит номер ;текущего видеорежима, причем этот байт имеет адрес 62h.

**MOV AX,40H**

**MOV ES, AX**

**MOV BH, ES: [ 62H]**

**MOV BL, 0** ; Цвет символа

**MOV AL, 10** ; Код символа, который я хочу отобразить

**MOV CX, 1 ;** Кол-во символов, которые я хочу отобразить

**MOV AH, 10 ;** Код функции для вывода символа

**INT 10H ;** Используем прерывание «Управление дисплеем»

; Функция с номером 3 может найти координаты курсора так, ; что в регистре DL , будет иметь номер столбца, а в регистре ; DH – номер строки

**MOV AH, 3**

**INT 10H**

**MOV AH, 2** ; Код функции для смены положения курсора

**ADD DL, 2** ; перемещаю курсор на 2 позиции вправо

; Повторяю действия для кода символа 13

**INT 10h**

**MOV AL, 13**

**MOV CX, 1**

**MOV ah, 10**

**int 10h**

**mov AH, 4CH** ; EXIT - TERMINATE WITH RETURN CODE

**INT 21h**

**End UU** ; Конец блока кода

