

**МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ, НАУКИ И
ИННОВАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ**

**“УТВЕРЖДАЮ”
Декан факультета
“Программный инжиниринг”**

_____ **О.Б. Рузibaев**
«_____» _____ **2024 г.**

**ВОПРОСНИК ПО ИТОГОВОЙ КОНТРОЛИ
ПО КУРСУ REVERSE ENGINEERING**

Область знания:	600 000	– Информационно-коммуникационные технологии
Сфера образования :	610 000	– Информационно-коммуникационные технологии
Направление образования:	60610600	– Программный инжиниринг

ТАШКЕНТ – 2024

1.	Введение в предмет «Reverse engineering». Основные понятия.
2.	Применение реверс – инжиниринга в разных отраслях.
3.	Обзор инструментальных программ.Отладчики.
4.	Дизассемблеры. Распаковщики. Дамперы. Редакторы ресурсов. Мониторы. Копировщики защищенных дисков.
5.	Эмулирующие отладчики и эмуляторы. Вводная информация об эмуляторах.
6.	Области применения эмуляторов.
7.	Аппаратная виртуализация. Обзор популярных эмуляторов. DOSBox. Bochs и QEMU. VMware. Microsoft Virtual PC. Xen.
8.	Выбор подходящего эмулятора. Защищенность. Расширяемость. Доступность исходных текстов.
9.	Качество эмуляции. Встроенный отладчик. Встроенный отладчик.
10.	Инструментарий для UNIX и Linux. Отладчики. Дизассемблеры. Шпионы. Шестнадцатеричные редакторы. Дамперы.
11.	Ассемблеры. Философия ассемблера.
12.	Объяснение ассемблера на примерах C++.
13.	Ассемблерные вставки как тестовый стенд. Необходимый инструментарий.
14.	Сравнение ассемблерных трансляторов. Основополагающие критерии. MASM. TASM. FASM. NASM. NASM.
15.	Введение в защитные механизмы. Классификация защит по роду секретного ключа. Надежность защиты.
16.	Распространенные ошибки реализации защитных механизмов.
17.	Защита от несанкционированного копирования и распространения серийных номеров.
18.	Защита испытательным сроком и ее слабые места. Реконструкция алгоритма.
19.	Общие рекомендации по защите информации. Защита от модификации на диске и в памяти.
20.	Противодействие дизассемблеру. Антиотладочные приемы. Антимониторы.
21.	Знакомство с дизассемблером.
22.	Пакетные дизассемблеры и интерактивные дизассемблеры.
23.	Использование пакетных дизассемблеров.
24.	Знакомство с отладкой. Введение в отладку.
25.	Дизассемблер и отладчик в одной упряжке. Точки останова на функции API.
26.	Точки останова на сообщения. Точки останова на данные. Раскрытие стека. Отладка DLL.
27.	Идентификация ключевых структур языков высокого уровня.
28.	Идентификация функций. Методы распознавания функций. Перекрестные ссылки.
29.	Автоматическая идентификация функций посредством IDA Pro. Пролог. Эпилог.
30.	Идентификация встраиваемых (inline) функций.
31.	Идентификация стартовых функций.
32.	Идентификация функции WinMain.
33.	Идентификация функции DllMain.
34.	Идентификация функции main консольных Windows-приложений.
35.	Идентификация виртуальных функций. Идентификация чистой виртуальной функции.
36.	Совместное использование виртуальной таблицы несколькими экземплярами объекта. Копии виртуальных таблиц. Связный список. Вызов через шлюз.
37.	Идентификация производных функций. Идентификация виртуальных таблиц.
38.	Идентификация конструктора и деструктора. Объекты в автоматической памяти — ситуация, когда конструктор/деструктор идентифицировать невозможно.
39.	Идентификация конструктора/деструктора в глобальных объектах. Виртуальный деструктор. Виртуальный конструктор.
40.	Идентификация объектов, структур и массивов. Идентификация структур. Идентификация объектов. Объекты и экземпляры.
41.	Идентификация this. Идентификация new и delete. Идентификация new. Идентификация delete. Подходы к реализации кучи.
42.	Идентификация библиотечных функций.

43.	Идентификация аргументов функций. Соглашения о передаче параметров. Определение количества и типа передачи аргументов.
44.	Адресация аргументов в стеке. Идентификация локальных стековых переменных. Адресация локальных переменных. Детали технической реализации.
45.	Идентификация механизма выделения памяти. Инициализация локальных переменных. Размещение массивов и структур. Выравнивание в стеке.
46.	Идентификация регистровых и временных переменных. Регистровые переменные.
47.	Временные переменные. Создание временных переменных при пересылках данных и вычислении выражений.
48.	Создание временных переменных для сохранения значения, возвращенного функцией, и результатов вычисления выражений. Область видимости временных переменных.
49.	Идентификация глобальных переменных. Техника восстановления перекрестных ссылок. Отслеживание обращений к глобальным переменным контекстным поиском их смещения в сегменте кода [данных].
50.	Идентификация констант и смещений. Идентификация литералов и строк. Определение типа непосредственного операнда. Определение типа строк.
51.	Идентификация разветвляющих конструкций. Идентификация конструкций IF — THEN — ELSE.
52.	Идентификация конструкций SWITCH — CASE — BREAK.
53.	Идентификация циклов. Циклы с предусловием. Циклы с постусловием. Циклы со счетчиком.