**Основы отладки и анализа программ**

**1. Ключевые концепции**

**Отладчики и их функционал**  
Отладчик — инструмент для интерактивного контроля выполнения программ, предоставляющий:

* Пошаговое выполнение с возможностью паузы
* Просмотр и изменение переменных во время работы
* Установку контрольных точек
* Анализ цепочки вызовов функций

**Сравнительная таблица инструментов**

| **Функция** | **Отладчик** | **Дизассемблер** |
| --- | --- | --- |
| Режим | Динамический (с выполнением) | Статический (анализ без запуска) |
| Основное назначение | Поиск логических ошибок | Обратная разработка |
| Возможности | Изменение состояния программы | Анализ структуры кода |

**Типичные категории ошибок**

* Логические (некорректная реализация алгоритмов)
* Проблемы с памятью (утечки, переполнения буфера)
* Исключительные ситуации (нарушения доступа)
* Проблемы многопоточности (гонки, взаимные блокировки)

**Виды точек останова**

1. Простые — срабатывают при достижении строки
2. Условные — активируются по заданному выражению
3. Аппаратные — используют возможности процессора

**2. Практические методы отладки**

**Работа в Visual Studio**

1. Открытие проекта
2. Запуск отладки (F5 или Debug → Start Debugging)
3. Использование:
   * F10 для обхода функций
   * F11 для детализации вызовов

**Управление выполнением**

| **Действие** | **Горячие клавиши** | **Назначение** |
| --- | --- | --- |
| Продолжить | F5 | До следующей точки |
| Шаг с обходом | F10 | Пропуск внутренних функций |
| Шаг с входом | F11 | Детальный анализ |
| Выход | Shift+F11 | Возврат на уровень выше |

**Инструменты анализа**

* Locals: автоматический вывод локальных переменных
* Watch: мониторинг произвольных выражений
* Immediate Window: интерактивная консоль для проверки

**Диагностика сложных сбоев**

1. Активация всех исключений (Exception Settings)
2. Варианты диагностики:
   * Логирование в файл
   * Анализ дампов памяти
   * Трассировка системных вызовов

**3. Анализ исполняемого кода**

**Принципы работы дизассемблеров**  
Преобразование машинного кода в:

* Ассемблерные листинги
* Псевдокод высокого уровня

**Популярные инструменты**

* Ghidra (open-source от NSA)
* IDA Pro (профессиональный стандарт)
* Binary Ninja (облачное решение)
* radare2 (консольный инструмент)

**Методы анализа безопасности**

1. Поиск строковых констант авторизации
2. Локализация функций сравнения
3. Анализ условных переходов

**4. Практические примеры**

**Обработка исключений**  
Для предотвращения деления на ноль:

c

Copy

Download

if (divisor == 0) return ERROR\_INVALID\_PARAMETER;

**Этапы реверс-инжиниринга**

1. Идентификация точек ввода/вывода
2. Анализ преобразований данных
3. Восстановление логики через графы вызовов

**5. Правовые аспекты**

**Разрешенные действия**

* Анализ собственного ПО
* Исследование уязвимостей с согласия владельца
* Образовательная деятельность

**Запрещенные практики**

* Обход систем защиты
* Модификация чужих программ
* Распространение декомпилированного кода

*Примечание: В большинстве юрисдикции несанкционированный реверс-инжиниринг нарушает законы об авторском праве.*