Приднестровский Государственный Университет им. Т.Г. Шевченко

Бендерский Политехнический Филиал

Кафедра «Промышленность и информационные технологии»

**Обратное проектирование алгоритма.**

Выполнил:

Студент 3 курса, БК22АР52ИС2 группы, дневной формы обучения,

специальность: «Информационные системы и программирование»

Череш Артём Александрович

Руководитель учебной практики  
Лазовский А.В.

Бендеры, 2025

**Обратное проектирование алгоритма (или реверс-инжиниринг алгоритма)** — это процесс анализа уже существующего программного обеспечения, системы или кода с целью понять, **как работает алгоритм**, который лежит в его основе, **без доступа к его исходному коду или документации**.

**Что такое обратное проектирование?**

Это метод, при котором пытаются «развернуть» готовую систему, чтобы:

* восстановить структуру и логику алгоритма,
* понять, какие шаги он выполняет,
* выявить входные и выходные данные,
* определить, как данные обрабатываются.

**Когда используется обратное проектирование алгоритма?**

1. **Анализ стороннего ПО** — если есть доступ только к исполняемому файлу, а не к исходному коду.
2. **Миграция старых систем** — если нужно переписать алгоритм, но оригинальная документация утеряна.
3. **Безопасность** — для поиска уязвимостей, вредоносных алгоритмов или защиты от копирования.
4. **Анализ конкурентов** — попытка понять, как работает алгоритм, реализованный в чужом продукте.

**Пример: Обратное проектирование алгоритма сортировки**

Допустим, у вас есть программа, которая сортирует список чисел, но вы не знаете, **какой алгоритм сортировки** она использует (быстрая, пузырьковая, слиянием и т. д.).

1. Вы подаёте на вход разные данные (например, уже отсортированные, случайные, обратно отсортированные).
2. Замеряете время выполнения и изучаете порядок операций (например, через трассировку или дизассемблирование).
3. По характеру поведения (числу сравнений, обменов) делаете вывод, какой алгоритм был применён.

**Инструменты и методы**

* **Дизассемблеры** (например, IDA Pro, Ghidra) — преобразуют машинный код в понятную форму.
* **Отладчики** (например, x64dbg, OllyDbg) — позволяют наблюдать выполнение программы по шагам.
* **Логический анализ** — запись поведения на разных входных данных.
* **Символьное выполнение** — анализ кода на уровне логики и условий.

**Заключение**

Обратное проектирование алгоритма — это сложный, но полезный процесс, который позволяет восстановить или понять работу скрытого или утерянного алгоритма. Он требует технической подготовки, знаний в области программирования, архитектуры компьютеров и анализа данных.