**Лабораторная работа #3. Разработка администратора ресурсов (АР) устройства «Шифратор OTP»**

Задача: на основе предыдущих лабораторных работ разработать администратор ресурсов QNX Neutrino, имитирующий работу устройства «Шифратор OTP», а также клиентскую программу, выполняющую тестовый запрос к АР . Суть OTP (шифра Вернама) раскрывалась ранее в задание к лабе #2.

Платформа: QNX Neutrino v6.6, x86.

**Требования к клиентской программе:**

* использует дескриптор-ориентированные интерфейсы POSIX API для обращения к АР: [open()](http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fd%2Fdevctl.html), [devctl()](http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fd%2Fdevctl.html), [close()](http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fd%2Fdevctl.html);
* читает в ***бинарном*** виде тестовые данные из файла ограниченного размера (1024 байт);
* отправляет посредством функции [devctl()](http://www.qnx.com/developers/docs/6.5.0/index.jsp?topic=%2Fcom.qnx.doc.neutrino_lib_ref%2Fd%2Fdevctl.html) команду CYPHER\_TEXT, определённую с помощью макроса [DIOTF](http://www.qnx.com/developers/docs/qnxcar2/index.jsp?topic=%2Fcom.qnx.doc.neutrino.resmgr%2Ftopic%2Fmessages_HANDLING_devctl.html) в общем разделяемом файле клиента и АР «otp\_dev.h» и данные, упакованные в структуру OtpContext (параметры ЛКГ, тестовые данные в виде статического массива фиксированного размера);
* получает данные от АР;
* сохраняет данные в файл;
* пути к входному и выходному файлу получаются из параметров командной строки.

Вход: текстовый файл с открытым текстом.

Выход: текстовый файл с шифротекстом.

Пример клиентской программы:

<http://www.qnx.com/developers/docs/7.0.0/#com.qnx.doc.neutrino.resmgr/topic/skeleton_ClientSide.html>

**Требования к АР:**

* за основу использует [скелет однопоточного АР](http://www.qnx.com/developers/docs/7.0.0/#com.qnx.doc.neutrino.resmgr/topic/skeleton_SIMPLE_ST_EG.html);
* [регистрируется](http://www.qnx.com/developers/docs/7.0.0/#com.qnx.doc.neutrino.lib_ref/topic/r/resmgr_attach.html) и функционирует с использованием библиотеки resmgr\_library ОС QNX Neutrino;
* реализует обработку клиентского запроса с помощью функции [io\_devctl](https://it.wikireading.ru/2461)(), передаваемую библиотеке io\_funcs через соответствующую структуру указателей в скелете;
* извлекает структуру данных из сообщения, производит приведение типа void \* к типу структуры OtpContext с помощью [reinterpret\_cast](https://ru.cppreference.com/w/cpp/language/reinterpret_cast);
* валидирует данные, в случае ошибки передаёт её клиенту;
* генерирует ключ с помощью ЛКГ соответственно параметрам, переданным в запросе;
* производит побитовое XOR для текста и ключа в том же потоке;
* отправляет результат клиенту.

**Тесты**

1. Аналогично тесту лабы #2: двойное преобразование даёт исходный текст.
2. Проверка того, что и клиент, и сервер, работают в отдельных процессах (команда ps UNIX), клиент работает с сервером как с АР.