

УТВЕРЖДЕН

643.ИАИА.00001-01 ТЗ 01-ЛУ

ШАБЛОН ПРОГРАММЫ-СЕРВЕРА ДЛЯ QNX6

Техническое задание

643.ИАИА.00001-01 ТЗ 01

Листов 10

2018

Литера

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

Наименование разработки «Шаблон программы-сервера для *QNX6*».

Целью является создание основы для написания компонентов программных комплексов для *QNX6*.

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Личная инициатива.

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Шаблон предназначен для разработчиков программного обеспечения для *QNX6*. Он является основой для написания компонентов программных комплексов. Самостоятельного применения не имеет.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

Программные комплексы для QNX6, как правило, состоят из нескольких компонентов, взаимодействующих между собой посредством механизма обмена сообщениями. Практически все модули построены по серверной схеме, то есть имеют однотипную архитектуру. Это обстоятельство позволяет выделить общую для всех программ часть и использовать ее в дальнейшем в качестве основы для разработки программ.

4.1. Требования к функциональным характеристикам

Выделить фиксированную часть, обеспечивающую функционирование программы как сервера. Для выполнения действий, специфических для прикладной программы, использовать внешние вызовы.

На базе фиксированной части создать шаблон программы. Он должен включать в себя следующие заготовки:

- описания типов данных для обмена сообщениями;
- библиотека функций, являющихся обертками для обмена сообщениями;
- обработка параметров, передаваемых через строку запуска программы;
- реализация внешних вызовов;
- фактические обработчики сообщений.

В шаблон включить обработку сообщений, поступающих от операционной системы, а также сообщений, которые могут часто использоваться в прикладных программах.

Шаблон должен поддерживать механизм расширения функциональности путем добавления обработки новых сообщений.

Шаблон должен поддерживать механизм наследования функциональности. В этом случае создается новый модуль, который обрабатывает определенные сообщения, а при поступлении сообщений других типов вызывает обработчик из модуля, созданного ранее.

Должно быть выработано соглашение об именовании используемых типов данных, функций и файлов.

Шаблон должен быть написан на языках *C* и *C++*. Оба варианта концептуально и текстуально должны быть максимально близки друг к другу.

Шаблон должен сопровождаться примером использования на каждом языке.

4.2. Дополнительные требования

Не предъявляются.

К дополнительным требованиям, в частности, относятся:

- требования к надежности;
- требования к условиям эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортировке и хранению;
- специальные требования.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

В состав программной документации должны входить:

- техническое задание;
- пояснительная записка;
- описание программы;
- руководство программиста.

Документация должна быть оформлена в соответствии с требованиями ГОСТ 19.

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Использование шаблона упростит и ускорит написание компонентов, так как в нем полностью обеспечивается функционирование программы как сервера, и разработчику остается реализовать только прикладные аспекты модуля.

Надежность программ повысится ввиду того, что в шаблоне учтены все вопросы организации сервера, а при разработке модуля с нуля, некоторые моменты могут быть упущены.

Применение единого шаблона улучшит понимание и сопровождение программных комплексов, особенно при совместной деятельности нескольких разработчиков.

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Не определяются.

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Не определяется.