

Требования к ПО телекоммуникационных систем

I. Основные (обязательные) требования

Их нарушение делает невозможным использование по назначению разработанных программных средств

II. Дополнительные (вспомогательные) требования

Влияют главным образом на эффективность ПО

Чтобы получить конкурентоспособный программный продукт высокого качества без дополнительных затрат времени и финансов, все эти требования должны учитываться с самого начала разработки.

I. Основные требования к ПО

1. **Функциональная полнота** – обеспечение программной реализации всех требуемых функций
2. **Высокая производительность** – ПО должно обеспечивать скорость обработки данных, достаточную для реализации всех заданных функций с соблюдением временных ограничений на процессы обслуживания вызовов.

Оценивается с помощью следующих показателей, которые относятся ко всей системе, но существенно зависят и от характеристик ПО:

- число обслуживаемых вызовов в ЧНН;
- время реакции на поступивший сигнал

3. Высокая надежность при минимальных эксплуатационных затратах.

С учетом возможных программных ошибок должно обеспечиваться общее требование к современным системам коммутации с программным управлением – *не более 2 часов полного простоя за 40 лет.*

II. Дополнительные требования к ПО

1. Минимальная стоимость хранения программ и данных

Обеспечивается организацией многоуровневой виртуальной памяти

2. Открытость и гибкость ПО

Обеспечивает минимум затрат при внесении изменений с целью модернизации ПО в процессе длительной эксплуатации.

3. Простота организации ПО

Облегчает разработку, производство, изучение и эксплуатацию программ.

Выдержки из учебника Терехова А.Н. «Технологии программирования» (2006)

Часть II. Технология программирования встроенных систем реального времени

Встроенная система – это программно-аппаратная система, в которой вычислительные средства управляют аппаратурой ***в реальном масштабе времени***.

Масштаб времени зависит от типа системы, но обычно это микросекунды или миллисекунды.

Главная особенность встроенных систем – это необходимость во многих случаях уложиться в заданный интервал времени

(иначе безвозвратно пропадает информация или

Типичными примерами встроенных систем являются телефонные станции, системы вооружения, роботы, медицинское оборудование и т.д.

Многие специалисты относят создание встроенных систем к числу самых трудных задач: здесь сосредоточены практически все проблемы **системного программирования**.

Особенности ПО встроенных систем

1. Повышенная надежность программных средств, поскольку ошибки в программах могут приводить к катастрофическим последствиям.

При этом отладка ПО реального времени осложняется трудностями проведения экспериментов.

Следовательно, в программах могут оставаться ошибки, которые проявляются только при определенном сочетании внешних и внутренних параметров.

Чтобы «поймать» такую редкую ошибку, придется

2. Реактивность ПО, т.е. способность всегда укладываться в заданные временные рамки.

Отсюда вытекает необходимость **тщательного анализа** времени протекания различных процессов и выделения **критически важных задач**.

3. Компактность. Это приводит к необходимости оптимизировать не только время выполнения программы, но требования к вычислительным ресурсам (т.е. мощность процессора и объем используемой памяти).

4. Работа с базами данных. Например, все однотипные АТС от одного производителя имеют **одинаковый пакет программ**, однако работа отдельного экземпляра станции с учетом **конкретных условий эксплуатации** обеспечивается благодаря информации, которая хранится

5. Взаимодействие с человеком. Это приводит к необходимости организовать **удобный интерфейс** для оперативного отображения текущего состояния всех элементов системы с программным управлением.

Кроме того, требуются **дополнительные** программные средства для поддержки специального **языка взаимодействия «человек-машина» (Man-Machine Language – MML).**

Выдержки из диссертации Б. Деккера (фирма Ericsson, 2000)

Требования к технологиям программирования для систем автоматической коммутации

1. Обработка очень большого количества параллельных задач.
2. Многие действия должны выполняться не позднее заданного момента времени или в пределах определенного интервала.
3. Средства вычислительной системы имеют распределенную структуру.
4. Активное взаимодействие с аппаратными средствами объекта управления.

5. Очень большой объем программных средств.
6. Сложная функциональность ПО при многообразии взаимодействующих элементов.
7. Непрерывная эксплуатация в течение длительного периода.
8. Организация эксплуатационного сопровождения программных средств (устранение ошибок, реконфигурация и т.п.) без остановки системы.
9. Наличие очень жестких требований к качеству и надежности программных средств.
10. Устойчивая работа при аппаратных отказах и программных ошибках.

Принципы построения ПО телекоммуникационных систем

1. Принцип модульности

Все ПО разбивается на отдельные части, которые называют **модулями**.

Модули должны быть:

- небольшими по размеру;
- функционально-законченными

(это приводит к меньшему количеству входов/выходов, что облегчает стыковку модулей и их взаимодействие).

Существуют различные типы модулей:

- ▣ **программные модули** – обеспечивают обработку данных;
- ▣ **информационные модули** – содержат данные, которые подлежат обработке.

В технологиях объектно-ориентированного программирования (ООП) широкое распространение получают программные элементы (компоненты), которые называют **объектами**.

Текущее **состояние объекта** характеризуется некоторыми данными, а его **функциональные возможности** реализуются с помощью методов (процедур).

ОБЪЕКТ = ПРОГРАММЫ + ДАННЫЕ

В программах для управления интерактивными системами такие объекты называют **конечными автоматами** (Finite Message Machine – **FMM**)



Преимущества использования принципа модульности ПО

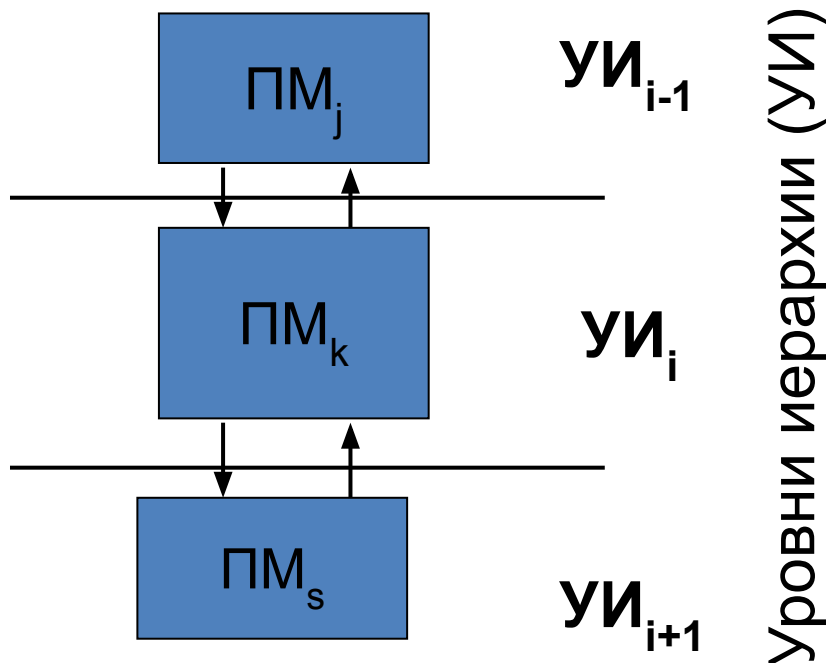
- Ускорение сроков разработки за счет организации параллельной работы с отдельными частями ПО
- Минимизация стоимости разработки, т.к. для работы с достаточно простыми модулями не требуются программисты очень высокой квалификации
- Простота сборки рабочей версии программ для конкретного образца телекоммуникационной системы
- Высокая гибкость ПО и простота внесения изменений при модернизации программ

2. Принцип иерархичности

Этот принцип затрагивает только взаимодействие между отдельными программными модулями (ПМ).

При одноуровневом построении (peer-to-peer – P2P) очень трудно держать под контролем все процессы взаимодействия в сложной системе.

Поэтому более предпочтительным становится иерархический порядок взаимодействия.



Условные обозначения:

↓ – запуск модуля

↑ – возврат управления

Правила иерархического взаимодействия

1. Любой модуль может быть запущен только другим модулем предшествующего УИ.
2. После выполнения требуемых функций управление возвращается в точку, из которой происходил запуск модуля.
3. Во время своей работы ПМ может обращаться к другим модулям более низкого УИ.

Недостатки:

- ▣ Слишком жесткая схема взаимодействия
- ▣ В некоторых ситуациях процесс взаимодействия замедляется.

Достоинство:

- ▣ Все взаимодействия стандартизованы, поэтому такую схему легче настраивать, проще выявлять в ней возможные нарушения.

3. Принцип приоритетности

С учетом срочности обработки информации и важности выполняемых функций формируются приоритетные уровни (ПУ), а затем программные модули распределяются по этим уровням:

- ПУ₀ – наиболее важные функции системы, которые должны выполняться с минимальными задержками;
- ПУ₁ – функции обработки информации, при выполнении которых можно допустить незначительные задержки;
- самый низкий ПУ – задачи, которые могут решаться в «фоновом режиме», т.е. при отсутствии других более важных задач.

Виды приоритетов

- Абсолютный приоритет – заявка с более высоким приоритетом прерывает обслуживание заявки более низкого приоритета.
- Относительный приоритет – прерывания отсутствуют и приоритет учитывается только при формировании очереди.
- Смешанные приоритеты – между приоритетными уровнями действует абсолютный приоритет, а внутри одного уровня применяется относительный приоритет.
- **Преимущество принципа приоритетности:**
требуемая производительность системы обеспечивается без использования мощных вычислительных средств
- **Недостаток:** усложняется организация вычислительного процесса