Типовые задачи в разработке ПО и их решения

Alexey Shrub

2012-09-13

Введение

- Зачем?

Введение

- Зачем?
- Уровни образования

Методологии разработки

- ООП

Методологии разработки

- ООП
- Функциональный стиль

Coding conventions

- Код для человека

Coding conventions

- Код для человека
- Именование, отступы

Coding conventions

- Код для человека
- Именование, отступы
- Всегда пишите код так, будто сопровождать его будет склонный к насилию психопат, который знает, где вы живете. — Martin Golding

Конфигурирование

- Зачем?

Конфигурирование

- Зачем?
- Файлы или база?

Конфигурирование

- Зачем?
- Файлы или база?
- Форматы файлов и библиотеки.

Журналирование

- Зачем?

Журналирование

- Зачем?
- Что? (Уровни)

Журналирование

- Зачем?
- Что? (Уровни)
- Куда? (БД или файлы)

Профилирование

- Зачем?

Профилирование

- Зачем?
- Инструменты.

Протоколы интеграции и маршалинг

- Интеграция: файлы, базы, очереди, вызов процедур

Протоколы интеграции и маршалинг

- Интеграция: файлы, базы, очереди, вызов процедур
- Вызовы процедур: SOAP, XML::RPC, JSON::RPC, в какой-то мере REST

Протоколы интеграции и маршалинг

- Интеграция: файлы, базы, очереди, вызов процедур
- Вызовы процедур: SOAP, XML::RPC, JSON::RPC, в какой-то мере REST
- Сериализация/Маршалинг: JSON, YAML

Хранилища данных

- РСУБД (PostgreSQL)

Хранилища данных

- РСУБД (PostgreSQL)
- Ключ-значение (Redis, Memcached, Riak)

Хранилища данных

- РСУБД (PostgreSQL)
- Ключ-значение (Redis, Memcached, Riak)
- Документоориентированные (mongoDB)

Хранилища данных

- РСУБД (PostgreSQL)
- Ключ-значение (Redis, Memcached, Riak)
- Документоориентированные (mongoDB)
- Очереди

Технологии параллелизма

- process

Технологии параллелизма

- process
- thread

Технологии параллелизма

- process
- thread
- event loop

Масштабирование ПО

- Один в поле не воин

Масштабирование ПО

- Один в поле не воин
- Как распределить запросы (с сессиями и без них)

Масштабирование ПО

- Один в поле не воин
- Как распределить запросы (с сессиями и без них)
- Репликация. Партицирование

Простота и сложность

- Отладка кода вдвое сложнее, чем его написание. Так что если вы пишете код настолько умно, насколько можете, то вы по определению недостаточно сообразительны, чтобы его отлаживать. — Brian W. Kernighan.

Простота и сложность

- Отладка кода вдвое сложнее, чем его написание. Так что если вы пишете код настолько умно, насколько можете, то вы по определению недостаточно сообразительны, чтобы его отлаживать. — Brian W. Kernighan.
- Есть два способа спроектировать систему. Один — сделать ее настолько простой, что в ней очевидно не будет недостатков, а второй — сделать ее настолько сложной, что в ней не будет очевидных недостатков. Первый способ намного более труден. — Тони Хоар

Простота и сложность

- Отладка кода вдвое сложнее, чем его написание. Так что если вы пишете код настолько умно, насколько можете, то вы по определению недостаточно сообразительны, чтобы его отлаживать. — Brian W. Kernighan.
- Есть два способа спроектировать систему. Один — сделать ее настолько простой, что в ней очевидно не будет недостатков, а второй — сделать ее настолько сложной, что в ней не будет очевидных недостатков. Первый способ намного более труден. — Тони Хоар
- Unix-way

Сложность алгоритмов

- Временная сложность алгоритма (в худшем случае)
  - $O(log\ n)$ - бинарный поиск по отсортированному массиву
  - $O(n)$ - поиск в одномерном массиве полным перебором
  - $O(n^2)$ - поиск в двумерном массиве полным перебором, многие виды сортировок
  - $O(c^n)$ - решение задачи коммивояжёра методами динамического программирования
  - $O(n!)$ - решение задачи коммивояжёра полным перебором

Сложность алгоритмов

- Временная сложность алгоритма (в худшем случае)
  - $O(log\ n)$ - бинарный поиск по отсортированному массиву
  - $O(n)$ - поиск в одномерном массиве полным перебором
  - $O(n^2)$ - поиск в двумерном массиве полным перебором, многие виды сортировок
  - $O(c^n)$ - решение задачи коммивояжёра методами динамического программирования
  - $O(n!)$ - решение задачи коммивояжёра полным перебором
- В лучшем и средних случаях

Сложность алгоритмов

- Временная сложность алгоритма (в худшем случае)
  - $O(log\ n)$ - бинарный поиск по отсортированному массиву
  - $O(n)$ - поиск в одномерном массиве полным перебором
  - $O(n^2)$ - поиск в двумерном массиве полным перебором, многие виды сортировок
  - $O(c^n)$ - решение задачи коммивояжёра методами динамического программирования
  - $O(n!)$ - решение задачи коммивояжёра полным перебором
- В лучшем и средних случаях
- Пространственная сложность - по раходу памяти

Оптимизация

- Всему своё время

Оптимизация

- Всему своё время
- Компилятор умный

Оптимизация

- Всему своё время
- Компилятор умный
- Узкие места на нужном языке

Установка ПО

- Версии

Установка ПО

- Версии
- Зависимости

Установка ПО

- Версии
- Зависимости
- Откат

Локализация и интернационализация (технологии и подходы)

- Интернационализация - чтоб могло (utf8, gettext, .po файлы и инструменты для них)

Локализация и интернационализация (технологии и подходы)

- Интернационализация - чтоб могло (utf8, gettext, .po файлы и инструменты для них)
- Локализация - для конкретной страны (перевод, форматы дат, денег)

Конечные автоматы?

- Регулярные выражения

Конечные автоматы?

- Регулярные выражения
- PCRE

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл
- "Программист-прагматик. Путь от подмастерья к мастеру"Э. Хант, Д. Томас

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл
- "Программист-прагматик. Путь от подмастерья к мастеру"Э. Хант, Д. Томас
- "Рефакторинг. Улучшение существующего кода"Мартин Фаулер

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл
- "Программист-прагматик. Путь от подмастерья к мастеру"Э. Хант, Д. Томас
- "Рефакторинг. Улучшение существующего кода"Мартин Фаулер
- "Приемы объектно-ориентированного проектирования. Паттерны проектирования"Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл
- "Программист-прагматик. Путь от подмастерья к мастеру"Э. Хант, Д. Томас
- "Рефакторинг. Улучшение существующего кода"Мартин Фаулер
- "Приемы объектно-ориентированного проектирования. Паттерны проектирования"Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес
- "Этюды для программистов"Чарльз Уэзерелл

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл
- "Программист-прагматик. Путь от подмастерья к мастеру"Э. Хант, Д. Томас
- "Рефакторинг. Улучшение существующего кода"Мартин Фаулер
- "Приемы объектно-ориентированного проектирования. Паттерны проектирования"Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес
- "Этюды для программистов"Чарльз Уэзерелл
- "Искусство программирования"Дональд Кнут

"Читай много, но не очень много книг."Бенджамин Франклин

- "Совершенный код"Стив Макконнелл
- "Программист-прагматик. Путь от подмастерья к мастеру"Э. Хант, Д. Томас
- "Рефакторинг. Улучшение существующего кода"Мартин Фаулер
- "Приемы объектно-ориентированного проектирования. Паттерны проектирования"Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес
- "Этюды для программистов"Чарльз Уэзерелл
- "Искусство программирования"Дональд Кнут
- "Регулярные выражения"Джеффри Фридл

Что ещё?

"I belive that you can never be a good programmer if you know only the syntax of the language but not how the compilation or run-time enviroment is implemented. For example, ... a Java programmer should know why a thread may never get control in a uniprocessor setup even if it is not blocked."
"I am of the opinion that an application developer should master at least the following six major areas: user interfaces, persistance, interprocess communication and networking, parsing and code generation, the Web, and the operating system"
Sriram Srinivasan "Advanced Perl Programming"

Вопросы?

Исходники презентации (LaTeX, Beamer):

https://github.com/worldmind/typical-dev-tasks-presentation-ru.git

Feedback to: ashrub@yandex.ru