Современный C++

Максим Федоренко

ИАТЭ НИЯУ МИФИ
Кафедра ПМ

3 декабря 2016 г.

Системы управления версиями

Проблема

- Проект – множество файлов с исходным кодом
- Код изменяется
- За изменениями нужно следить
- Необходимо поддерживать несколько ***версий*** проекта

Наивное решение

- Хранить копии файлов
- Копии файлов пронумерованы

Недостатки:

- Приходится хранить несколько практически идентичных копий
- Требуется повышенное внимание и дисциплина
- Возникают ошибки
- Это неудобно

Централизованные системы

- Хранилище (репозиторий) находится на **сервере**
- Сервер совершает операции над репозиторием
- Пользователь работает с ***рабочей копией***
- Изменения отправляются на сервер, формируется новая ***ревизия***

Примеры:

- CVS
- Subversion (SVN)

Недостатки

- Клиент-серверная архитектура не обладает гибкостью
- Слабая поддержка ветвления
- Линейный подход к разработке

Распределённые системы

- Каждая рабочая копия – полноценный репозиторий
- Для работы не нужен сервер
- Хранилища синхронизируются между собой

Примеры:

- Git
- Mercurial

Git

- Линус Торвальдс
- Ядро Linux

Принципы работы

- Рабочая копия
- Добавление изменений в индекс
- Фиксация изменений
- Синхронизация между репозиториями

Удалённые репозитории

- Командная работа над проектом
- Работа на разных устройствах
- ***Голые*** репозитории
- **GitHub** – хостинг удалённых репозиториев

Ветки

- Одна из главных достоинств Git
- Ветка – множество ***коммитов***
- Главная ветка – **master**
- Слияние веток

Инструменты для работы с Git

- Командная строка
- SourceTree
- GitKraken

Git Flow

- Стандартные подходы
- Несколько ключевых веток: **master** и **develop**
- ...

Системы сборки

Системы сборки

- CMake

Заключение

Заключение

- Заключение