

Университет ИТМО

Кафедра вычислительной техники

# Отчет по прохождению практики

Студента

Р3311 группы

Морозова С.Д.

Руководитель  
Соснин В.В.

Санкт-Петербург

2016

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Система компьютерной верстки T<sub>E</sub>X(L<sup>A</sup>T<sub>E</sub>X)</b>	<b>4</b>
2.1	Краткое описание . . . . .	4
2.2	Сравнение L <sup>A</sup> T <sub>E</sub> X и MS Word . . . . .	6
2.3	Выбор инструмента редактирования . . . . .	7
<b>3</b>	<b>Системы контроля версий</b>	<b>9</b>
3.1	Краткое описание . . . . .	9
3.2	Git . . . . .	9
3.2.1	Особенности . . . . .	10
3.2.2	Основные команды . . . . .	12
3.3	GitHub . . . . .	13
<b>4</b>	<b>Паралельные вычисления</b>	<b>14</b>
4.1	История . . . . .	14
4.2	Что-нибудь из теории . . . . .	14
4.3	Что-нибудь еще... . . . .	14
<b>5</b>	<b>Функции замера времени</b>	<b>15</b>
5.1	Принцип работы . . . . .	15
5.2	Windows . . . . .	15
5.2.1	func1 . . . . .	15

5.2.2	func2 . . . . .	15
5.2.3	... . . . .	15
5.3	Linux . . . . .	15
5.3.1	func4 . . . . .	15
5.3.2	func5 . . . . .	15
5.3.3	... . . . .	15
5.4	Кроссплатформенные . . . . .	16
5.4.1	func7 . . . . .	16
5.4.2	func8 . . . . .	16
5.4.3	... . . . .	16
5.5	Проблемы и сложности замеров времени при параллельный вычислениях . . . . .	16
<b>6</b>	<b>Практическая часть?</b>	<b>17</b>
6.1	Описание экспериментальной программы . . . . .	17
6.2	Результаты работы программы . . . . .	17
6.3	Выводы . . . . .	17
<b>7</b>	<b>Вывод по производственной практике</b>	<b>18</b>
<b>8</b>	<b>Список литературы</b>	<b>19</b>

# 1 Введение

Тема прохождения практики — параллельные вычисления. Цель задания — сравнить различные функции в языке C, которые можно использовать для измерения времени работы параллельных программ.

Однако требования руководителя практики таковы, что перед тем как приступить к выполнению основного задания нужно ознакомиться с системой компьютерной вёрстки TeX (LaTeX), которая должна использоваться для написания отчёта, и ознакомиться с системой контроля версий Git, с последующим созданием учетной записи на сайте GitHub или аналогичном.

## 2 Система компьютерной верстки $\text{\TeX}$ ( $\text{\LaTeX}$ )

### 2.1 Краткое описание

$\text{\TeX}$  — система компьютерной вёрстки с формулами, разработанная американским профессором информатики Дональдом Кнутом. Название происходит от греческого слова  $\tau\epsilon\chi\upsilon\eta$  — «искусство», «мастерство», поэтому последняя буква читается как русская Х. Хотя  $\text{\TeX}$  является системой набора и верстки, развитые возможности макроязыка  $\text{\TeX}$  делают его Тьюринг-полным языком программирования.

$\text{\TeX}$  работает с боксами (box) и клеем (glue). Бокс — двумерный объект прямоугольной формы, характеризуется тремя величинами (высота, ширина, глубина). Элементарные боксы — это буквы, которые объединяются в боксы-слова, которые в свою очередь сливаются в боксы-строки, боксы-абзацы и т.д.

Между боксами располагается клей, который имеет некоторую ширину по умолчанию и степени увеличения/уменьшения этой ширины. Объединяясь в бокс более высокого порядка, боксы могут шевелиться, но после того как найдено оптимальное решение, это состояние закрепляется, и полученный бокс выступает как единое целое.

Интересный факт. На версии 3.0 дизайн был заморожен, поэтому в новых версиях не будет добавления новой функциональности, только исправление ошибок. Версия  $\text{\TeX}$ 'а асимптотически прибли-

жается к числу  $\pi$ . Это факт говорит о том, что последняя версия 3.14159265 (январь 2014) является крайне стабильной и возможны лишь мелкие исправления. Дональд Кнут заявил, что последнее обновление (сделанное после его смерти) сменит номер версии на  $\pi$ , и с этого момента все ошибки станут особенностями.

L<sup>A</sup>T<sub>E</sub>X — созданный Лесли Лэмпортом набор макрорасширений (или макропакет) системы компьютерной вёрстки T<sub>E</sub>X, который облегчает набор сложных документов. Стоит отметить, что как и любой другой макропакет<sup>1</sup> L<sup>A</sup>T<sub>E</sub>X не может расширить возможности T<sub>E</sub>X (все, что можно сделать в одном пакете можно сделать и в любом другом). Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая набор текста на нескольких языках, нумерацию разделов и формул, размещение иллюстраций и таблиц на странице, ведение библиографии и др. Все это делает L<sup>A</sup>T<sub>E</sub>X крайне удобным инструментом для написания научных статей, диссертаций и т.п..

---

<sup>1</sup> Так же существуют Plain T<sub>E</sub>X, AMS-T<sub>E</sub>X, AMS-LaT<sub>E</sub>X и т.д.

## 2.2 Сравнение $\text{\LaTeX}$ и MS Word

В качестве сравнения — перечислим плюсы и минусы  $\text{\LaTeX}$  перед MS Word(а так же всеми его аналогами).

Плюсы  $\text{\LaTeX}$ :

- Проста работы с любыми математическими формулами
- Кроссплатформенность
- Без особых трудностей можно получить сноски, список литературы, оглавление, список таблиц, указатель и т. п.
- Имеется несколько стандартных стилей (книга, статья, доклад, письмо), с помощью которых получаются документы очень высокого полиграфического качества
- Гибкая работа с логической структурой текста
- Язык международного обмена по математике и физике (большинство научных издательств принимают тексты в печать только в этом формате)

Минусы  $\text{\LaTeX}$ :

- Не является системой типа WYSIWYG <sup>2</sup>
- При серьезных отклонениях от стандартных стилей документов требуется достаточно сложное программирование

То есть, выбирая между  $\text{\LaTeX}$  и MS Word, стоит обратить внимание на то, какой текст вы собираетесь печатать, насколько нестандартный будет стиль текста, на его примерный объем. В некоторых случаях достаточно использовать MS Word, в других — использование  $\text{\LaTeX}$  может заметно упростить работу.

## 2.3 Выбор инструмента редактирования

В ходе изучения всех возможных вариантов работа с  $\text{\LaTeX}$  для создания данного отчета, была выбрана программа Textmaker <sup>3</sup>. Выбор Textmaker'а обусловлен следующими его особенностями:

- Автоматическая подсветка синтаксиса
- Функция автодополнения команд  $\text{\LaTeX}$
- Соккрытие блоков кода (Code folding)

---

<sup>2</sup>What You See Is What You Get(Что видишь, то и получишь). Стоит отметить, что существуют дистрибутивы  $\text{\TeX}$  в которых есть попытки реализовать WYSIWYG. Например платный дистрибутив BaKoMa  $\text{\TeX}$  + текстовый редактор BaKoMa  $\text{\TeX}$  Word.

<sup>3</sup>Официальный сайт Textmaker: <http://www.xmlmath.net/textmaker/>



- Быстрая навигация по структуре документа
- Указание на строку с ошибкой, для быстрой отладки
- Интегрированный просмотр PDF

## 3 Системы контроля версий

### 3.1 Краткое описание

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

СКВ широко используются при разработке программного обеспечения, для хранения кодов разрабатываемых программ. Однако данные системы подходят не только программистам. Художники, которые хотят сохранять каждое изображение/эскиз своей работы, писатели пишущие книги или научные статьи, бухгалтеры, которые хранят разные версии отчетов и т.д., все они могут использовать СКВ для достижения своих целей.

Иначе говоря СКВ можно применять в любых областях в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов.

### 3.2 Git

Git — созданная Линусом Торвальдсом, распределенная система контроля версий.

### 3.2.1 Особенности

Одной из основных особенностей Git состоит в способе хранения данных. В принципе, большинство других систем хранит информацию как список изменений (патчей) для файлов. Эти системы (CVS, Subversion, Perforce, Bazaar и другие) относятся к хранимым данным как к набору файлов и изменений, сделанных для каждого из этих файлов во времени, как показано на Рис. 1

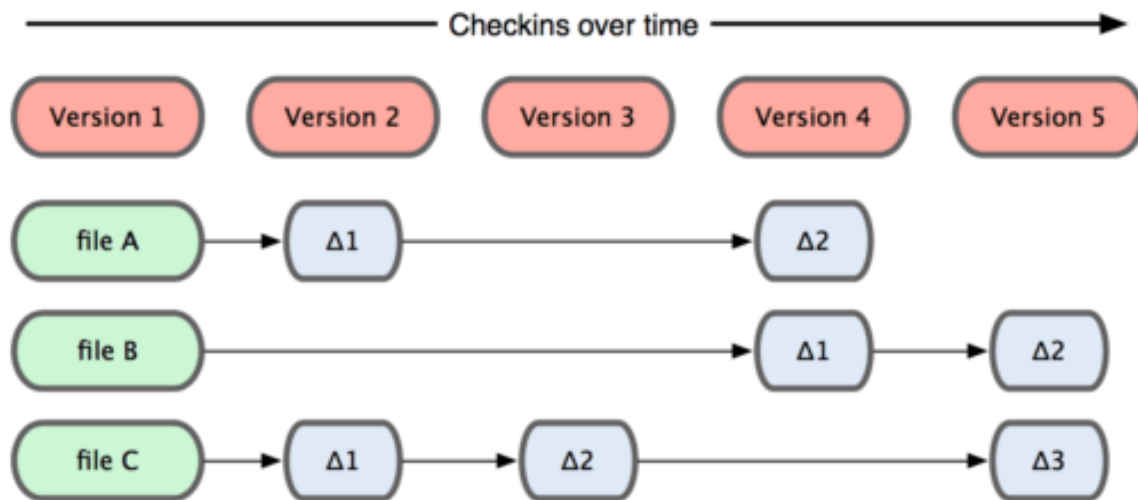


Рис. 1: Другие системы хранят данные как изменения к базовой версии для каждого файла.

Git не хранит свои данные в таком виде. Вместо этого Git считает хранимые данные набором слепков небольшой файловой системы. Каждый раз, когда вы фиксируете текущую версию проекта, Git, по сути, сохраняет слепок того, как выглядят все файлы проекта на текущий момент. Ради эффективности, если файл не менялся, Git не сохраняет файл снова, а делает ссылку на ранее сохранённый файл. То, как Git подходит к хранению данных, похоже на Рис. 2

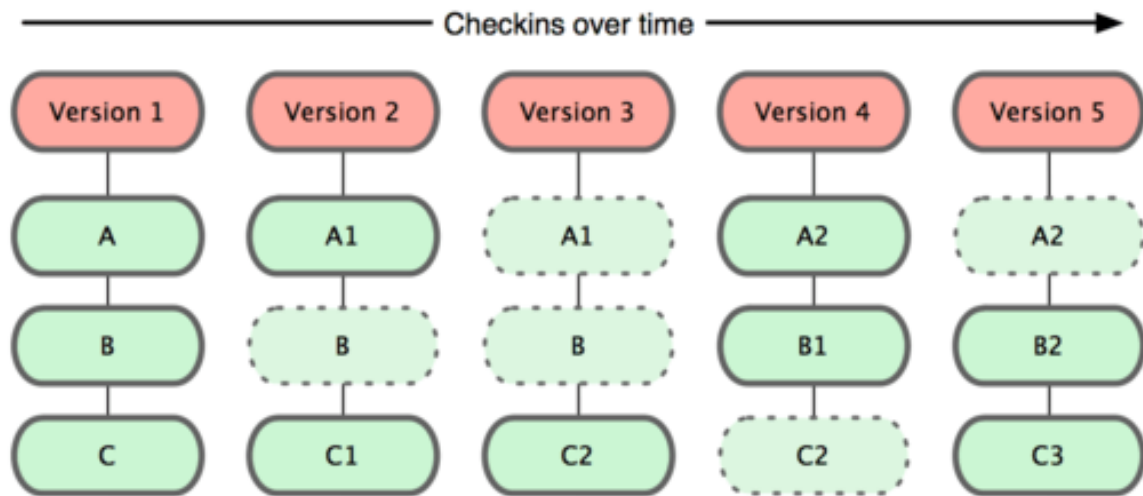


Рис. 2: Git хранит данные как слепки состояний проекта во времени.

За счет этого, для большинства операций в Git нужны только локальные ресурсы и файлы. Что в свою очередь определяет два основных преимущества Git перед остальными СКВ.

- Быстродействие. Поскольку вся история проекта хранится локально у вас на диске, большинство операций кажутся практически мгновенными (в отличие от централизованных систем, где практически на каждую операцию накладывается сетевая задержка).
- Возможность работать (делать коммиты) без доступа к сети или VPN.

### 3.2.2 Основные команды

В целом, следующая картинка (Рис. 3) наглядно демонстрирует основные команды Git, знание которых достаточно, чтобы начать им пользоваться.

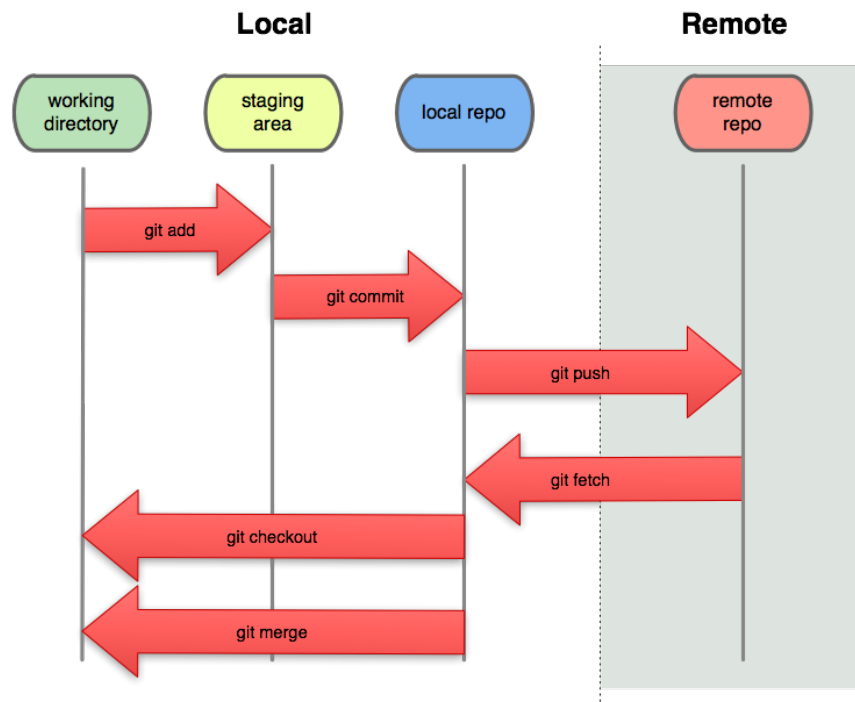


Рис. 3: Основные команды при работе с Git.

## 3.3 GitHub

## 4 Паралельные вычисления

### 4.1 История

### 4.2 Что-нибудь из теории

### 4.3 Что-нибудь еще...

## 5 Функции замера времени

### 5.1 Принцип работы

### 5.2 Windows

#### 5.2.1 func1

#### 5.2.2 func2

#### 5.2.3 ...

### 5.3 Linux

#### 5.3.1 func4

#### 5.3.2 func5

#### 5.3.3 ...



## 5.4 Кроссплатформенные

### 5.4.1 func7

### 5.4.2 func8

### 5.4.3 ...

## 5.5 Проблемы и сложности замеров времени при параллельный вычислениях

## 6 Практическая часть?

6.1 Описание экспериментальной программы

6.2 Результаты работы программы

6.3 Выводы

## 7 Вывод по производственной практике

## 8 Список литературы