

Содержание

ГЛАВА 1. ВВЕДЕНИЕ В GIT	9
1.1. НЕОБХОДИМОСТЬ РЕЗЕРВНОГО КОПИРОВАНИЯ	10
1.2. ЭТАПЫ РАЗВИТИЯ	11
1.3. ЧТО В ИМЕНИ ТВОЕМ?	12
ГЛАВА 2. УСТАНОВКА GIT	13
2.1. ИСПОЛЬЗОВАНИЕ ДВОИЧНЫХ ДИСТРИБУТИВОВ LINUX	14
Debian/Ubuntu.....	15
Другие дистрибутивы.....	16
2.2. ПОЛУЧЕНИЕ ИСХОДНОГО КОДА	17
2.3. СБОРКА И УСТАНОВКА	18
2.4. УСТАНОВКА GIT В WINDOWS	20
Установка пакета Cygwin Git	21
Установка msysGit	22
ГЛАВА 3. НАЧАЛО ИСПОЛЬЗОВАНИЯ	25
3.1. КОМАНДНАЯ СТРОКА GIT	26
3.2. БЫСТРОЕ ВВЕДЕНИЕ В GIT	29
Создание начального репозитория	29
Добавление файлов в ваш репозиторий.....	30
Неясные сообщения об ошибках.....	32
Настройка автора коммита (фиксации)	32
Вносим другой коммит (фиксацию).....	33
Просмотр ваших фиксаций	33
Просмотр разницы между коммитами (фиксациями)	35
Удаление и переименование файлов в вашем репозитории.....	36
Копирование вашего репозитория	37
3.3. ФАЙЛЫ КОНФИГУРАЦИИ	38
Настройка псевдонимов	40
Резюме	41



ГЛАВА 4. БАЗОВЫЕ ПОНЯТИЯ GIT	42
4.1. БАЗОВЫЕ ПОНЯТИЯ	43
Репозитории	44
Типы объектов Git.....	44
Индекс	46
Глобально уникальные идентификаторы	47
Git отслеживает контент.....	47
Путь по сравнению с содержанием	48
Pack-файлы	49
4.2. ИЗОБРАЖЕНИЯ ХРАНИЛИЩА ОБЪЕКТОВ	50
4.3. ПОНЯТИЯ GIT В ДЕЙСТВИИ	53
Внутри каталога .git.....	53
Объекты, хэши, блобы.....	55
Откуда мы знаем, что хэш SHA1 уникален?.....	55
Файлы и деревья.....	56
Примечание относительно использования SHA1	57
Иерархия деревьев	59
Фиксации.....	60
Теги	62
ГЛАВА 5. УПРАВЛЕНИЕ ФАЙЛАМИ. ИНДЕКС.....	65
5.1. ВСЕ ОБ ИНДЕКСЕ.....	67
5.2. КЛАССИФИКАЦИЯ ФАЙЛОВ В GIT.....	68
5.3. ИСПОЛЬЗОВАНИЕ GIT ADD	70
5.4. НЕКОТОРЫЕ ЗАМЕЧАНИЯ ОТНОСИТЕЛЬНО ИСПОЛЬЗОВАНИЕ GIT COMMIT.....	73
5.5. ИСПОЛЬЗОВАНИЕ GIT RM	75
5.6. ИСПОЛЬЗОВАНИЕ GIT MV.....	78
5.7. ЗАМЕЧАНИЕ ОТНОСИТЕЛЬНО ОТСЛЕЖИВАНИЯ ПЕРЕИМЕНОВАНИЙ.	80
5.8. ПРОБЛЕМЫ С ОТСЛЕЖИВАНИЕМ ПЕРЕИМЕНОВАНИЯ	81
5.9. ФАЙЛ .GITIGNORE.....	82

ГЛАВА 6. КОММИТЫ (ФИКСАЦИИ)	85
6.1. АТОМАРНЫЕ НАБОРЫ ИЗМЕНЕНИЙ	87
6.2. ИДЕНТИФИКАЦИЯ ФИКСАЦИЙ	88
Абсолютные имена фиксации	89
Ссылки и символьные ссылки	90
Относительные имена фиксации.....	92
6.3. ИСТОРИЯ ФИКСАЦИЙ	95
Просмотр старых фиксации	95
Графы фиксации	98
Использование gitk для просмотра графа фиксации	102
Диапазоны фиксации	103
6.4. ДОСТИЖИМОСТЬ В ГРАФАХ	104
6.5. ПОИСК ФИКСАЦИЙ	108
Использование git bisect	108
Использование git blame	114
Использование Pickaxe	115
ГЛАВА 7. ВЕТКИ В GIT	117
7.1. ПРИЧИНЫ ИСПОЛЬЗОВАНИЯ ВЕТОК	118
7.2. ВЕТКА ИЛИ ТЕГ?	119
Имена веток.....	120
Правила именования веток	120
7.3. ИСПОЛЬЗОВАНИЕ ВЕТОК	121
7.4. СОЗДАНИЕ ВЕТОК	123
7.5. ВЫВОД ИМЕН ВЕТОК	124
7.6. ПРОСМОТР ВЕТОК	125
7.7. ПЕРЕКЛЮЧЕНИЕ ВЕТОК И ВЫГРУЗКА ФАЙЛОВ	128
Базовое переключение между ветками	128
Переключение веток при имеющихся незафиксированных изменениях.....	129
Объединение изменений в другую ветку	131
Создание новой ветки и переключение на нее	133
Отсоединение головы ветки.....	135

7.8. УДАЛЕНИЕ ВЕТОК	136
ГЛАВА 8. РАЗЛИЧИЯ В GIT	139
8.1. ФОРМЫ КОМАНДЫ GIT DIFF	141
8.2. ПРОСТОЙ ПРИМЕР GIT DIFF	144
8.3. КОМАНДА GIT DIFF И ДИАПАЗОНЫ ФИКСАЦИЙ	148
8.4. КОМАНДА GIT DIFF С ОГРАНИЧЕНИЕМ ПУТИ	151
Сравнение как Subversion (SVN) и Git получают разницы	154
ГЛАВА 9. СЛИЯНИЯ	157
9.1. ПРИМЕРЫ СЛИЯНИЯ	159
Подготовка к слиянию	159
Объединение двух веток	160
Слияние с конфликтом	162
9.2. РАБОТА С КОНФЛИКТАМИ СЛИЯНИЯ	167
Обнаружение конфликтных файлов.....	168
Исследование конфликтов	169
git diff с конфликтами	170
git log с конфликтами	173
Как Git отслеживает конфликты.....	175
Завершение разрешения конфликта.....	177
Отмена или перезапуск слияния	179
9.3. СТРАТЕГИИ СЛИЯНИЯ.....	179
Вырожденные слияния.....	183
Обычные слияния.....	185
Рекурсивные слияния	185
Слияния осьминога.....	186
Специальные слияния.....	187
9.4. ПРИМЕНЕНИЕ СТРАТЕГИЙ СЛИЯНИЯ	187
Использование стратегий ours и subtree	189
Драйверы слияния	190
Как Git думает о слияниях.....	191
Слияния и объектная модель Git.....	191

Слияние squash.....	192
Почему не выполнять слияние каждого изменения одно за другим?	193
ГЛАВА 10. ИЗМЕНЕНИЕ КОММИТОВ (ФИКСАЦИЙ)...	195
10.1. ЗАЧЕМ ПРОИЗВОДИТЬ ИЗМЕНЕНИЯ ИСТОРИИ	197
10.2. ИСПОЛЬЗОВАНИЕ КОМАНДЫ GIT RESET	199
10.3. ИСПОЛЬЗОВАНИЕ КОМАНДЫ GIT CHERRY-PICK	209
10.4. ИСПОЛЬЗОВАНИЕ КОМАНДЫ GIT REVERT	211
10.5. КОМАНДЫ RESET, REVERT И CHECKOUT	212
10.6. ИЗМЕНЕНИЕ ПОСЛЕДНЕЙ ФИКСАЦИИ.....	214
10.7. ПЕРЕБАЗИРОВАНИЕ ФИКСАЦИЙ	217
Использование команды git rebase -i.....	220
Сравнение rebase и merge.....	225
ГЛАВА 11. STASH И REFLOG	233
11.1. STASH	234
11.2. REFLOG	246
ГЛАВА 12. УДАЛЕННЫЕ РЕПОЗИТАРИИ	253
12.1. ПОНЯТИЯ РЕПОЗИТАРИЯ.....	256
Чистый репозиторий и репозиторий разработки	256
Клоны репозитариев	257
Удаленные	258
Ветки отслеживания	260
12.2. ССЫЛКИ НА ДРУГИЕ РЕПОЗИТАРИИ	261
Ссылки на удаленные репозитории	262
Refspec	264
12.3. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ УДАЛЕННЫХ РЕПОЗИТАРИЕВ	267
Создание авторитетного репозитория	268
Создание вашей собственной удаленной origin.....	270
Разработка в вашем репозитории	273
Передача ваших изменений	274

Добавление нового разработчика	275
Получение обновлений репозитория.....	278
12.4. УДАЛЕННАЯ КОНФИГУРАЦИЯ	285
Использование команды git remote	285
Использование git config	287
Редактирование файла конфигурации вручную.....	288
Несколько удаленных репозитариев	288
12.5. РАБОТА С ВЕТКАМИ ОТСЛЕЖИВАНИЯ	289
Создание веток отслеживания	289
Вперед и назад	293
12.6. ДОБАВЛЕНИЕ И УДАЛЕНИЕ УДАЛЕННЫХ ВЕТОК	295
12.7. ЧИСТЫЕ (BARE) РЕПОЗИТАРИИ И КОМАНДА GIT PUSH	297

