|  |
| --- |
| ## Front matter title: “Шаблон отчёта по лабораторной работе№2” subtitle: “Система контроля версий GIT” author: “Шуваев Сергей” |
| ## Generic otions lang: ru-RU toc-title: ” 1. Цель работы. 2. 3.2. Теоретическое введение 3. 3.2.1. Системы контроля версий. Общие понятия 4. 3.2.2. Система контроля версий Git 5. 3.2.3. Основные команды git. 6. 3.2.4. Стандартные процедуры работы при наличии центрального репозитория 7. 3.3. Техническое обеспечение 8. 3.4. Порядок выполнения лабораторной работы 3.4.1. Настройка github 3.4.2. Базовая настройка git 3.4.3. Создание SSH ключа 3.4.4. Сознание рабочего пространства и репозитория курса на основе Шаблона. 3.4.5. Сознание репозитория курса на основе шаблона 3.4.6. Настройка каталога курса Самостоятельная работа” |
| ## Bibliography bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl |
| ## Pdf output format toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs: name: english ## I18n babel babel-lang: russian babel-otherlangs: english ## Fonts mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9 ## Biblatex biblatex: true biblio-style: “gost-numeric” biblatexoptions: - parentracker=true - backend=biber - hyperref=auto - language=auto - autolang=other\* - citestyle=gost-numeric ## Pandoc-crossref LaTeX customization figureTitle: “Рис.” tableTitle: “Таблица” listingTitle: “Листинг” lofTitle: “Список иллюстраций” lotTitle: “Список таблиц” lolTitle: “Листинги” ## Misc options indent: true header-includes: - |

# 1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

# 2 Задание

• Создать базовую конфигурацию для работы с git.  
• Создать ключ SSH.  
• Создать ключ PGP.  
• Настроить подписи git.  
• Зарегистрироваться на Github.  
• Создать локальный каталог для выполнения заданий по предмету.

# 3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Table 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно об Unix см. в (**gnu-doc:bash?**; **newham:2005:bash?**; **zarrelli:2017:bash?**; **robbins:2013:bash?**; **tannenbaum:arch-pc:ru?**; **tannenbaum:modern-os:ru?**).

# 4 Выполнение лабораторной работы

Существует несколько доступных серверов репозиториев с возможностью бесплатного размещения данных. Например, http://bitbucket.org/, https://gith ub.com/ и https://gitflic.ru. Для выполнения лабораторных работ предлагается использовать Github. Создадим учётную запись на сайте https://github.com/ и заполним основные данные. Базовая настройка git Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория: git config –global user.name “” git config –global user.email “[work@mail](mailto:work@mail)”

Настроим utf-8 в выводе сообщений git: git config –global core.quotepath false Зададим имя начальной ветки (будем называть её master): git config –global init.default Branch master Параметр autocrlf: git config –global core.autocrlf input

Сначала сделаем предварительную конфигурацию git Зададим имя начальной ветки (будем называть её master): переходим на сайт регистрации git hub Создание SSH ключа создаем учетную запись Далее необходимо загрузить сгенерированный открытый ключ. Откройте терминал и создайте каталог для предмета «Архитектура компью- тера» Сознание репозитория курса на основе шаблона В открывшемся окне задайте имя репозитория (Repository name) Откройте терминал и перейдите в каталог курса:cd ~/work/study/2022-2023/“Архитектура компьютера”Клонируйте созданный репозиторий: Ссылку для клонирования можно скопировать на странице созданного репозитория Code -> SSH: Проверьте правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.:

# 5 Выводы

Практически изучил идеологию средств контроля версий, приобрел практические навыки по работе с системой git.

# Список литературы

• .Colvin H. VirtualBox : An Ultimate Guide Book on Virtualization with Virtual Box.  
• 2. Newham C. Learning the bash Shell: Unix Shell Programming. O’Reilly Media,  
• 2005. 354 с.  
• 3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.  
• 4. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.  
• 5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874