

Front matter

title: "Отчёт по лабораторной работе №1

Информационная безопасность" subtitle: "Настройка рабочего пространства и конфигурация операционной системы на виртуальную машину. Система контроля версий Git. Язык разметки Markdown" author: "Выполнила: Данзанова Саяна,

НФИбд-01-21, 1032217624"

Generic otions

lang: ru-RU toc-title: "Содержание"

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 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:

- parenttracker=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" lolTitle: "Листинги"

Misc options

indent: true header-includes:

- \usepackage{indentfirst}
- \usepackage{float} # keep figures where there are in the text

- \floatplacement{figure}{H} # keep figures where there are in the text

Цель работы

Настроить рабочее пространство для лабораторных работ, приобрести практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов. Изучить идеологию и применение средств контроля версий, освоить умения по работе с git. Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

Теоретическое введение

Oracle VM VirtualBox — это мощная и бесплатная виртуализационная платформа, разработанная корпорацией Oracle, которая позволяет пользователям создавать и управлять виртуальными машинами на своих компьютерах. [1]

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. [2]

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.

Примеры команд для Git:

Название команды	Описание команды
git clone	Клонирование репозитория на ПК
git commit -m "Initial	

Markdown - это легковесный язык разметки, который широко используется для создания форматированного текста в веб-среде. Его простота и читаемость делают его популярным среди разработчиков, писателей и блогеров. Синтаксис Markdown состоит из простых символов и правил форматирования, которые позволяют создавать заголовки, списки, ссылки, изображения и другие элементы веб-страниц без необходимости использовать сложные HTML-теги. Он также легко читается в исходном виде и может быть конвертирован в различные форматы, такие как HTML, PDF или документы Microsoft Word, делая Markdown удобным инструментом для создания содержательного и красочного контента в интернете. [3]

Выполнение лабораторной работы

Установка и конфигурация операционной системы на виртуальную машину


Virtual Box

 (рис. 1. Общие настройки){ #fig:001 width=70% height=70% }

 (рис. 2. Имя и путь ОС){ #fig:002 width=70% height=70% }


 (рис. 3. Размер памяти и число процессоров){ #fig:003 width=70% height=70% }


 (рис. 4. Виртуальный жесткий диск){ #fig:004 width=70% height=70% }


 (рис. 5. Итог настроек){ #fig:005 width=70% height=70% }


 (рис. 6. Носители){ #fig:006 width=70% height=70% }


 (рис. 7. Стартовое меню установки){ #fig:007 width=70% height=70% }

 (рис. 8. Выбор программ){ #fig:008 width=70% height=70% }

 (рис. 9. KDUMP){ #fig:009 width=70% height=70% }

 (рис. 10. Сеть и имя узла){ #fig:009 width=70% height=70% }

 (рис. 11. Пароль пользователя){ #fig:010 width=70% height=70% }

 (рис. 12. Создание пользователя){ #fig:011 width=70% height=70% }


 (рис. 13. Успешное завершение установки и перезагрузка системы){ #fig:012 width=70% height=70% }


Переход в ОС Linux


 (рис. 14. Успешное создание пользователя){ #fig:013 width=70% height=70% }

 (рис. 15. Подключение гостевых настроек){ #fig:014 width=70% height=70% }

Домашнее задание


 (рис. 16. dmesg){ #fig:015 width=70% height=70% }

 (рис. 17. dmesg | less, версия ядра линукс, частота процессора, модель процессора){ #fig:016 width=70% height=70% }

 (рис. 18. Объем доступной оперативной памяти){ #fig:017 width=70% height=70% }

 (рис. 19. Тип обнаруженного гипервизора){ #fig:018 width=70% height=70% }

 (рис. 20. Тип файловой системы корневого раздела){ #fig:019 width=70% height=70% }

 (рис. 21. Последовательность монтирования файловых систем){ #fig:020 width=70% height=70% }


Управление версиями

Настройка github


Уже имеется аккаунт: <https://github.com/SayanaDanzanova>

Установка программного обеспечения


Установка git-flow

 (рис. 22. Установка git-flow){ #fig:021 width=70% height=70% }

Базовая настройка git


 (рис. 23. Базовая настройка git){ #fig:022 width=70% height=70% }

Создание ключа SSH


 (рис. 24. Создание ключа SSH){ #fig:023 width=70% height=70% }


 (рис. 25. Ключ SSH успешно добавлен на git){ #fig:024 width=70% height=70% }

Создание ключа PGP

 (рис. 26. Создание ключа PGP(1)){ #fig:025 width=70% height=70% }


Добавление PGP ключа в GitHub

 (рис. 27. Создание ключа PGP(2)){ #fig:026 width=70% height=70% }


 (рис. 28. Создание ключа PGP(3)){ #fig:027 width=70% height=70% }


 (рис. 29. Ключ PGP успешно добавлен на git){ #fig:028 width=70% height=70% }

Настройка автоматических подписей коммитов git

 (рис. 30. Настройка авто-коммитов){ #fig:029 width=70% height=70% }


Настройка gh


 (рис. 31. Авторизация gh){ #fig:030 width=70% height=70% }

 (рис. 32. Успешная авторизация gh){ #fig:031 width=70% height=70% }

Создание репозитория курса на основе шаблона

 (рис. 33. Создание репозитория по шаблону){ #fig:032 width=70% height=70% }

 (рис. 34. Рекурсивное клонирование репозитория){ #fig:033 width=70% height=70% }

 (рис. 36. Репозиторий успешно создан){ #fig:034 width=70% height=70% }

Далее дополнительно настроили репозиторий курса согласно заданию, написали отчет на Markdown и конвертировали .md в .pdf и .docs при помощи команды make и файла Makefile.

Отправили все изменения и коммиты на GitHub.

Вывод

Были настроено рабочее пространство для лабораторных работ, приобретены практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов. Были изучены идеология и применение средств контроля версий, освоены умения по работе с git. Были приобретены практические навыки оформления отчётов с помощью легковесного языка разметки Markdown.

Список литературы. Библиография

[1] Документация по Virtual Box: <https://www.virtualbox.org/wiki/Documentation>

[2] Документация по Git: <https://git-scm.com/book/ru/v2>

[3] Документация по Markdown: <https://learn.microsoft.com/ru-ru/contribute/markdown-reference>