

Отчёт по лабораторной работе №3

Коновалова Татьяна Борисовна

7 Сентября 2023

РУДН, Москва, Россия

Отчет по лабораторной работы №1

Запуск виртуальной машины

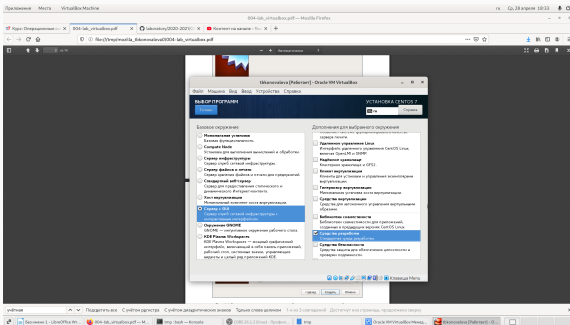
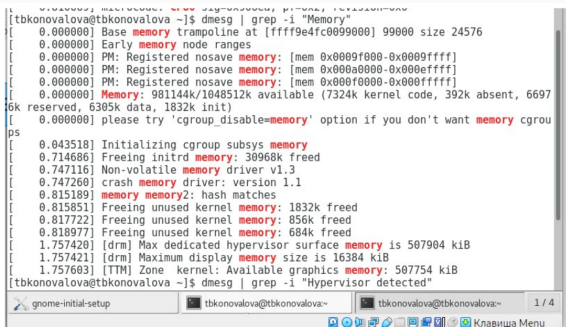


Рис. 2: Продолжение настроек после запуска машины

Работа с виртуальной машиной (Домашняя работа №1)

- На данном слайде представлено одно из заданий Домашней работы к лабораторной работе №1



```
[tbkonovalova@tbkonovalova ~]$ dmesg | grep -i "Memory"
[ 0.000000] Base memory trampoline at [ffff9e4fc0099000] 99000 size 24576
[ 0.000000] Early memory node ranges
[ 0.000000] PM: Registered nosave memory: [mem 0x0009f000-0x0009ffff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000effff]
[ 0.000000] PM: Registered nosave memory: [mem 0x000f0000-0x000fffff]
[ 0.000000] Memory: 981144k/1048512k available (7324k kernel code, 392k absent, 6697
6k reserved, 6305k data, 1832k init)
[ 0.000000] please try 'cgroup_disable=memory' option if you don't want memory cgroup
ps
[ 0.043518] Initializing cgroup subsys memory
[ 0.714686] Freeing initrd memory: 30968k freed
[ 0.747116] Non-volatile memory driver v1.3
[ 0.747260] crash memory driver: version 1.1
[ 0.815189] memory memory2: hash matches
[ 0.815851] Freeing unused kernel memory: 1832k freed
[ 0.817722] Freeing unused kernel memory: 856k freed
[ 0.818977] Freeing unused kernel memory: 684k freed
[ 1.757420] [drm] Max dedicated hypervisor surface memory is 507904 kiB
[ 1.757421] [drm] Maximum display memory size is 16384 kiB
[ 1.757603] [TTM] Zone kernel: Available graphics memory: 507754 kiB
[tbkonovalova@tbkonovalova ~]$ dmesg | grep -i "Hypervisor detected"
```

Рис. 3: Работа в виртуальной машине

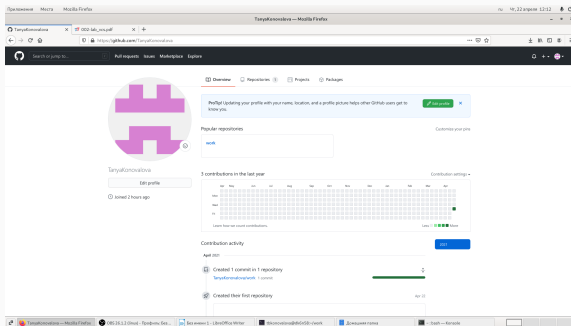


Рис. 4: Страница на GitHub

- На следующем скриншоте представлены команды git, которые позволяют отправить созданные файлы в репозиторий на GitHub

```
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ touch lab02/2 .txt
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git commit -am "first commit"
На ветке main
Ваша ветка обновлена в соответствии с «origin/main».

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
..

ничего не добавлено в коммит, но есть неотслеживаемые файлы (используйте «git add», чтобы отслеживать их)
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git add .
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $ git commit -am "first commit"
[main 3a96461] first commit
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2020-2021/OS/.txt
create mode 100644 2020-2021/OS/lab02/2
tbkonovalova@dk6n58 ~/laboratory/2020-2021/OS $
```

Рис. 5: Основные команды git

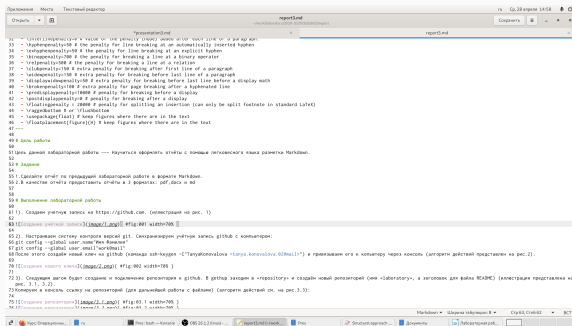
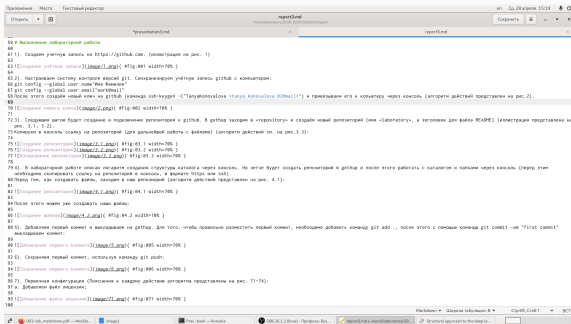


Рис. 6: Записываем и оформляем цель и задания лабораторной работы

Выполнение лабораторной работы в Markdown

- На скриншотах представлен алгоритм выполнения лабораторной работы №1 в Markdown



```
60 в выполнении лабораторной работы
61
62(1). Создаем учетную запись на https://github.com. (инструкция на рис. 1)
63
64[[Создание учетной записи]](image/1.png) { #fig:891 width=70% }
65
66
67(2). Настраиваем систему контроля версий git. Скачиваем утилиту gitHub с компьютером:
68git config --global user.name"Макс Фомин"
69git config --global user.email"mcf@yandex.ru"
70После этого создаем новый ключ на gitHub (команда ssh-keygen -C"mcf@yandex.ru" -t rsa) и привязываем его к компьютеру через консоль (алгоритм действий представлен на рис.2).
71
72[[Создание нового ключа]](image/2.png) { #fig:902 width=70% }
73
74(3). Следующим шагом будет создание и подключение репозитория к gitHub. В gitHub заходим и «repositories» и создаем новый репозиторий (имя «laboratory»), а в поле для файла README (инструкция представлена на рис. 3.1, 3.2).
75Конвертируем в Markdown статью на репозиторий (для дальнейшей работы с файлами) (алгоритм действий см. на рис.3.3):
76
77[[Создание репозитория]](image/3.1.png) { #fig:893.1 width=70% }
78[[Создание репозитория]](image/3.2.png) { #fig:893.2 width=70% }
79[[Создание репозитория]](image/3.3.png) { #fig:893.3 width=70% }
80
81
82(4). В лабораторной работе описаны задания создания структуры каталогов через консоль. Но легче будет создать репозиторий в gitHub и после этого работать с каталогами и файлами через консоль (этот этап необходимо скопировать ссылки на репозиторий в консоль, в формате https://github.com).
83После этого, как создавать файлы, папки и подпапки (алгоритм действий представлен на рис. 4.1):
84
85[[Создание репозитория]](image/4.1.png) { #fig:894.1 width=70% }
86
87
88После этого можно уже создавать новые файлы:
89
90[[Создание файла]](image/4.2.png) { #fig:894.2 width=70% }
91
92(5). Добавление первого коммита и выгрузка на gitHub. Для того, чтобы правильно разместить первый коммит, необходимо добавить команду git add ., после этого в консоли команда git commit -m "first commit" выкладываем коммит:
93
94[[Добавление первого коммита]](image/5.png) { #fig:895 width=70% }
95
96(6). Сохранение первого коммита, используем команду git push:
97
98[[Создание первого коммита]](image/6.png) { #fig:896 width=70% }
99
100(7). Проверка конфигурации (Помощник к каждому действию алгоритм представлен на рис. 71-74):
101
102Добавление файла markdown:
103
104[[Добавление файла markdown]](image/71.png) { #fig:871 width=70% }
105
```

Рис. 7: Выполнение лабораторной работы

Выполнение лабораторной работы в Markdown

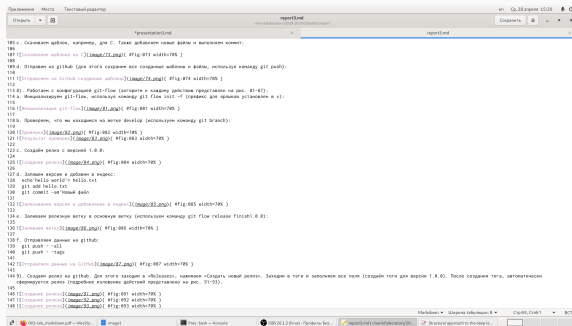
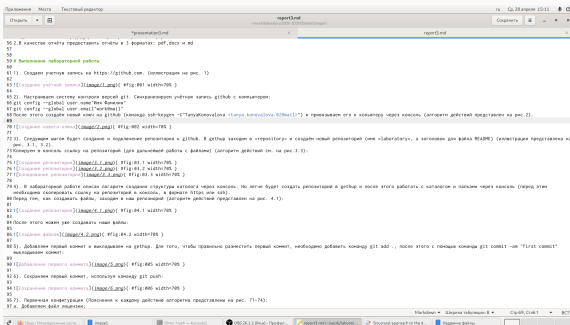


Рис. 8: Выполнение лабораторной работы

Оформляем ход работы в Markdown

- Расписываем полностью алгоритм работы с прошлой лабораторной работы.
- Оформление скриншота в Markdown: обязательно указывать полную ссылку для каждого изображения (пример оформления ссылки представлен на скриншоте)



- Установила VirtualBox, изучила её работу.
- Изучила идеологию и научилась применять средства контроля версий.
- Научилась работать с Markdown-файлами.
- Научилась создавать pdf и docx файлы из файла Markdown (с помощью команды make);

Спасибо за внимание!