Отчет по лабораторной работе 4

Основы интерфейса взаимодействия пользователя с системой Unix на уровне командной строки

Генералов Даниил, НПИ-01-21, 1032212280

Содержание

# 1 Цель работы

Целью данной работы является приобретение практических навыков взаимодействия пользователя с системой посредством командной строки. Для выполнения этого задания делаются несколько стандартных действий для взаимодействия с системой, и в этом отчете задокументированно, какие именно действия я выполняю для этого.

# 2 Теоретическое введение

Операционные системы, основанные на Unix, имеют широкий набор инструментов для взаимодействия с системой посредством командной строки. Это не случайно – когда-то это был единственный способ взаимодействия, и даже сейчас он остается одним из самых удобных.

Фокус на текстовые интерфейсы 20 века, такие как телепринтеры и стеклянные терминалы, приводит к тому, что команды имеют короткие, иногда неочевидные названия – ls для *list directory*, cd для *change directory*, rm для *remove* и так далее. В этой работе мы рассмотрим самые базовые из этих команд и будем объяснять их назначение по ходу работы.

# 3 Выполнение лабораторной работы

Для начала, для соблюдения соглашения о наименовании, нужно создать пользователя с правильным именем. Этот шаг уже сделан на виртуальных машинах, но его еще требуется сделать на моей рабочей машине. Увидеть то, как именно это происходит, можно на рисунке 1

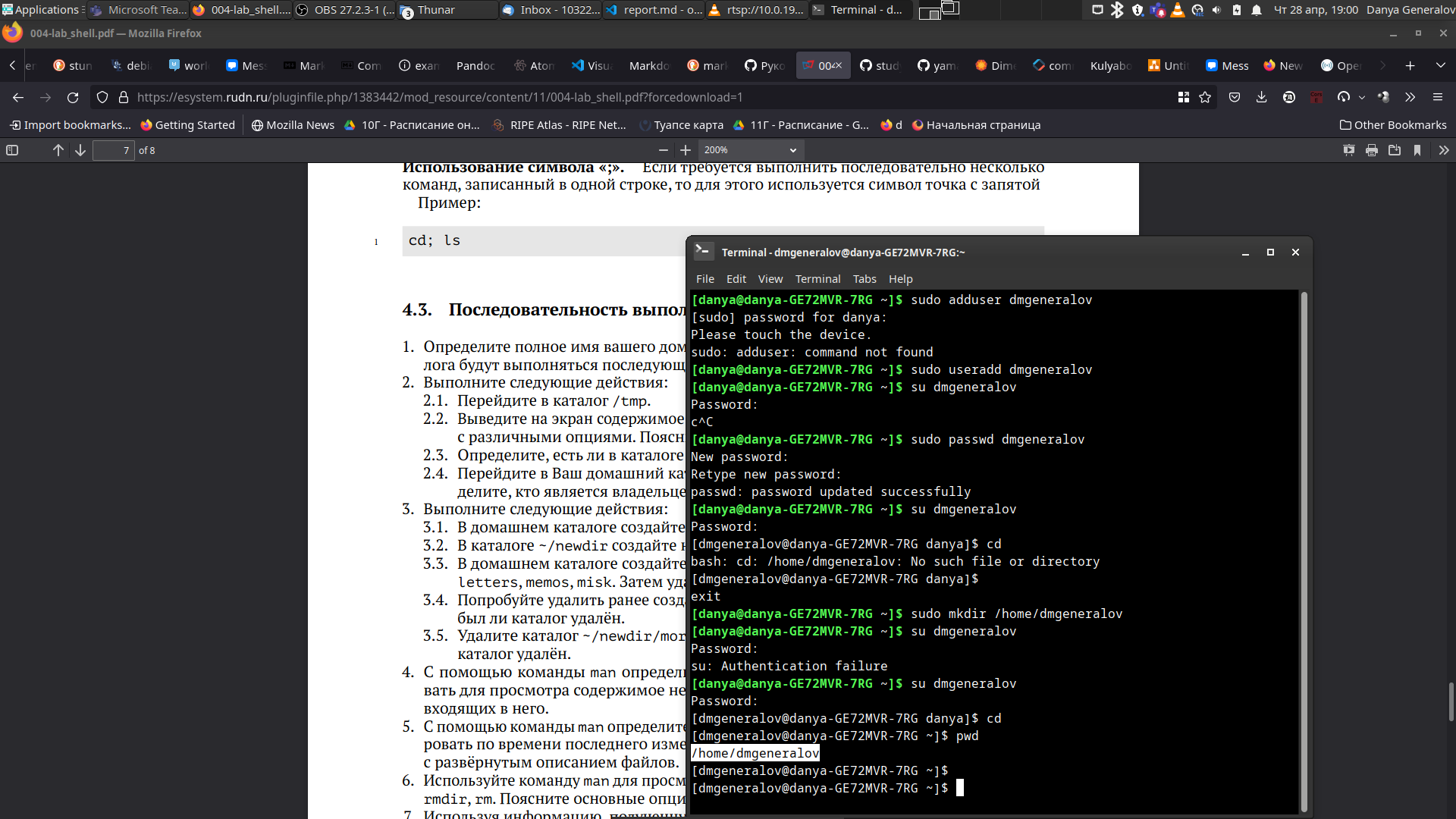


Рис. 1: Создание пользователя для работы

После этого можно начать выполнение задания, и первым пунктом является переход в директорию /tmp и просмотр её содержимого. Чтобы перейти в эту директорию, нужно выполнить cd /tmp, а чтобы просмотреть файлы – ls, как на рисунке 2. Можно также посмотреть дополнительную информацию о файлах и увидеть специальные записи . и .., которые есть в каждой директории и указывают на её саму и её родителя, соответственно. Чтобы посмотреть эти записи, можно использовать флаг -a, а чтобы посмотреть дополнительную информацию – -l. Эти флаги можно также использовать вместе, как видно на рисунке 3.

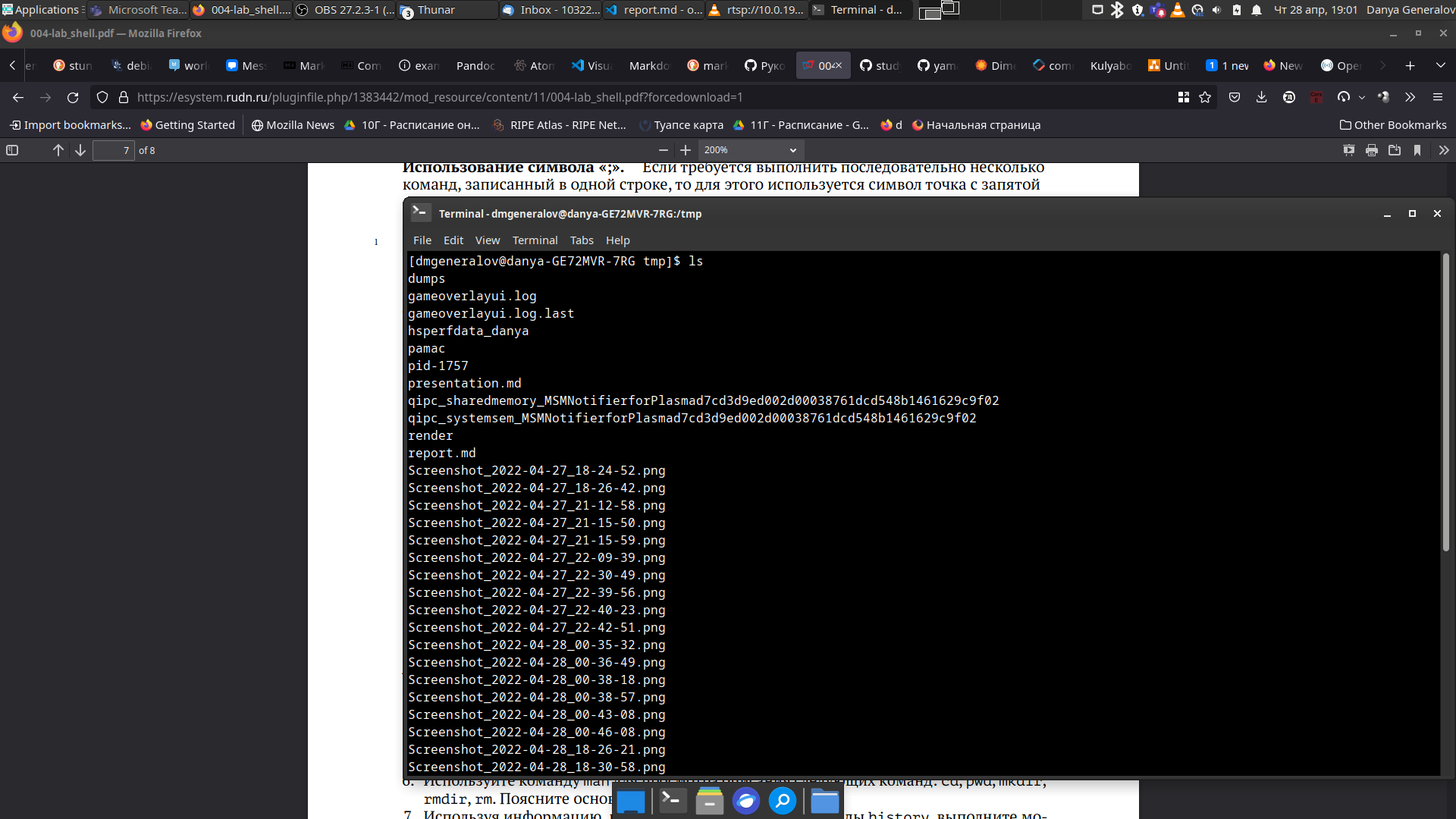


Рис. 2: Переход в директорию /tmp и просмотр содержимого

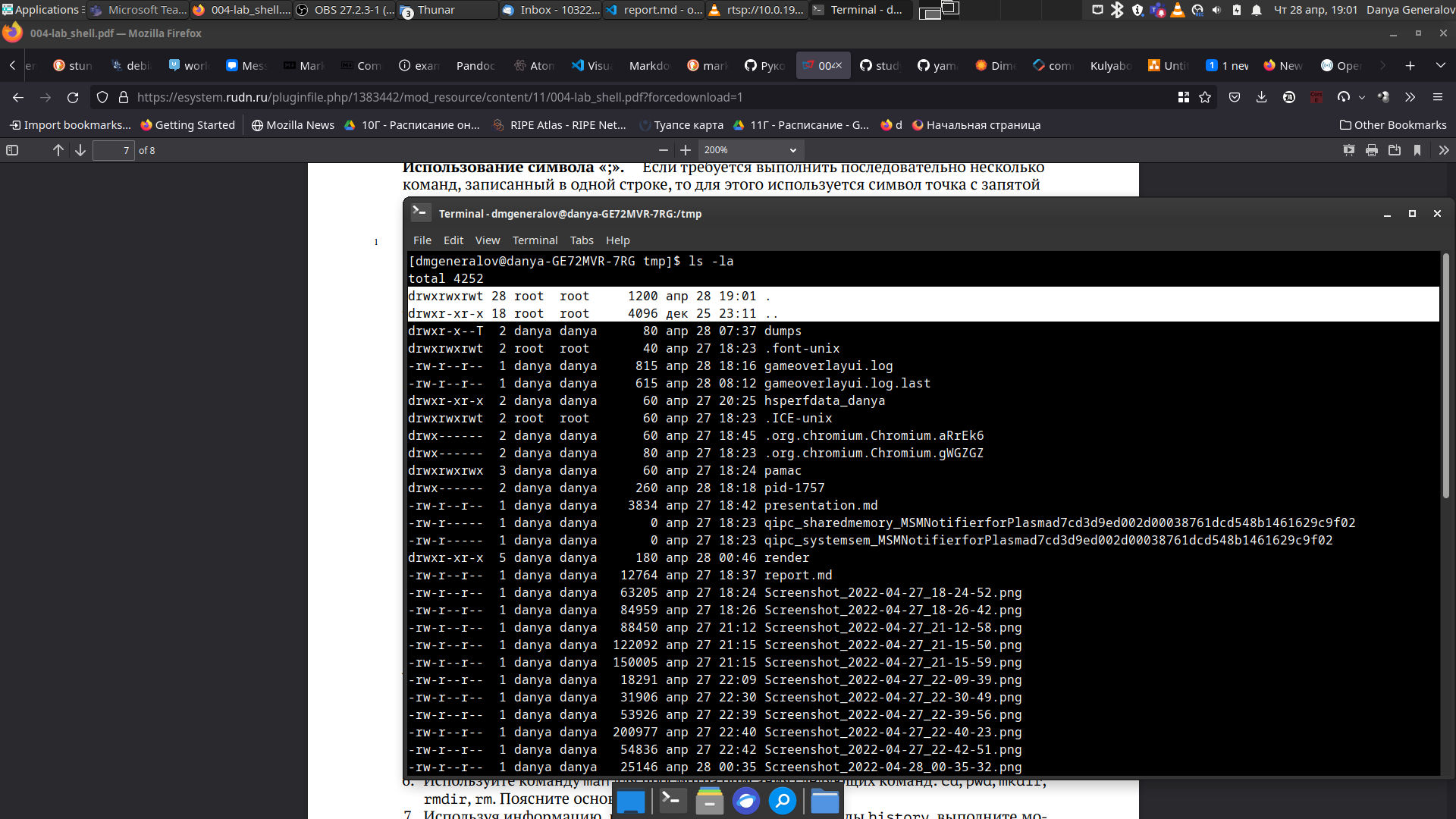


Рис. 3: Просмотр содержимого с дополнительной информацией

Следующий пункт – проверить наличие директории под названием /var/spool/cron, что можно сделать, посмотрев на список файлов в /var/spool и проверив, что cron присутствует в нем и является директорией – это действительно так, что можно увидеть на рисунке 4.

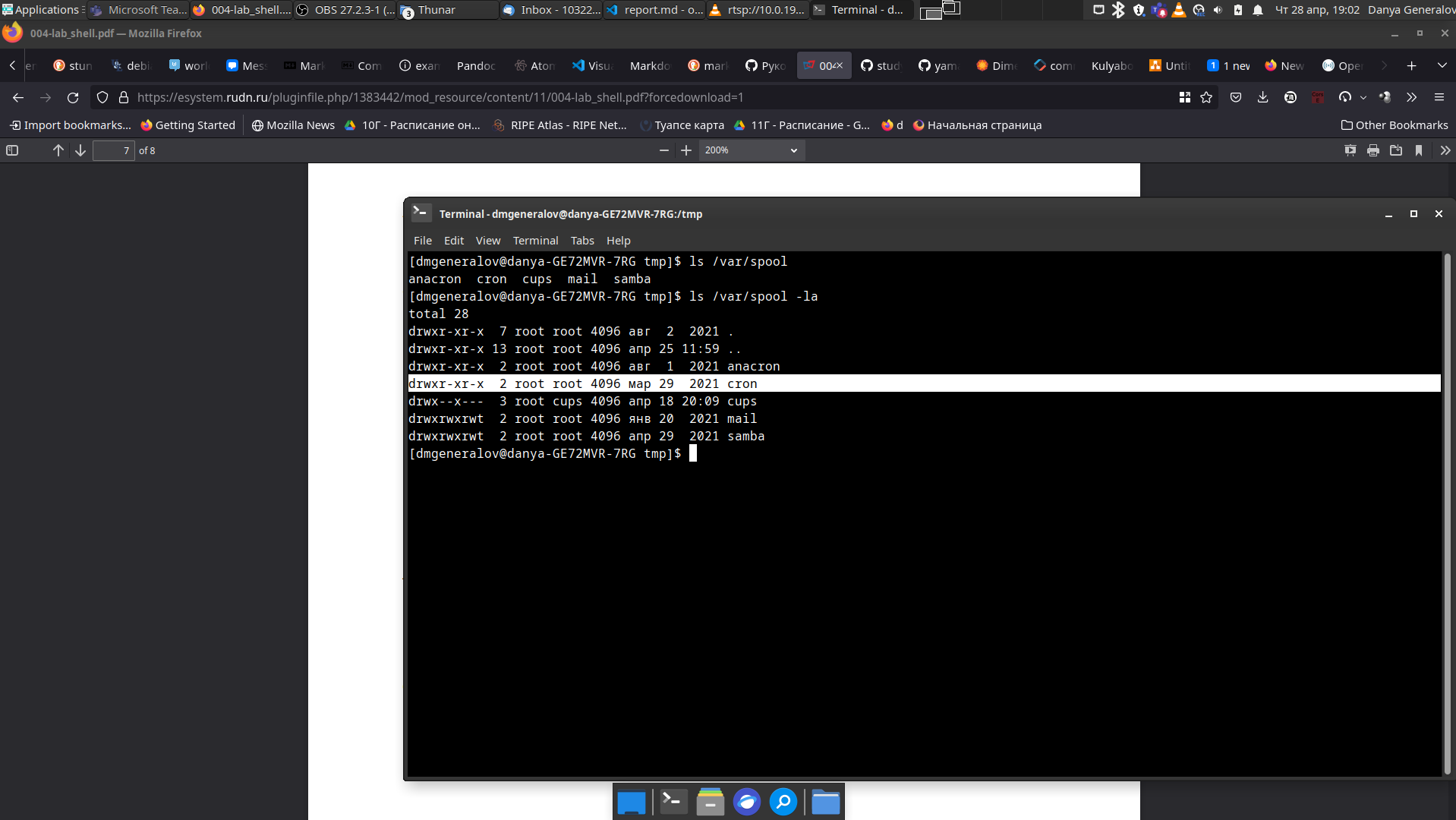


Рис. 4: Проверка наличия директории /var/spool/cron

После этого предлагается перейти в домашнюю директорию и проверить владельца и группу файлов, которые находятся там. Из-за того, что это новый пользователь, то его домашняя директория пуста. Более того, она принадлежит пользователю root, из-за чего нельзя добавлять туда файлы, но это не проблема – с помощью одной команды chown можно поменять владельца этой директории, и после этого новые файлы принадлежат нашему пользователю. Все это видно на рисунке 5.

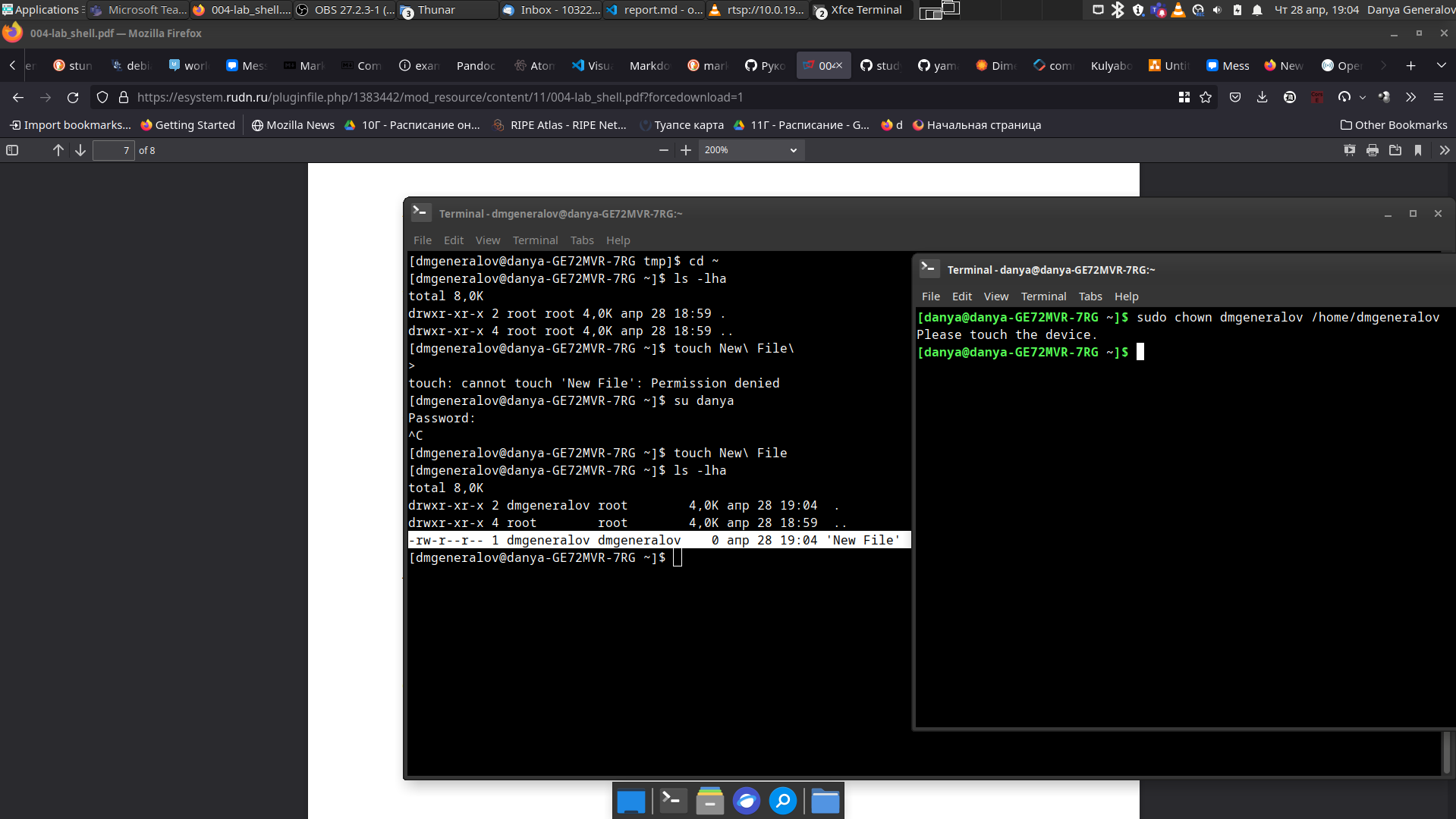


Рис. 5: Проверка владельца файлов в домашней директории

Теперь нужно создать папку и внутри нее создать подпапку. Это можно сделать двумя командами mkdir. После этого нужно создать три других папки, и здесь я ошибся: я написал mkdir {letters, memos, misk}, ожидая, что будет использован синтаксис раскрытия перечислений. Однако, поскольку в этой команде есть пробелы, то эта строка воспринимается иначе: создаются три папки, имеющих название {letters, и memos, и misk} и это не то, что нужно. Поэтому я удаляю их с помощью rmdir, и переписываю эту команду без пробелов, что на этот раз правильно интерпретируется. Наконец, я делаю rmdir, чтобы удалить эти новые папки. Это можно увидеть на рисунке 6.

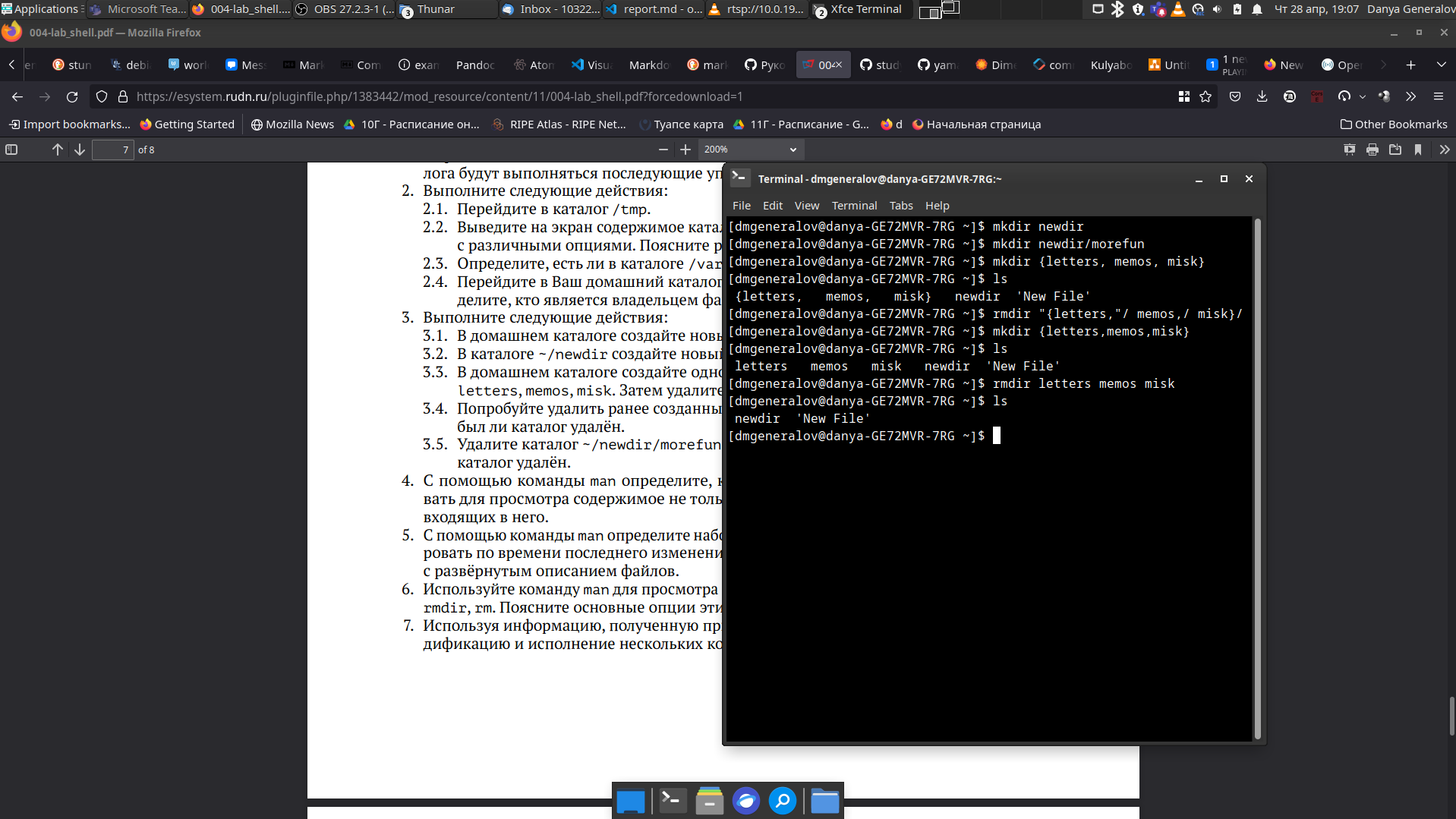


Рис. 6: Создание и удаление папок в домашней директории

Для удаления здесь мы использовали rmdir, но также можно использовать rm. Для того, чтобы это работало, необходимо указывать флаг -r, который диктует рекурсивное исполнение: для того, чтобы удалить папку, нужно сначала удалить все файлы внутри папки, и некоторые из этих файлов могут сами быть папками. Можно увидеть, что с этим флагом работает удаление директорий, на рисунке 7.

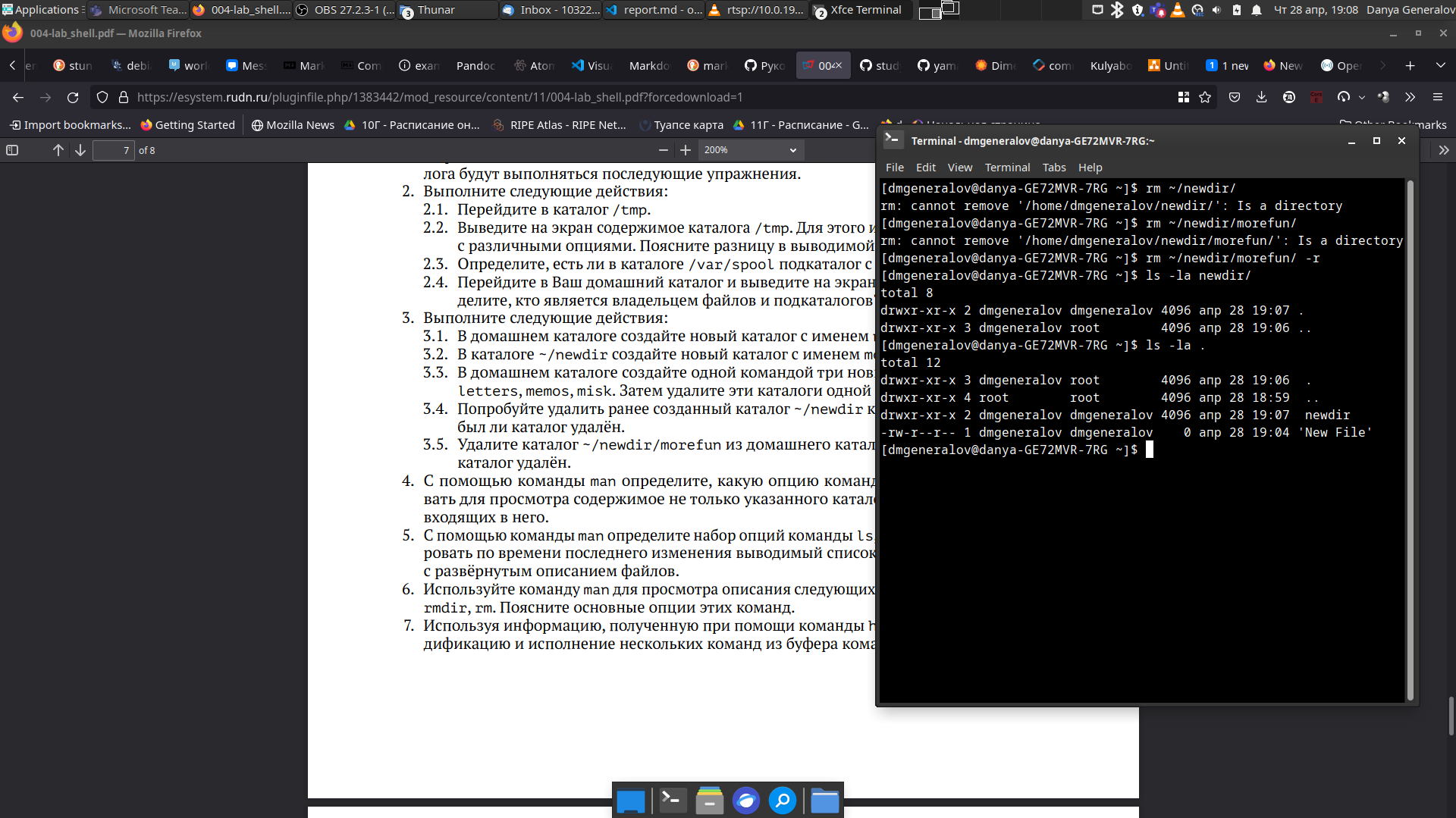


Рис. 7: Удаление папок с помощью rm

Следующий важный шаг – научиться, как читать инструкцию. В Linux для большинства программ есть так называемые man-страницы, которые рассказывают, среди прочего, какие опции есть у программ и как ими пользоваться. Так, можно прочитать, что у ls есть ключ -R, который позволяет просмотреть директории рекурсивно. Это написано в документе, который можно открыть, введя команду man ls, как показано на рисунке 8.

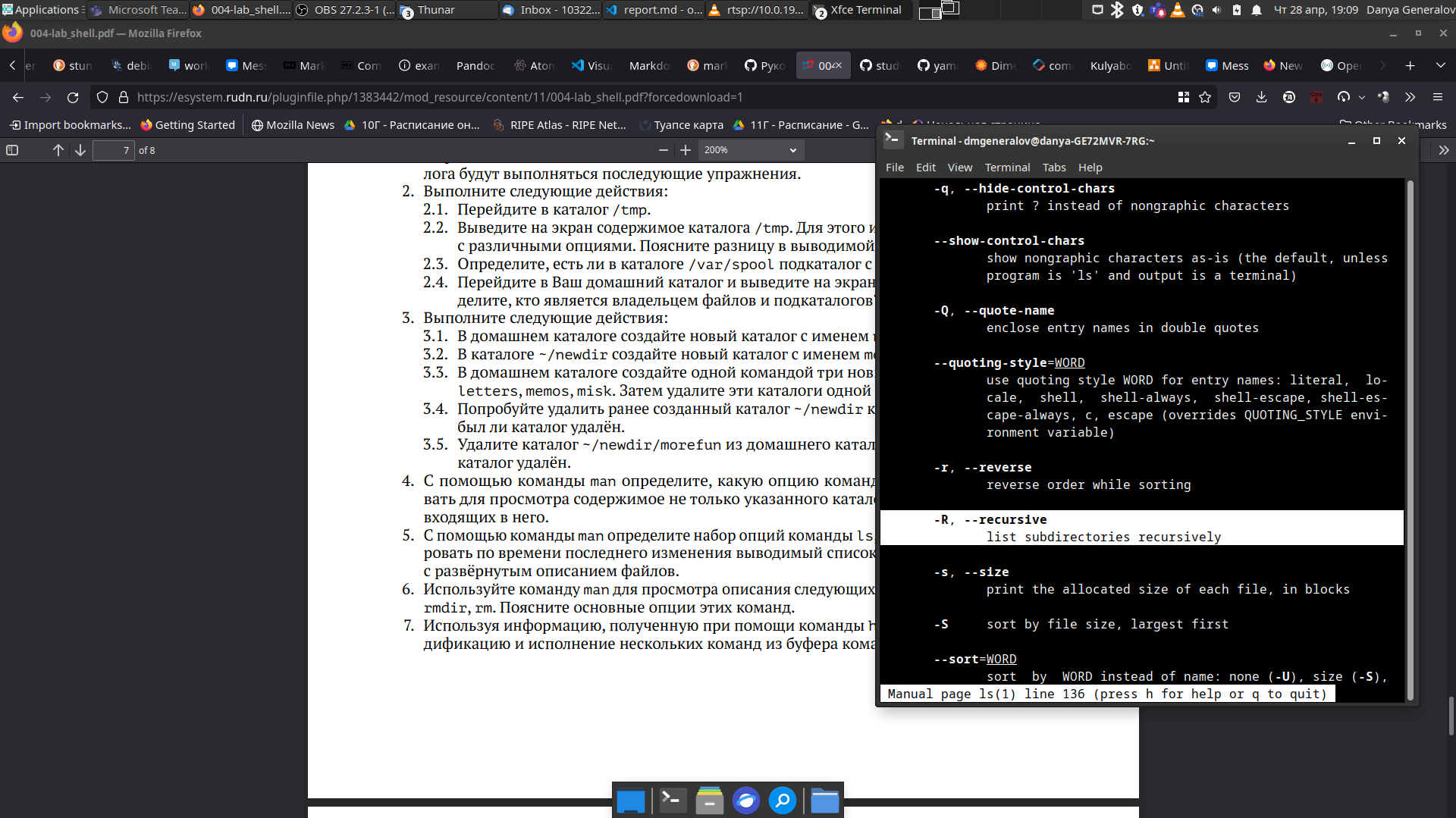


Рис. 8: man-страница для ls

В этом документе можно прочитать и про более сложные ключи, которые принимают параметры. Так, например, можно узнать, что можно посмотреть на расширенную информацию о файлах и отсортировать их по дате изменения, используя ключи -l, --sort=time и --time=ctime, как показано на рисунке 9. После этого можно выполнить команду с этими ключами и убедиться, что она делает именно то, что обещано, на рисунке 10.

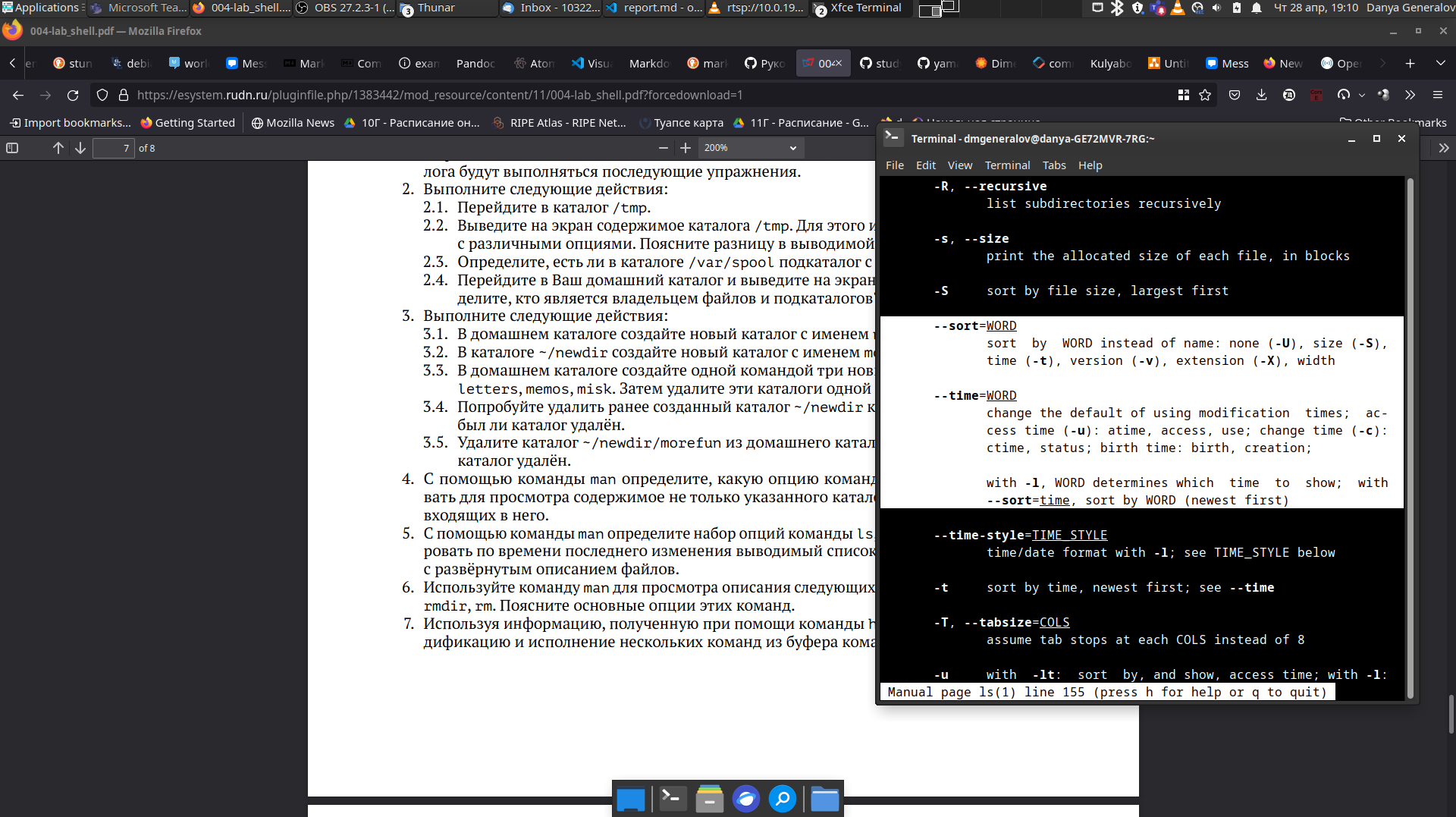


Рис. 9: Инструкции по сортировке

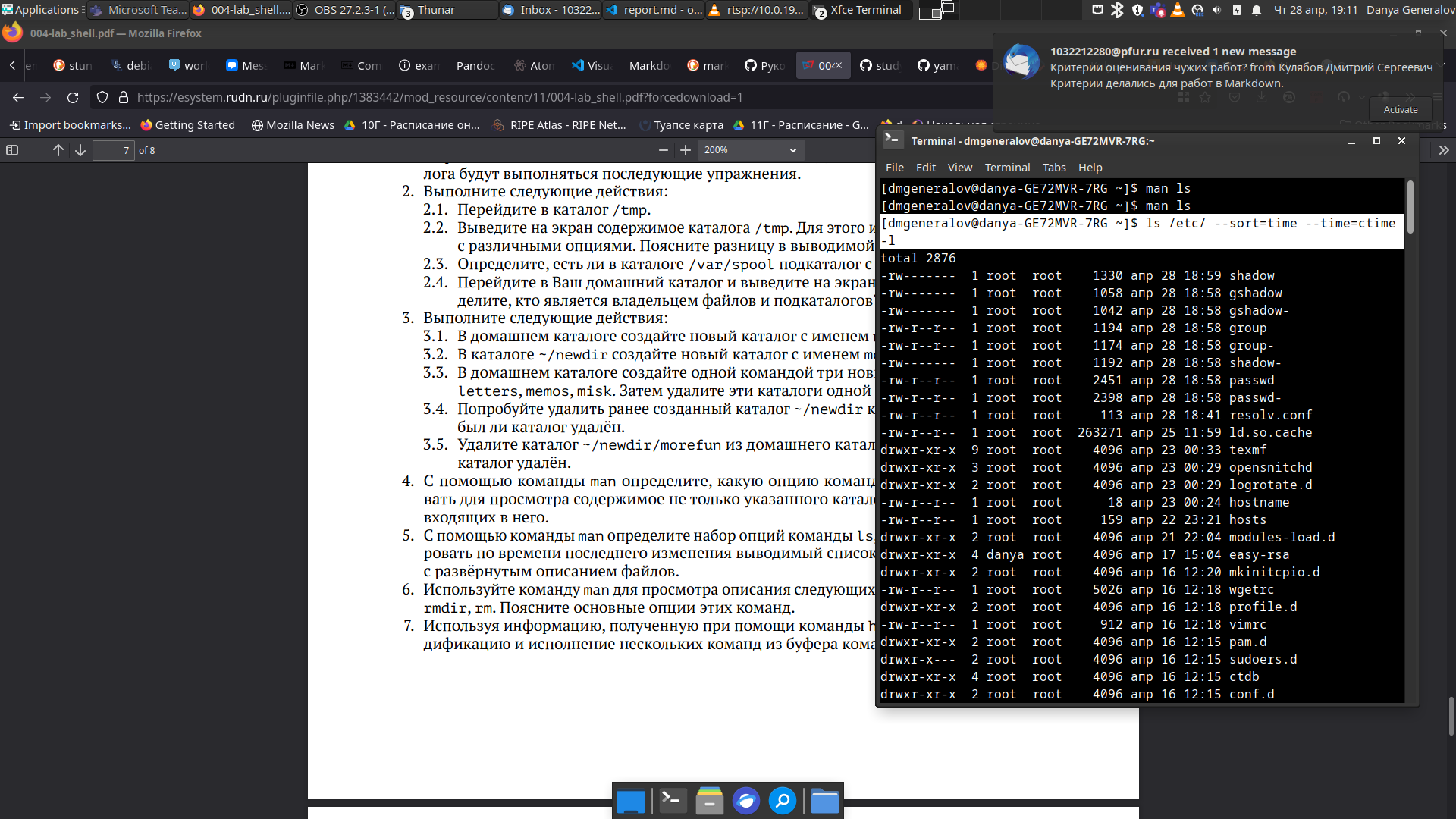


Рис. 10: Просмотр отсортированного вывода

С помощью man можно узнать много интересного о командах вроде cd, pwd, mkdir, rmdir и rm – хотя стоит заметить, что cd является встроенной в интерпретатор команд (в моем случае это bash), поэтому документация о этой команде не совсем относится к тому, как именно она реализована – и документ даже говорит об этом, как можно увидеть на рисунке 11.

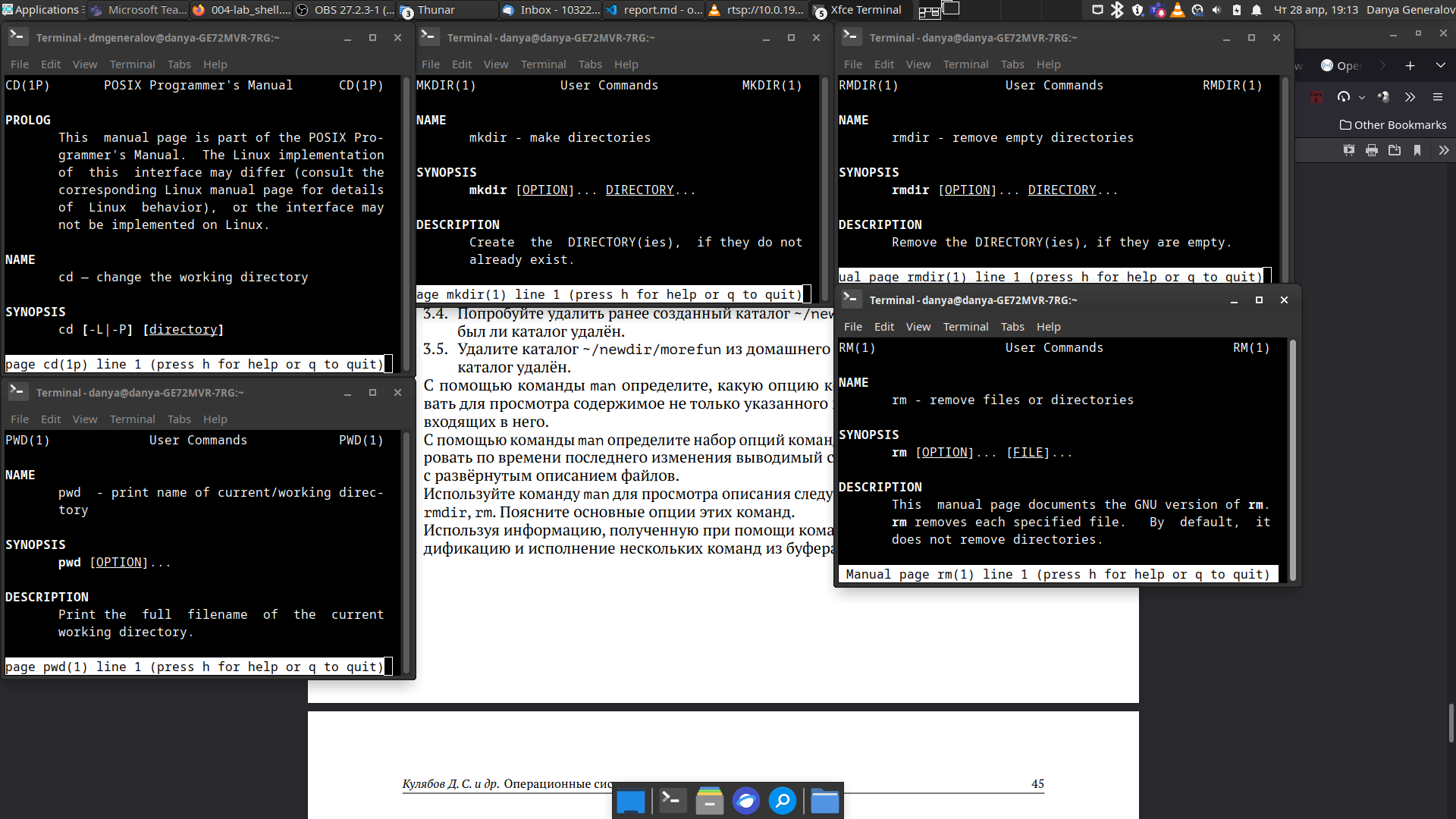


Рис. 11: Документация по разным командам

Помимо этого, можно узнать, что:

* cd имеет параметры -L и -P, которые определяют, как именно следует воспринимать особую запись .. в пути – как настоящую символическую ссылку, или только как команду перейти на один уровень наверх;
* pwd имеет параметры -L и -P, которые говорят, следует ли показывать символические ссылки в отображаемом пути или стоит их разрешить в канонический путь;
* mkdir имеет параметр -m, который позволяет указать права доступа к создаваемой директории; параметр -p, который позволяет создавать все необходимые директории в пути, если они не существуют; и параметры -Z и --context, которые управляют контекстом безопасности SELinux, в котором будут находиться созданные директории;
* rmdir имеет параметр -p, который удалит не только саму директорию, но и всех ее пустых родителей, и параметр -v, который позволяет выводить информацию о том, какие директории были удалены;
* rm имеет параметр -r, который позволяет программе работать рекурсивно; параметр -f, который позволяет удалять файлы и директории без предупреждения; и параметр -i, который, наоборот, спрашивает разрешения прежде чем удалить каждый файл.

Наконец, используя встроенную команду history, можно посмотреть историю выполнения команд, и из этой истории можно достать и изменить команды, чтобы выполнить одно и то же действие несколько раз, как показано на рисунке 12.

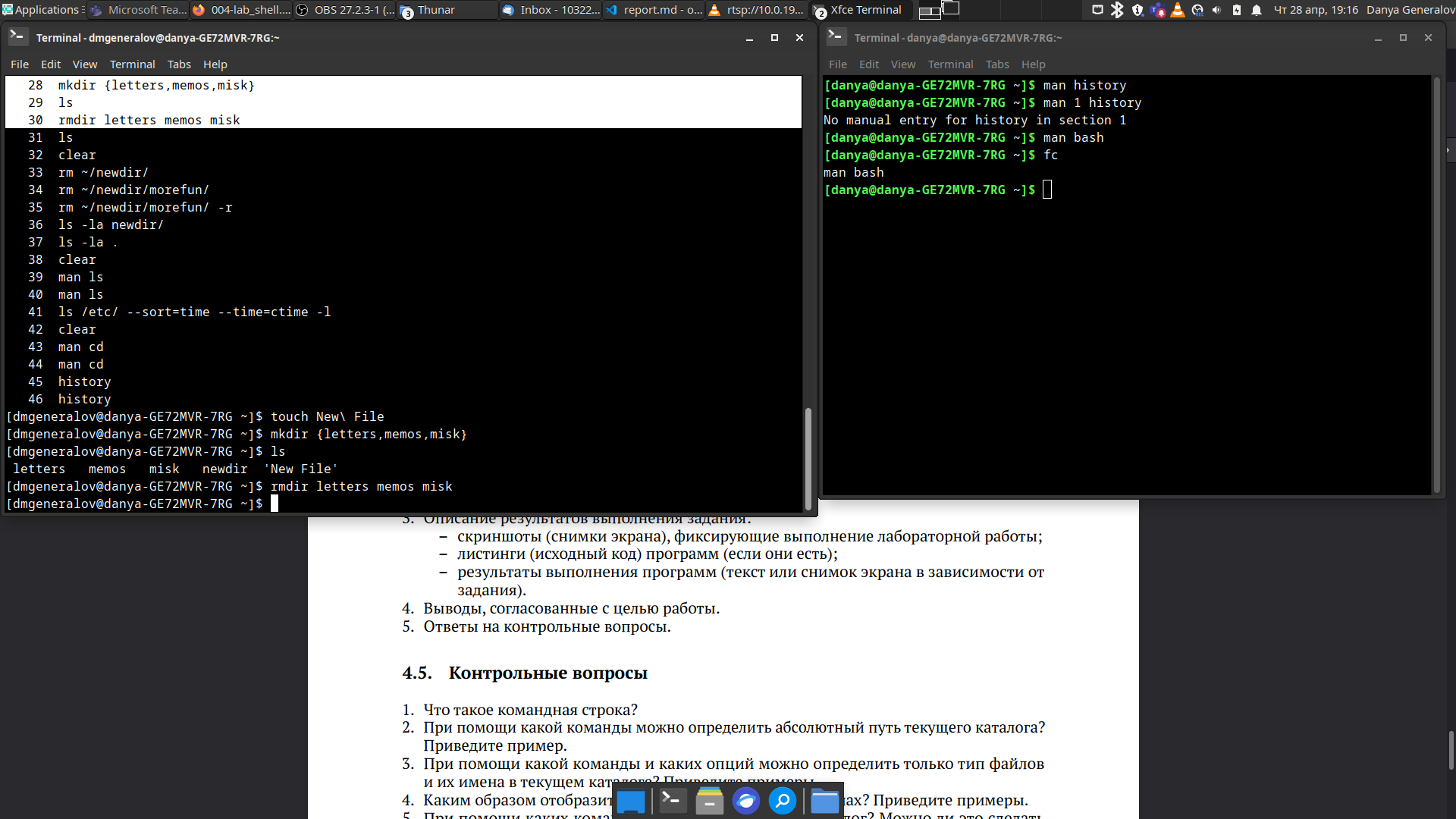


Рис. 12: История команд

# 4 Выводы

В этой работе мы рассмотрели основы работы с командной строкой Linux, рассмотрев команды перехода между папками, создания и удаления папок, и просмотра документации. Используя эти навыки, можно работать с компьютером быстрее и эффективнее, чем с графическим интерфейсом, и эта простота и быстрота остается причиной, почему текстовый интерфейс до сих пор доступен в современных компьютерах.