

Инструкция по выполнению лабораторных работ (общая)

Соколов А.П.*; Першин А.Ю.†
кафедра «Системы автоматизированного проектирования»
МГТУ им. Н.Э. Баумана

2021-09-18

1 Введение

Инструкция разработана для ознакомления с принципами реализации лабораторных работ, выполняемых в рамках курсов: «Вычислительная математика» (бакалавриат), «Методы математического моделирования сложных процессов и систем» (магистратура), «Управление проектированием информационных систем» (магистратура), проводимых на кафедре «Системы автоматизированного проектирования» МГТУ им. Н.Э.Баумана.

Управление процессом реализации лабораторных работ включает необходимость решения следующих задач:

№	Задача	Метод реализации
1	Контроль за выполнением	только по срокам
2	Представление к проверке	создание Merge Request в GitLab
3	Проверка подготовленных отчета и исходных кодов (защита)	peer-review
4	Хранение отчетов и реализованных исходных кодов	GitLab, Nextcloud

Хранение и организация отчетов о лабораторных работах и им соответствующих исходных кодов осуществляется с помощью системы контроля версий **git** и веб-приложения **GitLab**: <https://sa2systems.ru:88>. Доступ организован централизованным образом с использованием единой учетной записи студента, зарегистрированной в базе данных сервиса **OpenLDAP** администратором (учетные данные следует уточнять у сотрудников кафедры).

2 Общие требования к знаниям для выполнения

Для выполнения лабораторных работ обучающийся должен обладать знаниями:

- знать принципы работы системы контроля версий Git;

*доцент РК-6, к.ф.-м.н., email: alsokolo@bmstu.ru

†ассистент РК-6, email: apershin@bmstu.ru

- владеть навыками использования системы контроля версий Git (уметь применять команды).

3 Общие принципы выполнения заданий

Разработку документов и исходных кодов программ, реализующих те или иные программные модули, функции, численные методы, следует осуществлять с использованием системы контроля версий **Git**.

Все лабораторные работы следует выполнять с использованием удалённого репозитория, соответствующего дисциплине, в рамках которой выполняются лабораторные. Адреса репозитория имеют единый формат

<https://sa2systems.ru:88/@cpx@/cpxsln@.git>, где @cpx@=edu, @cpxsln@ – идентификатор репозитория, отличный для каждой дисциплины:

- дисциплина «Вычислительная математика» (бакалавриат), @cpxsln@=educmm:
<https://sa2systems.ru:88/edu/educmm.git>;
- дисциплина «Методы математического моделирования сложных процессов и систем» (магистратура), @cpxsln@=edumms:
<https://sa2systems.ru:88/edu/edumms.git>;
- дисциплина «Управление проектированием информационных систем» (магистратура), @cpxsln@=edupmi:
<https://sa2systems.ru:88/edu/edupmi.git>;
- дисциплина «Модели и методы анализа проектных решений» (бакалавриат), @cpxsln@=edumma: <https://sa2systems.ru:88/edu/edumma.git>;

Перед выполнением лабораторных работ по определённой дисциплине соответствующий репозиторий следует клонировать (выполняется один раз).

Обновление загруженных материалов рекомендуется осуществлять с применением соответствующих команд системы контроля версий git

Для отслеживания изменений, управления процессом разработки, а также для сдачи результатов выполнения следует использовать систему GitLab, развернутую по адресу: <https://sa2systems.ru:88>. А именно: **после размещения финальных версий документов следует осуществить операцию merge request, что приведёт к уведомлению преподавателя(ей) о необходимости осуществить проверку.** Доступ к GitLab осуществляется с использованием общей единой учетной записи обучающегося: **логин: rk6_student пароль: 2afeu33f.**

Студенческий доступ к прочим сервисам, включая облачный сервис кафедры <https://archrk6.bmstu.ru>, также осуществляется с использованием этой учетной записи.

Для каждой реализуемой лабораторной работы должны быть разработаны:

1. исходные коды программы, реализующей решение поставленной задачи;
2. отчет о проведенной лабораторной работе согласно шаблону, размещенному в облачном хранилище кафедры (табл. ??).

Шаблон в формате .doc	https://archrk6.bmstu.ru/index.php/f/85083
Шаблон в формате L ^A T _E X	https://archrk6.bmstu.ru/index.php/f/844352

Таблица 1: Доступные шаблоны отчетов о ЛР

3.1 Требования к именам. Форматы идентификаторов

Листинг 1: Формат идентификатора обучающегося

```
1 @student_sid@ = @group@_@surname@@fname@
```

где @group@ – идентификатор учебной группы (английскими буквами, регистр нижний, например: rk6_62b, rk6_12m); @fname@ – первые буквы имени и отчества (английскими буквами, регистр нижний); @surname@ – фамилия (английскими буквами, регистр нижний).

Например: rk6_62b_ivanovii, rk6_12m_petrovpa.

Листинг 2: Формат идентификатора **лабораторной работы** (совпадает с **именем каталога** для размещения материалов лабораторной)

```
1 @labid@ = @lab.dirid@ = lab@i@
```

где @i@ – порядковый номер лабораторной работы, @lab.dirid@ – формат имени каталога, для размещения материалов одной лабораторной работы.

Например: lab1, lab2, lab3.

Листинг 3: Формат имени **ветки** разработки обучающегося (**НЕ** совпадает с именем **запроса на слияние**)

```
1 @student_branch_sid@ = @year@_@student_sid@
```

где @year@ – текущий год; @student_sid@ – идентификатор обучающегося, формат которого определён отдельно.

Например: 2020_rk6_62b_ivanovii.

Листинг 4: Формат имени **запроса на слияние**)

```
1 @merge_request_sid@ = @student_branch_sid@[_@labid@]
```

где материал, заключенный в квадратных скобках не обязателен.

Например: 2020_rk6_62b_ivanovii_lab1, 2021_rk6_64b_petrovpp.

Листинг 5: Формат имени **файла отчета** о проведённой лабораторной обучающегося

```
1 @lab_report_name@ =
  @cpxsln@_@tml@_@year@_@student_sid@_@labid@.@ext@
2 @cpxsln@ = (educmm|edumms|edupmi|edumma|...)
3 @ext@ = (doc|docx|pdf)
4 @tml@ = lab
```

где @student_sid@ – идентификатор обучающегося; @labid@ – идентификатор лабораторной работы; @ext@ – расширение файла; @tml@ – идентификатор шаблона лабораторной работы.

Например: edumms_lab_2019_rk6_62b_ivanovii_lab1.pdf

3.2 Структура каталогов для размещения результатов выполнения

В процессе выполнения лабораторных работ обучающимся формируются различные документы среди которых: исходные коды разработанных программ, отчет о выполненной лабораторной работе. Все документы следует размещать в Git-репозитории соответствующей дисциплины с использованием единой структуры каталогов (рисунок ??).

Конкретные имена представленных файлов в подкаталогах (рисунок ??) не формализованы и могут быть названы произвольным образом.

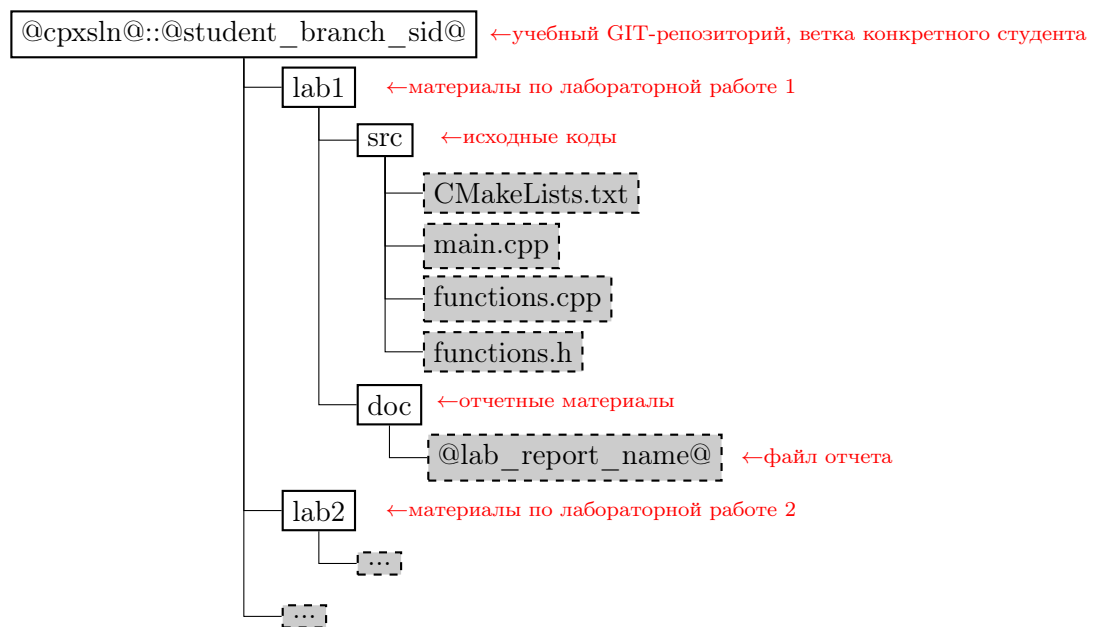


Рис. 1: Структура файловой системы при выполнении лабораторных работ

4 Принципы выполнения лабораторной работы

Следующий алгоритм предполагает, что удаленный Git-репозиторий был клонирован и размещен локально.

Алгоритм 1 Выполнение лабораторной работы (осуществляет студент)

- 1: Создать свою Git-ветку разработки и присвоить ей имя в формате согласно листингу ??¹.
 - 2: Разместить (обновить после отладки) материалы лабораторной работы в отдельном каталоге с использованием структуры каталогов согласно рисунку ??².
 - 3: Уведомить преподавателя о готовности лабораторной работы к сдаче, создав запрос на слияние **merge request** из своей ветки в **master** ветку с использованием системы GitLab и обязательно создать комментарий о необходимости провести проверку.
-

Наименование запроса на слияние **merge request** следует давать в формате согласно листингу ??. Выполнение лабораторных работ возможно не по порядку, все выполняемые лабораторные работы могут быть привязаны к одному запросу merge request.

Рекомендуется при каждой операции commit указывать автора, который осуществляет операцию commit, а также дату операции.

5 Алгоритм проведения проверки лабораторной работы

Алгоритм проведения проверки поэтапный. После того как студент загрузил **первую** версию отчетных материалов, включая: отчет и исходные коды созданных программ, а также сформировал “merge request”, преподаватель начинает проверку.

Алгоритм 2 Проверка лабораторной работы (осуществляет преподаватель)

- 1: Преподаватель изучает представленный на проверку материал и формирует *рецензию* на *первую* версию отчета, заполняя стандартную форму рецензии (<https://archrk6.bmstu.ru/index.php/f/788873>).
 - 2: Для осуществления проверки *заполняется интерактивная форма* рецензии³, путём ответов на ряд вопросов в рецензии автоматически выставляется оценка (по 100 балльной шкале). В соответствие полученному баллу определяется оценка по 5-и балльной шкале. В зависимости от качества реализации, оценок, полученных по множеству критериев, срока сдачи и т.д. за текущую лабораторную работу выставляется финальная оценка (зачет/незачет) и число баллов, которые будут учитываться при сдаче экзамена.
 - 3: Данная рецензия сохраняется вместе с первой версией отчета путём её загрузки в репозиторий с использованием системы контроля версий.
-

Если студент набрал более 60-и баллов, то лабораторная работа считается защищённой. Если студента не устраивает выставленная оценка более 60-и баллов, то он вправе внести корректировки в предоставленные материалы и попросить провести проверку повторно (на усмотрение преподавателя и при наличии соответствующей возможности преподаватель может осуществить повторную проверку).

Если студент не набрал 60-и баллов, то работу следует доработать.

В качестве проверяющих могут выступать магистры и аспиранты.

Детальные требования к отчетным материалам по лабораторным работам представлены в интерактивной форме рецензии и могут корректироваться от года к году.

6 Установка Git системы контроля версий и Git-клиента

Для работы с удалённым Git репозиторием на локальной машине должно быть установлено программное обеспечение:

1. Git (система контроля версий, серверная часть)⁴: <https://git-scm.com/>.
2. Git клиент (для работающих под операционной системой Windows): <https://tortoisegit.org>.

7 Частые ошибки

Частая ошибка 1 Создание файлов с именем “aих”, в том числе файлов, имена которых включают эти три буквы так, чтобы сразу после них следовала точка.

Комментарий

В операционных системах семейства Windows запрещается создание файлов с такими именами, поэтому при размещении такого файла в git-репозитории с использованием, положим, операционной системы семейства Linux, такие данные будет невозможно клонировать/обновить в Windows.

Частая ошибка 2 Создание очень длинных имён файлов или имён каталогов.

Комментарий

В операционных системах семейства Windows есть ограничение на создание файлов, полный путь которых содержит более 260 символов. В то же время файловая система NTFS разрешает создавать такие файлы и каталоги, **но только программно**. В операционных системах семейства Linux и Mac OS таких ограничений нет, как и в используемых ими файловых системах. Таким образом, не следует создавать длинные имена файлов и каталогов и загружать их в git-репозитории. Такие файлы и каталоги будут удаляться без проведения проверки.

⁴При работе в операционных системах Linux установка осуществляется командой: `$yum install git` (CentOS, Red Hat), `$apt-get install git` (Debian, Ubuntu) и т.д. Одновременно устанавливается git-клиент, работающий из командной строки.