**Российский университит дружбы народов**

**Факультет физико-математических и естественных наук**

**Лабораторная работа № 12. Программирование в командном процессоре ОС UNIX. Расширенное программирование**

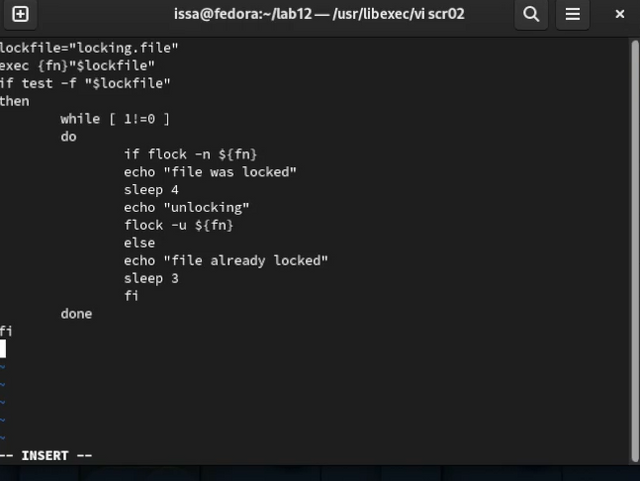
* Имя : исса гадир
* Студенческий билет : 1032218267
* Группа : нфибд-01-21

**Цель работы:**

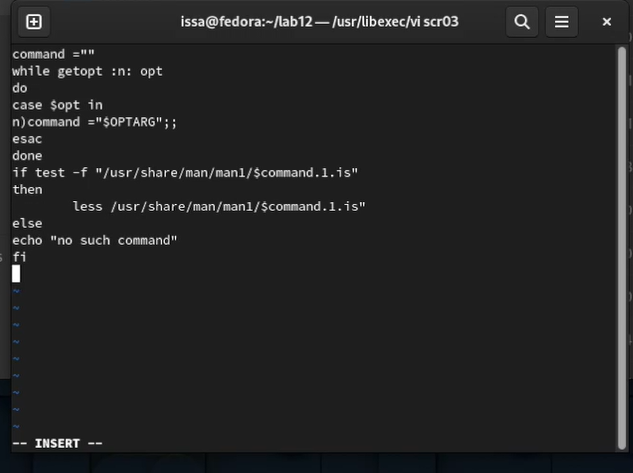
*Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.*

**выполнения работы**

1. Написать командный файл, реализующий упрощённый механизм семафоров. Ко- мандный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (> /dev/tty#, где # — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

**[](https://ibb.co/kgtGPrf)**

1. Реализовать команду man с помощью командного файла. Изучите содержимое ката- лога /usr/share/man/man1. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой less сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1.

**[](https://imgbb.com/)**

1. Используя встроенную переменную $RANDOM, напишите командный файл, генерирую- щий случайную последовательность букв латинского алфавита. Учтите, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

**[](https://imgbb.com/)**

**Контрольные вопросы:**

**1.Найдите синтаксическую ошибку в следующей строке : while [ $ 1 ! – " exit " ]**

\*Строковый тип не может быть использован \*

**2.Как объединить ( конкатенация ) несколько строк в одну ?** *str3 = " \(str1\) str2 "*

**3.Найдите информацию об утилите seq . Какими иными способами можно реализовать её функционал при программировании на bash ?**

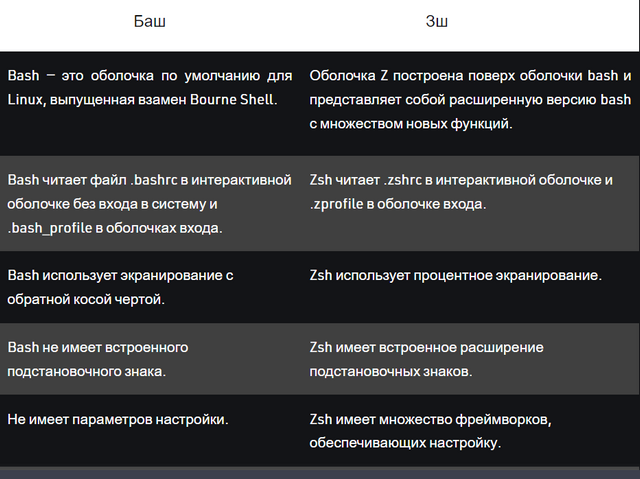
*используется для идентификации и аннотирования сайта вставки мутатора ,используемого при создании ресурсов BonnMu,Может быть реализован с помощью конвейера*

**4. Какой результат даст вычисление выражения $ ((10/3)) ?** *3*

**5. Укажите кратко основные отличия командной оболочки zsh от bash.**

*повторяющееся определение*

**6. Укажите кратко основные отличия командной оболочки zsh от bash.**

**[](https://ibb.co/jGmfWy7)**

**7. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки?**

*Оболочка — это язык программирования высокого уровня, что означает, что вам не нужно беспокоиться о сложных задачах, таких как управление памятью. Это облегчает изучение языка системного программирования, такого как C или C++. Программы оболочки, как правило, пишутся быстрее, чем соответствующие программы на языке C, и их часто легче отлаживать. Однако программы на C почти всегда работают быстрее и эффективнее. Таким образом, сценарии оболочки и программирование на C используются для очень разных задач. Для быстрого написания относительно коротких инструментов оболочка является гораздо лучшим выбором, но для крупных проектов системного программирования явно лучше подходит C.*