

# Презентация к лабораторной работе №14

Операционные системы

---

Бережной И. А., НКАбд-01-23

Российский университет дружбы народов, Москва, Россия

Для программиста необходимо уметь писать рабочие скрипты, чтобы оптимизировать процесс работы

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

1. Написать командный файл, реализующий упрощённый механизм семафоров
2. Реализовать команду `map` с помощью командного файла
3. Написать командный файл, генерирующий случайную последовательность букв латинского алфавита

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:

- оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;

- С-оболочка (или `cs`h) — надстройка на оболочке Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или `k`sh) — напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation).

Создадим файл в терминале и присвоим ему право на исполнение. Напишем скрипт, который в течение некоторого времени дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использует его в течение некоторого времени, также выдавая информацию о том, что ресурс используется соответствующим командным файлом. Запустим скрипт и проверим результат.

Создадим второй командный файл и также дадим ему право на исполнение. Реализуем команду `map` скриптом, который будет получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки. Запустим файл и получим такой результат.

Теперь напишем командный файл, генерирующий случайную последовательность букв латинского алфавита. Запустим файл.

Теперь напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N, при этом количество файлов будет передаваться в аргументы командной строки. Проверим, работает ли скрипт.

И, наконец, напишем командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Запустим скрипт через командную строку.



Мы изучили основы программирования в оболочке ОС UNIX и научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.