Расширенное программирование в UNIX

Лабораторная работа 14

Смольняков Данил Евгеньевич

Архитектура компьютеров и ОС

# Цель работы

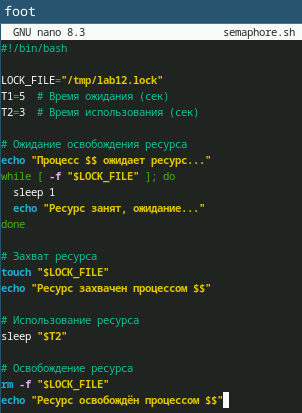
## Цель

* Изучить механизмы межпроцессного взаимодействия и управления ресурсами в UNIX
* Освоить работу с системными справочными страницами и генерацию случайных данных

# Скриншоты и описание действий

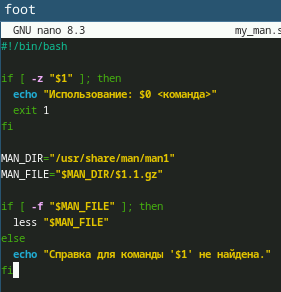
## 1. Семафоры

* **Действие:** Процессы по очереди захватывают/освобождают ресурс
* **Скриншот:**

****  
Семафоры

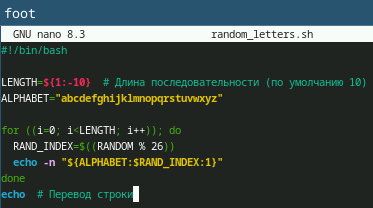
## 2. Кастомный man

* **Действие:** Показывает справку или ошибку, если команды нет
* **Скриншот:**

****  
Кастомный man

## 3. Случайные буквы

* **Действие:** Генерирует строку из 15 случайных букв
* **Скриншот:**

****  
Случайные буквы

# Контрольные вопросы

## Ответы

1.Ошибка: отсутствие пробелов внутри [ ] → while [ “$1” != “exit” ]

2.Конкатенация: result=“{str2}”

3.seq альтернативы: {1..N} или for ((i=1; i<=N; i++))

4.$((10/3)): 3 (целочисленное деление)

5.Zsh vs Bash: Zsh - расширенное автодополнение, Bash - стандарт для скриптов

6.for синтаксис: верен, если LIMIT определен

7.Bash vs Python/C: Плюсы: интеграция с консолью, быстрое написание скриптов; Минусы: слабая типизация, медленные вычисления

# Заключение

* Реализованы базовые механизмы синхронизации процессов через файловые семафоры
* Разработаны утилиты для работы с документацией и генерации случайных последовательностей