

# НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

## Отчёт по программе генерации случайных чисел и отображения на семисегментном индикаторе

**Студент:** Ошаров Александр

**Семинарист:** Кензин Игорь

**Дата:** 2025-10-18

---

### Общее описание

Программа написана на языке ассемблера RISC-V и предназначена для демонстрации работы с генератором псевдослучайных чисел (PRNG) и выводом результата на виртуальный семисегментный индикатор в среде RARS. Программа бесконечно генерирует два случайных байта (значения от 0 до 255), отображает младшие 4 бита каждого числа как шестнадцатеричные цифры: сначала правую, затем левую, с задержкой 1 секунда между обновлениями.

### Структура проекта

Проект состоит из трёх файлов:

- `main.asm` — основная программа;
- `macros.asm` — определения констант и таблицы сегментов;
- `digit_display.asm` — подпрограмма `show_digit` для отображения цифры.

### Основной цикл ( `main.asm` )

#### 1. Генерация случайного числа:

- Используется системный вызов `42` ( `rand` ) с параметрами:
  - `a0 = 0` — идентификатор стандартного генератора;
  - `a1 = 256` — верхняя граница (исключительно).
- Результат возвращается в регистре `a0` и лежит в диапазоне `[0, 255]`.

#### 2. Отображение:

- Младшие 4 бита значения ( `a0 & 0xF` ) интерпретируются как шестнадцатеричная цифра.
- Вызывается подпрограмма `show_digit` с адресом индикатора:
  - `RIGHT_DIGIT = 0xFFFF0010` — правый индикатор;
  - `LEFT_DIGIT = 0xFFFF0011` — левый индикатор.

#### 3. Задержка:

- Используется системный вызов 32 ( `sleep` ) с аргументом 1000 мс.

#### 4. Цикличность:

- После отображения двух цифр программа возвращается к началу цикла ( `j loop` ).

## Подпрограмма `show_digit` ( `digit_display.asm` )

### Назначение

Преобразует 8-битное значение в сигнал для семисегментного индикатора и записывает его в указанный адрес памяти.

### Алгоритм

#### 1. Сохранение контекста:

- Сохраняются регистры `ra` , `s0` , `s1` в стек.
- Аргументы `a0` (значение) и `a1` (адрес индикатора) копируются в `s0` и `s1` .

#### 2. Отладочный вывод:

- Выводится сообщение входа в функцию, значение и адрес назначения (только в отладочном режиме).

#### 3. Формирование образа индикатора:

- Если значение  $\geq 16$ , устанавливается бит точки ( `DOT_BIT = 0x80` );
- Младшие 4 бита ( `s0 & 0xF` ) используются как индекс в таблицу `sevseg_table` .
- Из таблицы загружается байт-маска активных сегментов.
- При необходимости добавляется бит точки через `OR` .
- Результат записывается по адресу `s1` .

#### 4. Восстановление контекста и возврат.

## Таблица сегментов ( `sevseg_table` )

Содержит 16 байт, соответствующих шестнадцатеричным цифрам 0–F в формате семисегментного индикатора (биты `a–g` , без точки). Например:

- `0x3F` → цифра 0 ;
- `0x06` → цифра 1 ;
- `0x71` → цифра F .

### Особенности реализации

- **Без инициализации PRNG:** RARS автоматически инициализирует внутреннее состояние генератора.
- **Отладочный вывод:** Включён в `show_digit` для проверки корректности передаваемых значений и адресов.
- **Совместимость с RARS:** Используются только системные вызовы, поддерживаемые RARS (включая `rand` и `sleep` ).

- **Работа с памятью:** Запись в фиксированные адреса 0xFFFF0010 и 0xFFFF0011 , зарезервированные RARS для виртуальных индикаторов.

## Поведение программы

Программа непрерывно отображает пары случайных шестнадцатеричных цифр в формате [левая] [правая] с частотой ~0.5 Гц (по одной цифре в секунду). Если исходное число  $\geq 16$ , на соответствующем индикаторе загорается десятичная точка.

## Заключение

Реализация демонстрирует базовые навыки программирования на RISC-V: использование системных вызовов, работа с памятью, подпрограммы с сохранением контекста и отладочный вывод. Код структурирован, легко читаем и соответствует конвенциям вызова RISC-V.