Лабораторная работа 1

По дисциплине “Системы ввода-вывода”

“Принципы организации ввода/вывода без операционной системы”

Вариант 2

Выполнил:

Баянов Равиль Динарович

P3334

Санкт-Петербург

2025 г.

Оглавление

Цель

Познакомится с принципами организации ввода/выводе без операционной системы на примере компьютерной системы на базе процессора с архитектурой RISC-V и интерфейсом OpenSBI с использованием эмулятора QEMU.

Задачи

1. Реализовать функцию putchar вывода данных в консоль
2. Реализовать функцию getchar для получения данных из консоли
3. На базе реализованных функций putchar и getchar написать программу, позволяющую вызывать определенным вариантом функции OpenSBI посредством взаимодействия пользователя через меню
4. Запустить программу и выполнить вызов пунктов меню, получив результаты их работы
5. Оформить отчет по работе в электронном формате

Вариант

1. Get SBI implementation version
2. Hart get status (должно быть возможно задавать номер ядра)
3. Hart stop
4. System Shutdown

Описание функций

- *Putchar*

```
void putchar(char ch) {
    sbi_call(ch, 0, 0, 0, 0, 0, 0, 1 /* Console Putchar */);
}
```

Вызываем sbi_call и указываем 1 в EID, для записи символа в консоль.

- Getchar

```
char getchar() {
    struct sbiret ret;
    ret.error = -1;
    while (ret.error == -1) {
        ret = sbi_call(0, 0, 0, 0, 0, 0, 0, 2);
    }
    return (char) ret.error;
}
```

Вызываем sbi_call пока ret.error не станет отличным от -1, так как -1 это означает, что ничего не пришло.

- Get SBI implementation version

```
uint32_t get_sbi_version() {
    struct sbiret ret = sbi_call(0, 0, 0, 0, 0, 0, 0x1, 0x10);
    return (uint32_t) ret.value;
}

void print_sbi_version() {
    uint32_t version = get_sbi_version();

    printf("SBI Implementation Version: %d\n", version);
}
```

Вызов sbi_call с соответствующим EID, а именно 0x10 и указав правильный FID.

- Hart get status (должно быть возможно задавать номер ядра)

```
uint32_t get_hart_status(uint32_t hart_id) {
    struct sbiret ret = sbi_call(hart_id, 0, 0, 0, 0, 0, 0, 0x101);
    return ret.value;
}

void print_hart_status(uint32_t hart_id) {
    uint32_t status = get_hart_status(hart_id);
    printf("Hart %d Status: %d\n", hart_id, status);
}
```

Вызов sbi_call с соответствующим EID, а именно 0x101 и вывод на экран статуса.

- Hart stop

```
void stop_hart(uint32_t hart_id) {
    sbi_call(hart_id, 0, 0, 0, 0, 0, 0, 0x102);
}
```

Вызов sbi_call с соответствующим EID, а именно 0x102, предварительно указав id нужного нам hart.

- System Shutdown

```
void shutdown() {
    sbi_call(0, 0, 0, 0, 0, 0, 0, 0x08);
}
```

Вызов sbi_call с соответствующим EID, а именно 0x08.

Вывод

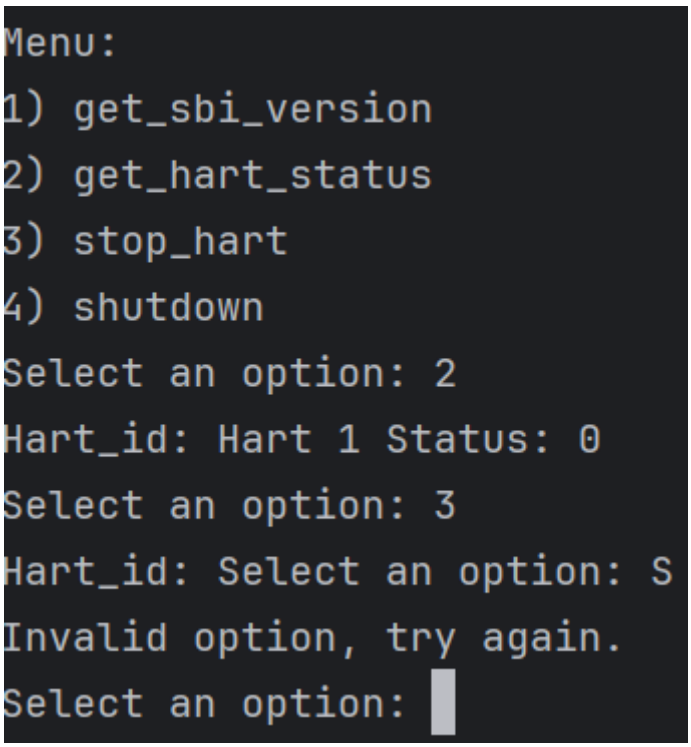
Get SBI implementation version

```
Menu:
1) get_sbi_version
2) get_hart_status
3) stop_hart
4) shutdown
Select an option: 1
SBI Implementation Version: 1
Select an option:
```

Hart get status (должно быть возможно задавать номер ядра)

```
Menu:
1) get_sbi_version
2) get_hart_status
3) stop_hart
4) shutdown
Select an option: 2
Hart_id: Hart 1 Status: 0
Select an option:
```

Hart stop



System Shutdown