

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ

Лабораторная работа №1

по дисциплине

«Системы ввода-вывода»

Вариант 1

Выполнили:

Векшин А. И. Р3316/CBB 1.2

Дашкевич Е.В. Р3308/CBB 1.2

Преподаватель:

Сергей Михайлович Табунщик

Введение

Цель:

Познакомится с принципами организации ввода/вывода 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

Базовые функции

getchar

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

putchar

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

Основное задание

Get SBI specification version

```
long get_sbi_specification_version() {
    return sbi_call(0, 0, 0, 0, 0, 0, 0, 0x10).value;
}

void print_specification_version() {
    char i[10];
    long res = get_sbi_specification_version();
    long high = (res & 0xef000000) >> 24;
    long low = res & 0x00ffffff;
    print("Specification version: ");
    itoa(high, i);
    print(i);
    putchar('.');
    itoa(low, i);
    println(i);
}
```

Get number of counters

```
long get_counters_num(){
    return sbi_call(0, 0, 0, 0, 0, 0, 0, 0x504D55).value;
}

void print_counters_num() {
    char i[20];
    itoa(get_counters_num(), i);
    print("Number of counters: ");
    println(i);
}
```

Get details of a counter

```
struct sbiret get_counter_info_sys_call(unsigned long counter_idx) {
    return sbi_call(counter_idx, 0, 0, 0, 0, 0, 1, 0x504D55);
}

void get_counter_info(){
    print("\n> ");
    char a = (char)getchar();
    putchar(a);
    putchar('\n');
```

```

unsigned long idx = (unsigned long)(a - '0');
struct sbiret ret = get_counter_info_sys_call(idx);
    long err = ret.error;
    if (err == -3) {
        println("Invalid counter");
        return;
    }

    println("Counter info: ");
    char buf[20];
    long value = ret.value;
    long csr = (value & 0x00000fff);
    long width = (value & 0x0003f000) >> 12;
    long type = (value & 0xa0000000) >> 31;

    if (type == 1) {
        println("Firmware counter, no additional info");
        return;
    }

    println("Hardware counter");
    print("CSR: ");
    itoa(csr, buf);
    println(buf);
    print("Width: ");
    itoa(width, buf);
    println(buf);
}

```

System Shutdown

```

long call_sys_shutdown() {
    return sbi_call(0, 0, 0, 0, 0, 0, 0, 0x8).error;
}

```

Результат работы программы

```
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 1
Specification version: 1.0
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 2
Number of counters: 19
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 2
Number of counters: 19
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 3
> 2
Counter info:
Hardware counter
CSR: 3074
Width: 63
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
```