

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Информационных Технологий и Систем

Лабораторная работа №1
«Принципы организации ввода/вывода без операционной системы»
по дисциплине
«Системы ввода вывода»
вариант №1

ЫПОЛНИЛ:

Студент группы Р3313

Султанов А.Р.

Проверил:

Табунщик С.М.

г. Санкт Петербург

2025г.

Введение

Цель:

Познакомиться с принципами организации ввода вывода без операционной системы на примере компьютерной системы на базе процессора с архитектурой RISC V и интерфейсом OpenS I 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 OpenS I 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

Решение

Реализация функций putchar, getchar:

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

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

Реализация функций по варианту:

```
struct sbiret sbi_call(long arg0, long arg1, long arg2, long arg3, long
arg4,
                        long arg5, long fid, long eid) {
    register long a0 __asm__("a0") = arg0;
    register long a1 __asm__("a1") = arg1;
    register long a2 __asm__("a2") = arg2;
    register long a3 __asm__("a3") = arg3;
    register long a4 __asm__("a4") = arg4;
    register long a5 __asm__("a5") = arg5;
    register long a6 __asm__("a6") = fid;
    register long a7 __asm__("a7") = eid;

    __asm__ __volatile__("ecall"
                        : "=r"(a0), "=r"(a1)
                        : "r"(a0), "r"(a1), "r"(a2), "r"(a3), "r"(a4),
"r"(a5),
                        "r"(a6), "r"(a7)
                        : "memory");
    return (struct sbiret){.error = a0, .value = a1};
}

struct SpecVersion {
    long major;
    long minor;
};

struct SpecVersion get_spec_version(void) {
    struct sbiret ret = sbi_call(0, 0, 0, 0, 0, 0, 0, 0x10 /* Base
extension */);

    return (struct SpecVersion) {
```

```

        .major = (ret.value >> 24) & (0b11111111),
        .minor = (ret.value) & (0b11111111111111111111111111111111),
    };
}

long get_number_of_counters(void) {
    struct sbiret ret = sbi_call(0, 0, 0, 0, 0, 0, 0, 0x504D55 /* PMU
*/);
    return ret.value;
}

enum CounterType {
    COUNTER_TYPE_HARDWARE = 0,
    COUNTER_TYPE_FIRMWARE = 1,
};

const long SBI_SUCCESS = 0;
const long SBI_ERR_INVALID_PARAM = -3;

struct CounterDetails {
    long status;
    long csr;
    long width;
    enum CounterType type;
};

struct CounterDetails get_details_of_a_counter(unsigned long id) {
    struct sbiret ret = sbi_call(id, 0, 0, 0, 0, 0, 1, 0x504D55 /* PMU
*/);
    return (struct CounterDetails) {
        .status = ret.error,
        .type = (ret.value >> 31) & 1,
        .width = (ret.value >> 12) & 0b11111111,
        .csr = ret.value & 0b1111111111111111,
    };
}

void system_shutdown(void) {
    sbi_call(0, 0, 0, 0, 0, 0, 0, 0x08 /* System shutdown */);
}

```

Скриншот работы с меню:

```

Boot HART ID           : 0
Boot HART Domain       : root
Boot HART Priv Version  : v1.12
Boot HART Base ISA     : rv32imafdc
Boot HART ISA Extensions : sstc,zicntr,zihpm,zicboz,zicbom,sdtrig,svadu
Boot HART PMP Count     : 16
Boot HART PMP Granularity : 2 bits
Boot HART PMP Address Bits: 32
Boot HART MHPM Info     : 16 (0x0007fff8)
Boot HART Debug Triggers : 2 triggers
Boot HART MIDELEG       : 0x00001666
Boot HART MEDELEG       : 0x00f0b509
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 1
Major: 2, Minor: 0
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 2
Number of counters: 35
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 3
Enter ID: 1
Invalid ID provided
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 3
Enter ID: 12
Type: hardware, Width: 63, CSR: 3084
1. Get SBI specification version
2. Get number of counters
3. Get details of a counter
4. System Shutdown
> 4
Shutting down...

```

Исходный код

Исходный код доступен по ссылке:

<https://github.com/sultanowskii/itmo-edu/tree/master/io-systems/lab1>