



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА

ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)

НАПРАВЛЕНИЕ ПОДГОТОВКИ

09.03.04 Программная инженерия

## О Т Ч Е Т

По практикуму № 1

*«Разработка и отладка программ в вычислительном  
комплексе Тераграф»*

Дисциплина: Архитектура ЭВМ

Вариант: 17

Студент

ИУ7-51Б

(Группа)

Савинова М. Г.

(Ф. И. О)

Преподаватель

Ибрагимов С. В.

2023 г.

## Оглавление

Демо проект 1 .....	3
Демо проект 2 .....	3
Демо проект 3 .....	4
Индивидуальное задание .....	4
Заключение .....	7

## Демо проект 1

Пример демонстрирует основные механизмы инициализации гетерогенных ядер *gpc* и взаимодействие хост-подсистем с *Graph Processor Core*, используются аппаратные очереди.

### 1. Установка репозитория:

```
git clone --recursive  
https://latex.bmstu.ru/gitlab/hackathon2023/lab1/lab1.git  
cd lab1
```

### 2. Сборка проекта:

```
make
```

### 3. Запуск:

```
host/host_main sw-kernel/sw_kernel.rawbinary
```

### 4. Результат работы теста:

```
Open gpc on /dev/gpc0  
Rawbinary loaded from sw-kernel/sw_kernel.rawbinary  
sw_kernel version: 0x28102023  
Leonhard clock frequency (LNH_CF) 180.032751 MHz  
Test done
```

## Демо проект 2

Программа демонстрирует принципы взаимодействия устройств в системе и примеры хранения и обработки множеств в микропроцессоре *Lnh64*.

### 1. Установка репозитория:

```
git clone --recursive  
https://latex.bmstu.ru/gitlab/hackathon2023/lab2/lab2.git  
cd lab2
```

### 2. Сборка проекта:

```
make
```

### 3. Запуск:

```
host/host_main sw-kernel/sw_kernel.rawbinary  
select role from users where user=5 and time>7200;
```

### 4. Результат работы теста:

```
Роль: 999 - Время доступа: 3596400  
Роль: 998 - Время доступа: 3592800
```

```
Роль: 997 - Время доступа: 3589200
Роль: 996 - Время доступа: 3585600
Роль: 995 - Время доступа: 3582000
Роль: 994 - Время доступа: 3578400
```

## 5. Просмотр лога:

```
tail -f lab2.log
Открывается доступ к /dev/gpc2
Программное ядро загружено из файла sw-kernel/sw_kernel.rawbinary
Введен запрос: select role from users where user=5 and time>7200;
Запрос принят в обработку.
Поиск ролей пользователя 5и time > 7200
```

## Демо проект 3

Пример демонстрирует основные механизмы взаимодействия

микропроцессоров *CPE (resv64)* и *SPE (lnh64)*, и выполняет тестирование корректности команд *DISC*.

### 1. Установка репозитория:

```
git clone --recursive
https://latex.bmstu.ru/gitlab/hackathon2023/lab3.git
cd lab3
```

### 2. Сборка проекта:

```
make
```

### 3. Запуск:

```
host/host_main sw-kernel/sw_kernel.rawbinary
```

### 4. Результат работы теста:

```
Открывается доступ к /dev/gpc1
Программное ядро загружено из файла sw-kernel/sw_kernel.rawbinary
Ядро /dev/gpc1. Эксперимент 1 - тест вставки и поиска. Ошибок: 0
Ядро /dev/gpc1. Эксперимент 2 - тест вставки в заполненную
ассоциативную память. Ошибок: 0
Ядро /dev/gpc1. Эксперимент 3 - тест команд ближайшего меньшего и
большого (nsm и ngr). Ошибок: 0
...
```

## Индивидуальное задание

- **Устройство вычисления обратной функции.** Сформировать в хост-подсистеме и передать в SPE 256 записей key-value со значениями функции  $f(x)=x^2$  в диапазоне значений  $x$  от 0 до 1048576 (где  $f(x)$  - ключ,  $x$  - значение). Выполнить тестирование работы устройства, посылая из хост-подсистемы значение  $f(x)$  и получая от sw\_kernel

значение  $x$ . Если указанного значения  $f(x)$  не сохранено в SPE, выполнить поиск ближайшего (меньшего или большего) значения к  $f(x)$  и вернуть соответствующий  $x$ . Сравнить результат с ожидаемым.

- Измененный код *common\_struct.h*:

```
#ifndef COMMON_STRUCT
#define COMMON_STRUCT

#ifdef __riscv64__
#include "map.h"
#endif
#include "compose_keys.hxx"

//Номера структур данных в SPE
enum Structures : uint32_t {
    null = 0, //Нулевая структура не используется
    funcs_pnum = 1, //Таблица 1
    resources_pnum = 2 //Таблица 2
};

#ifdef __riscv64__
//Задание диапазонов и курсоров
template<typename Range>
struct reverse {
    Range r;
    [[gnu::always_inline]] reverse(Range r) : r(r) {}
    [[gnu::always_inline]] auto begin() {return r.rbegin();}
    [[gnu::always_inline]] auto end() {return r.rend();}
};

template<typename K, typename V>
struct Handle {
    bool ret_val;
    K k{get_result_key<K>()};
    V v{get_result_value<V>()};
    [[gnu::always_inline]] Handle(bool ret_val) : ret_val(ret_val) {}

    [[gnu::always_inline]] operator bool() const {
        return ret_val;
    }

    [[gnu::always_inline]] K key() const {
        return k;
    }

    [[gnu::always_inline]] V value() const {
        return v;
    }
};
#endif

////////////////////////////////////
// Описание формата ключа и значения
////////////////////////////////////
```

```

struct funcs {
    using vertex_t = uint32_t;
    int struct_number;
    constexpr funcs(int struct_number) : struct_number(struct_number) {}

    //Запись для формирования ключей (* - наиболее значимые биты
поля)
    STRUCT(key)
    {
        uint64_t f_x :64;
    };

    //Запись для формирования значений
    STRUCT(val)
    {
        uint64_t x :64;
    };
    //Обязательная типизация
    #ifdef __riscv64
    DEFINE_DEFAULT_KEYVAL(key, val)
    #endif
};

constexpr funcs FUNCS(Structures::funcs_pnum);

#endif //COMMON_STRUCT

```

- Измененный код функции *select* в *sw\_kernel\_main.cpp*:

```

void select() {
    while(1){

        uint64_t cur_f_x = mq_receive();

        if (cur_f_x == -1)
            break;

        auto crole = FUNCS.search(funcs::key{.f_x = cur_f_x});

        uint64_t result = crole.value();

        if (!(crole))
        {
            uint64_t small_result = -1;
            uint64_t great_result = -1;

            // МЕНЬШИЙ
            auto small = FUNCS.nsm(funcs::key{.f_x =
cur_f_x});

            // БОЛЬШИЙ
            auto great = FUNCS.ngr(funcs::key{.f_x =
cur_f_x});

            small_result = small.value();
            great_result = great.value();

            if (small && great)

```

```

        if (abs(cur_f_x - small.key()) <=
abs(great.key() - cur_f_x))
            mq_send(small_result);
        else
            mq_send(great_result);

        else if (small)
            mq_send(small_result);
        else
            mq_send(great_result);
    }
    else
        mq_send(crole.value());
}
}

```

- Пример работы программы:

```

Введите значение: 36
Ожидаемый результат: 6
Полученный результат: 6

Введите значение: 35
Ожидаемый результат: 5.91608
Полученный результат: 6

Введите значение: 25
Ожидаемый результат: 5
Полученный результат: 5

Введите значение: 26
Ожидаемый результат: 5.09902
Полученный результат: 5

Введите значение: 24
Ожидаемый результат: 4.89898
Полученный результат: 5

Введите значение: -1
Неверный ввод!

```

## Заключение

В результате работы изучены принципы работы вычислительного комплекса Тераграф и получены практические навыки решения задач обработки множеств на основе гетерогенной вычислительной структуры.