Университет ИТМО
Кафедра ВТ

Лабораторная работа №3
Низкоуровневое программирование

Выполнил: Федоров Сергей
Группа: P33113

Санкт-Петербург
2020 г.

Задание лабораторной работы:

В данной лабораторной работе, требовалось написать две простые функции на языке C, используя лучшие практики модных домов Парижа.

Функции для реализации:

- scalar_product – считает скалярное произведение двух массивов
- is_prime – проверка

Выполнение:

Files structure:

```
lab3/
├── Makefile
└── main.c
```

main.c

```c
#include <stdio.h>

void print_array(const int arr[], size_t s){
    size_t i;
    for(i = 0; i < s; i++){
        printf("%d ", arr[i]);
    }
    puts("");
}

int scalar_product(const int arr_a[], const int arr_b[], size_t s){
    int sum = 0;
    size_t iter;
    for(iter = 0; iter < s; iter++){
        sum += arr_a[iter] * arr_b[iter];
    }
    return sum;
}

int is_prime(unsigned long n){
    if (n == 0LU || n == 1LU){
        return 0;
    } else {
        unsigned long checks;
        for(checks = 2; checks < n; checks++) if (n % checks == 0LU) return 0;
    }

    return 1;
}

int arr_a[] = {1, 2, 3, 4};
int arr_b[] = {3, 2, 1, 1};

unsigned long number;

int main() {

    /* Scalar product of arrays */
    size_t s = sizeof(arr_a) / sizeof(int);
    puts("Given arrays:");
```

```
    print_array(arr_a, s); print_array(arr_b, s);
    puts("Scalar product: ");
    printf("%d \n", scalar_product(arr_a, arr_b, s));


    /* Is read number prime? */
    puts("Enter desired number: ");
    if (scanf("%lu", &number)) {
        char* ans[] = {"not prime", "prime"};
        printf("Number %lu is %s!\n", number, ans[is_prime(number)]);
    } else {
        puts("Reading problem occured");
    }
}
```

Makefile

```
all: main

main: main.o
    gcc -o main main.o

main.o: main.c
    gcc --ansi -pedantic-errors -Wall -Werror -c -o main.o main.c

clean:
    rm -f main.o main
```

Вывод:

Прежде всего, даже учитывая синтаксический сахар, который добавляет информации, которую нужно держать в голове, писать на языке C, становится гораздо легче и приятнее в сравнении с ASM.

Также заметно удобнее стало собирать исполняемый файл, так как gcc так же берет на себя этап линковки объектных файлов.