

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-
вычислительных систем (БИС)

КОМБИНИРОВАННЫЕ ПРОГРАММЫ. СВЯЗЫВАНИЕ РАЗНОЯЗЫКОВЫХ
МОДУЛЕЙ.

Отчет по лабораторной работе №3
по дисциплине «Системное программирование»
1 вариант

Студент гр. 738-1

_____ Г.К. Беляев

“ ” _____ 2022г.

Принял

Преподаватель кафедры КИБЭВС

Е.О. Калинин

1 Введение

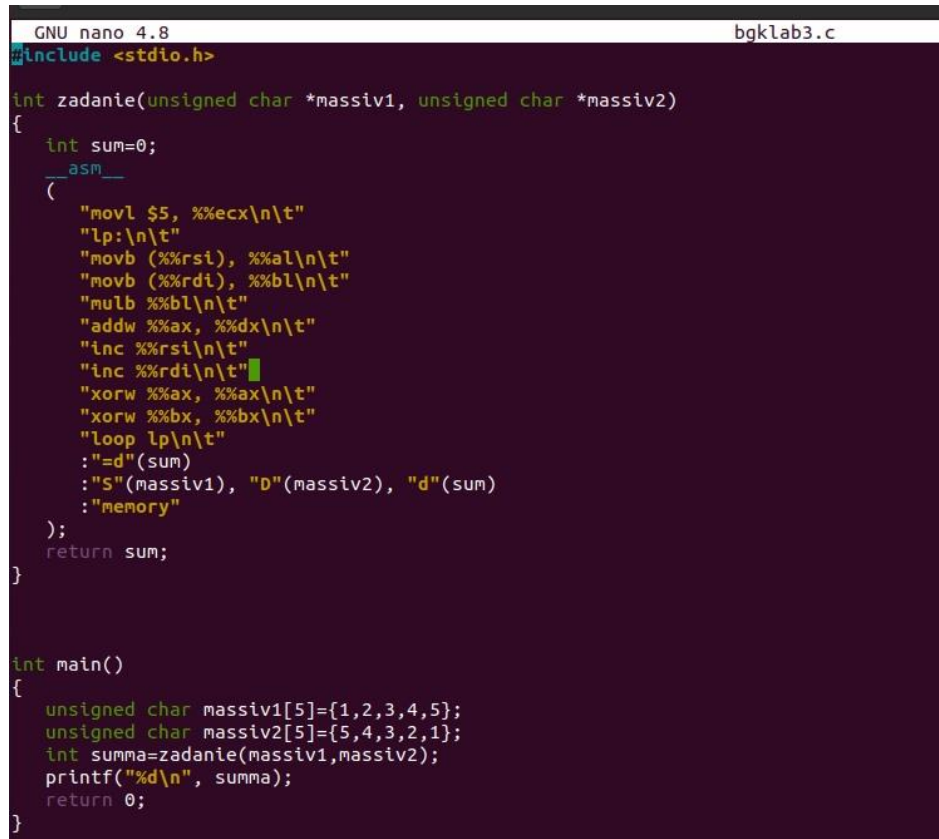
Цель работы: познакомиться с основными способами передачи параметров подпрограмм, особенностями передачи управления между модулями, научиться писать комбинированные программы, в которых модули Ассемблера вызываются из модулей, написанных на высокоуровневых языках программирования.

Задание на лабораторную работу:

Задача: напишите программу, в которой создается два числовых массива одинакового размера. Необходимо вычислить сумму попарных произведений элементов этих массивов. Так, если через a_k и b_k , обозначить элементы массивов (индекс $0 \leq k < n$), то необходимо вычислить сумму $\sum_{k=0}^{n-1} a_k b_k$.

2 Ход Работы

Согласно заданию, была написана программа со вставкой на языке Си (рисунок 2.1). После программа была запущена и выдан ее результат (рисунок 2.2).

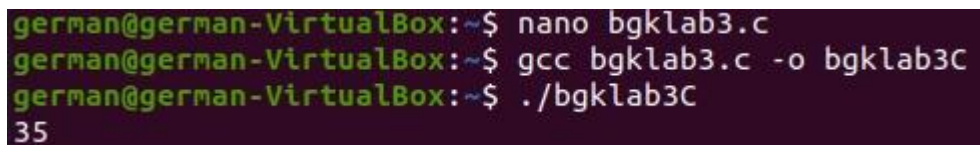


```
GNU nano 4.8 bgklab3.c
#include <stdio.h>

int zadanie(unsigned char *massiv1, unsigned char *massiv2)
{
    int sum=0;
    __asm__
    (
        "movl $5, %%ecx\n\t"
        "lp:\n\t"
        "movb (%rsi), %%al\n\t"
        "movb (%rdi), %%bl\n\t"
        "mulb %%bl\n\t"
        "addw %%ax, %%dx\n\t"
        "inc %rsi\n\t"
        "inc %rdi\n\t"
        "xorw %%ax, %%ax\n\t"
        "xorw %%bx, %%bx\n\t"
        "loop lp\n\t"
        : "d"(sum)
        : "S"(massiv1), "D"(massiv2), "d"(sum)
        : "memory"
    );
    return sum;
}

int main()
{
    unsigned char massiv1[5]={1,2,3,4,5};
    unsigned char massiv2[5]={5,4,3,2,1};
    int summa=zadanie(massiv1,massiv2);
    printf("%d\n", summa);
    return 0;
}
```

Рисунок 2.1 – Код программы



```
german@german-VirtualBox:~$ nano bgklab3.c
german@german-VirtualBox:~$ gcc bgklab3.c -o bgklab3C
german@german-VirtualBox:~$ ./bgklab3C
35
```

Рисунок 2.2 – Результат работы программы

Теперь создадим docker контейнер (рисунок 2.3).

Ссылка на github: <https://github.com/CoReShAdA/->

```
german@german-VirtualBox:~$ docker build -t lab3 -f dockerfilebgk .
Sending build context to Docker daemon 152.9MB
Step 1/7 : FROM ubuntu
----> 2b4cba85892a
Step 2/7 : RUN apt update
----> Using cache
----> dc5965e80136
Step 3/7 : RUN apt install gcc -y
----> Using cache
----> 6917906afcaa
Step 4/7 : RUN apt install nano -y
----> Using cache
----> 9792c00f7482
Step 5/7 : COPY bgklab3.c .
----> Using cache
----> c1d6f7be1046
Step 6/7 : RUN gcc bgklab3.c -o bgklab3C
----> Using cache
----> f1d49dccf516
Step 7/7 : CMD ./bgklab3C
----> Running in 2f9c6e77452e
Removing intermediate container 2f9c6e77452e
----> 94138e194ab5
Successfully built 94138e194ab5
Successfully tagged lab3:latest
german@german-VirtualBox:~$ docker run -it lab3
35
```

Рисунок 2.3 – Создание контейнера

3 Заключение

В ходе работы было произведено ознакомление с основными способами передачи параметров подпрограмм, особенностями передачи управления между модулями, научиться писать комбинированные программы, в которых модули Ассемблера вызываются из модулей, написанных на высокоуровневых языках программирования.