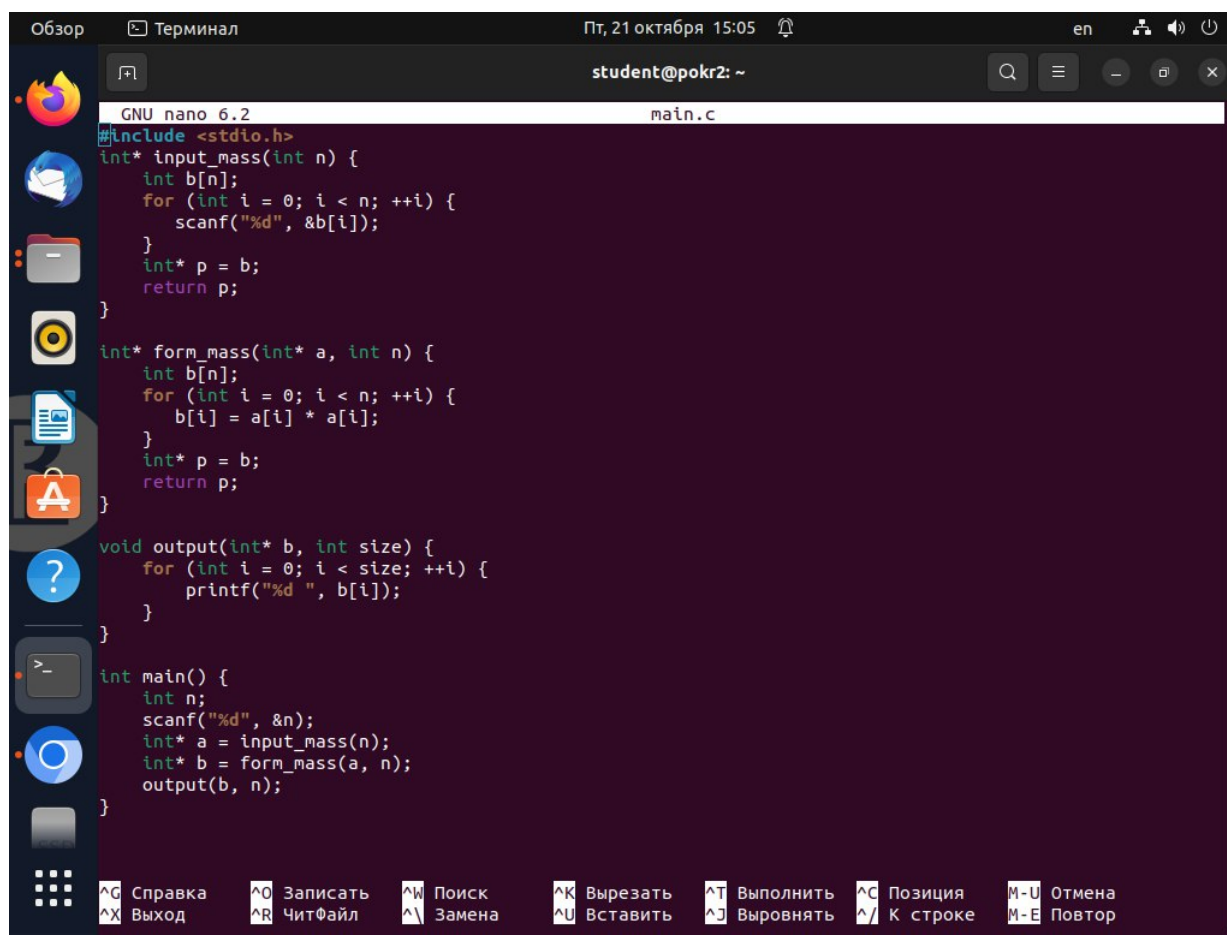


Ефимов Иван БПИ 212
Домашнее задание №1 ABC.
Вариант 39

Предварительно написал код на C, ввод, формирование, вывод массива выделил в отдельные функции Далее открыл этот код в nano

```
1  #include <stdio.h>
2  int* input_mass(int n) {
3      int b[n];
4      for (int i = 0; i < n; ++i) {
5          scanf("%d", &b[i]);
6      }
7      int* p = b;
8      return p;
9  }
10
11 int* form_mass(int* a, int n) {
12     int b[n];
13     for (int i = 0; i < n; ++i) {
14         b[i] = a[i] * a[i];
15     }
16     int* p = b;
17     return p;
18 }
19
20 void output(int* b, int size) {
21     for (int i = 0; i < size; ++i) {
22         printf("%d ", b[i]);
23     }
24 }
25
26 int main() {
27     int n;
28     scanf("%d", &n);
29     int* a = input_mass(n);
30     int* b = form_mass(a, n);
31     output(b, n);
32 }
```

Далее открыл этот код в nano



```
Обзор Терминал Пт, 21 октября 15:05 en
student@pokr2: ~
GNU nano 6.2 main.c
#include <stdio.h>
int* input_mass(int n) {
    int b[n];
    for (int i = 0; i < n; ++i) {
        scanf("%d", &b[i]);
    }
    int* p = b;
    return p;
}

int* form_mass(int* a, int n) {
    int b[n];
    for (int i = 0; i < n; ++i) {
        b[i] = a[i] * a[i];
    }
    int* p = b;
    return p;
}

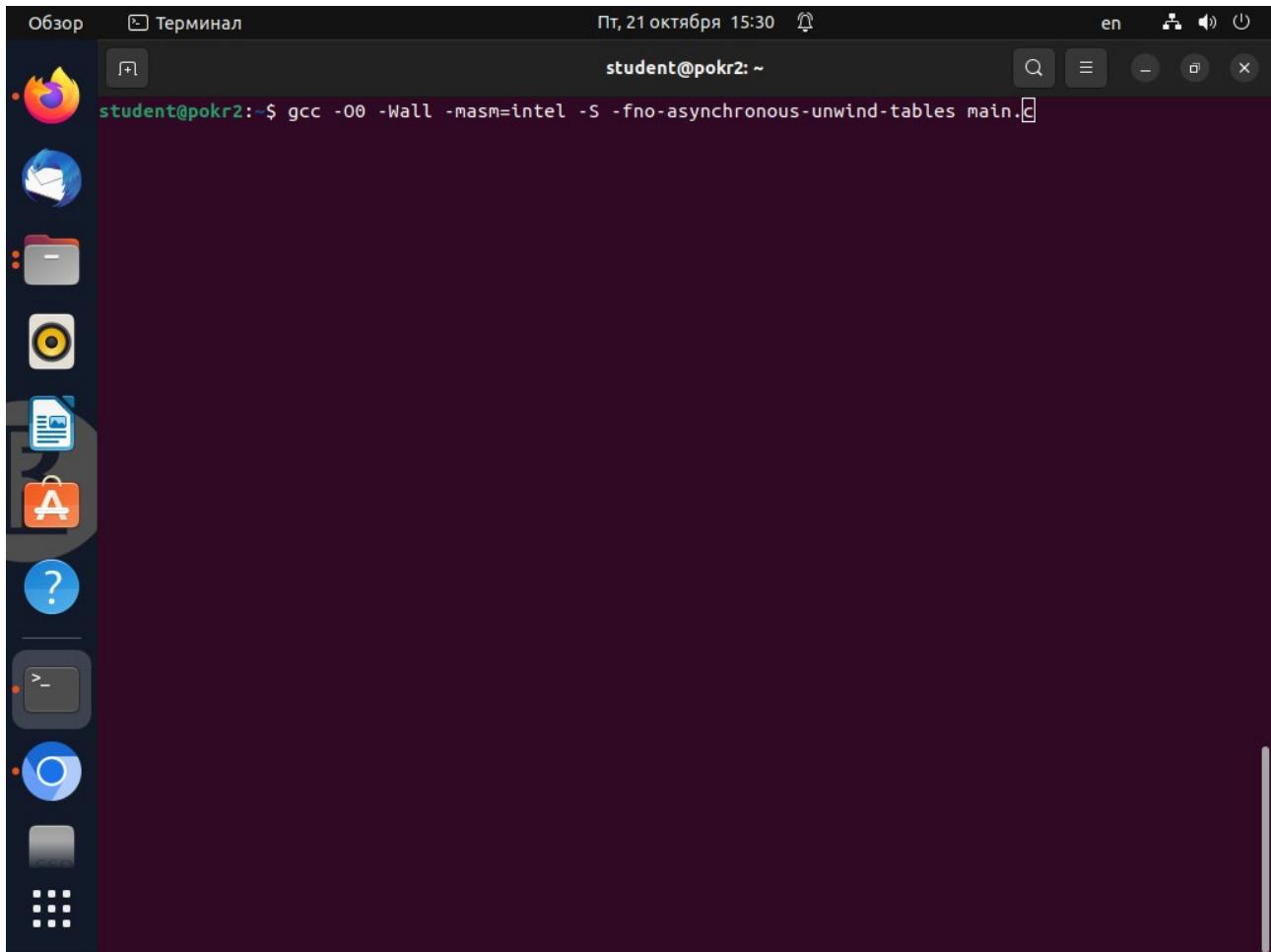
void output(int* b, int size) {
    for (int i = 0; i < size; ++i) {
        printf("%d ", b[i]);
    }
}

int main() {
    int n;
    scanf("%d", &n);
    int* a = input_mass(n);
    int* b = form_mass(a, n);
    output(b, n);
}
```

⌘ Справка ⌘ Записать ⌘ Поиск ⌘ Вырезать ⌘ Выполнить ⌘ Позиция ⌘ Отмена
⌘ Выход ⌘ ЧитФайл ⌘ Замена ⌘ Вставить ⌘ Выводить ⌘ К строке ⌘ Повтор

Далее скомпилировал программу, получил исполняемый файл source (использовал команду `$ gcc source.c -o source`, этот момент забыл заскринить, простите)

Далее нужно было получить ассемблерный файл. Это я сделал с помощью команды



```
student@pokr2: ~  
student@pokr2:~$ gcc -O0 -Wall -masm=intel -S -fno-asynchronous-unwind-tables main.c
```

Получил Ассемблерный файл main.s

Так как я прописал дополнительные команды, код получился максимально компактным(если можно так сказать про код на 300 строк), без лишних символов.

Ассемблерный файл прикреплю в гите.

Далее нужно было заняться комментированием в ассемблере. Я выбрал комментировать по блокам, так как каждую строчку для меня тяжело грамотно объяснить, а вот блоки программы я разобрать могу.

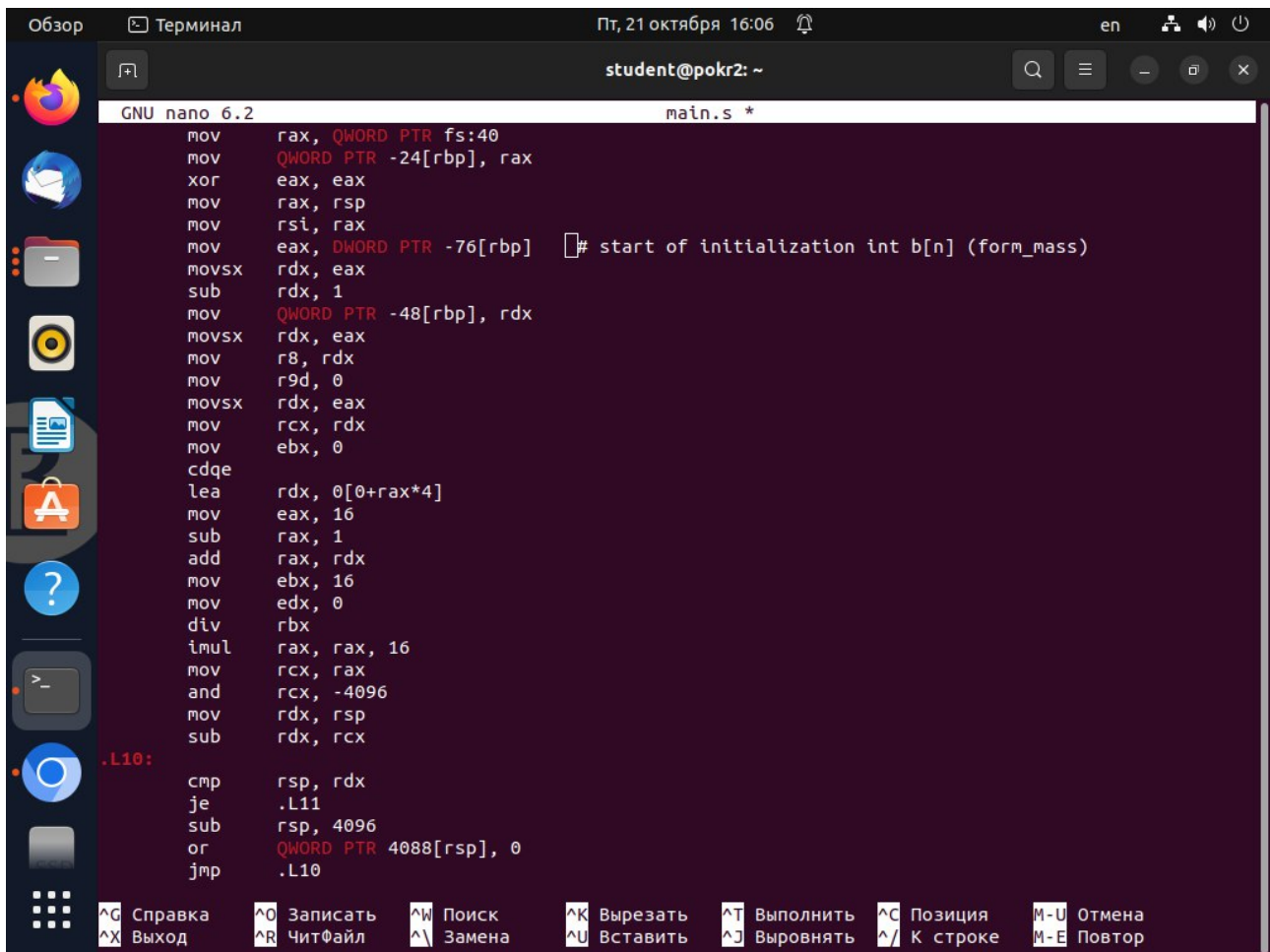
Первый блок: Декларация массива `int b[n]` в функции `input_mass` (в коде я указал Initialization, хотя конечно правильней было бы указать Declaration)

```
Обзор Терминал student@pokr2: ~
GNU nano 6.2 main.s *
mov     QWORD PTR -24[rbp], rax
xor     eax, eax
mov     rax, rsp
mov     rbx, rax
mov     eax, DWORD PTR -68[rbp] # Initialization of int b[n]
movsx   rdx, eax
sub     rdx, 1
mov     QWORD PTR -48[rbp], rdx
movsx   rdx, eax
mov     r10, rdx
mov     r11d, 0
movsx   rdx, eax
mov     r8, rdx
mov     r9d, 0
cdq     rdx
lea     rdx, 0[0+rax*4]
mov     eax, 16
sub     rax, 1
add     rax, rdx
mov     esi, 16
mov     edx, 0
div     rsi
imul    rax, rax, 16
mov     rcx, rax
and     rcx, -4096
mov     rdx, rsp
sub     rdx, rcx
.L2:    cmp     rsp, rdx
je      .L3
sub     rsp, 4096
or      QWORD PTR 4088[rsp], 0
jmp     .L2 # end of initialization int b[n]
.L3:
^G Справка      ^O Записать    ^W Поиск       ^K Вырезать    ^T Выполнить   ^C Позиция     M-U Отмена
^X Выход        ^R ЧитФайл    ^_ Замена     ^U Вставить    ^J Выводить    ^/_ К строке   M-E Повтор
```

Далее идет блок инициализации `int* p` массивом `b` и `return p` из функции `input_mass`

```
Обзор Терминал student@pokr2: ~
GNU nano 6.2 main.s *
sub     rax, 8
add     rax, rsp
or      QWORD PTR [rax], 0
.L4:    mov     rax, rsp
add     rax, 3
shr     rax, 2
sal     rax, 2
mov     QWORD PTR -40[rbp], rax
mov     DWORD PTR -52[rbp], 0
.L6:    mov     eax, DWORD PTR -52[rbp]
cdq     rax
lea     rdx, 0[0+rax*4]
mov     rax, QWORD PTR -40[rbp]
add     rax, rdx
mov     rsi, rax
lea     rax, .LC0[rip]
mov     rdi, rax
mov     eax, 0
call    __isoc99_scanf@PLT
add     DWORD PTR -52[rbp], 1
.L5:    mov     eax, DWORD PTR -52[rbp]
cmp     eax, DWORD PTR -68[rbp]
jl      .L6
mov     rax, QWORD PTR -40[rbp] # initialization of int* p of b
mov     QWORD PTR -32[rbp], rax # end of initialization
mov     rax, QWORD PTR -32[rbp] #return p
mov     rsp, rbx # end of return p
mov     rdx, QWORD PTR -24[rbp]
sub     rdx, QWORD PTR fs:40
je      .L8
.L8:
```

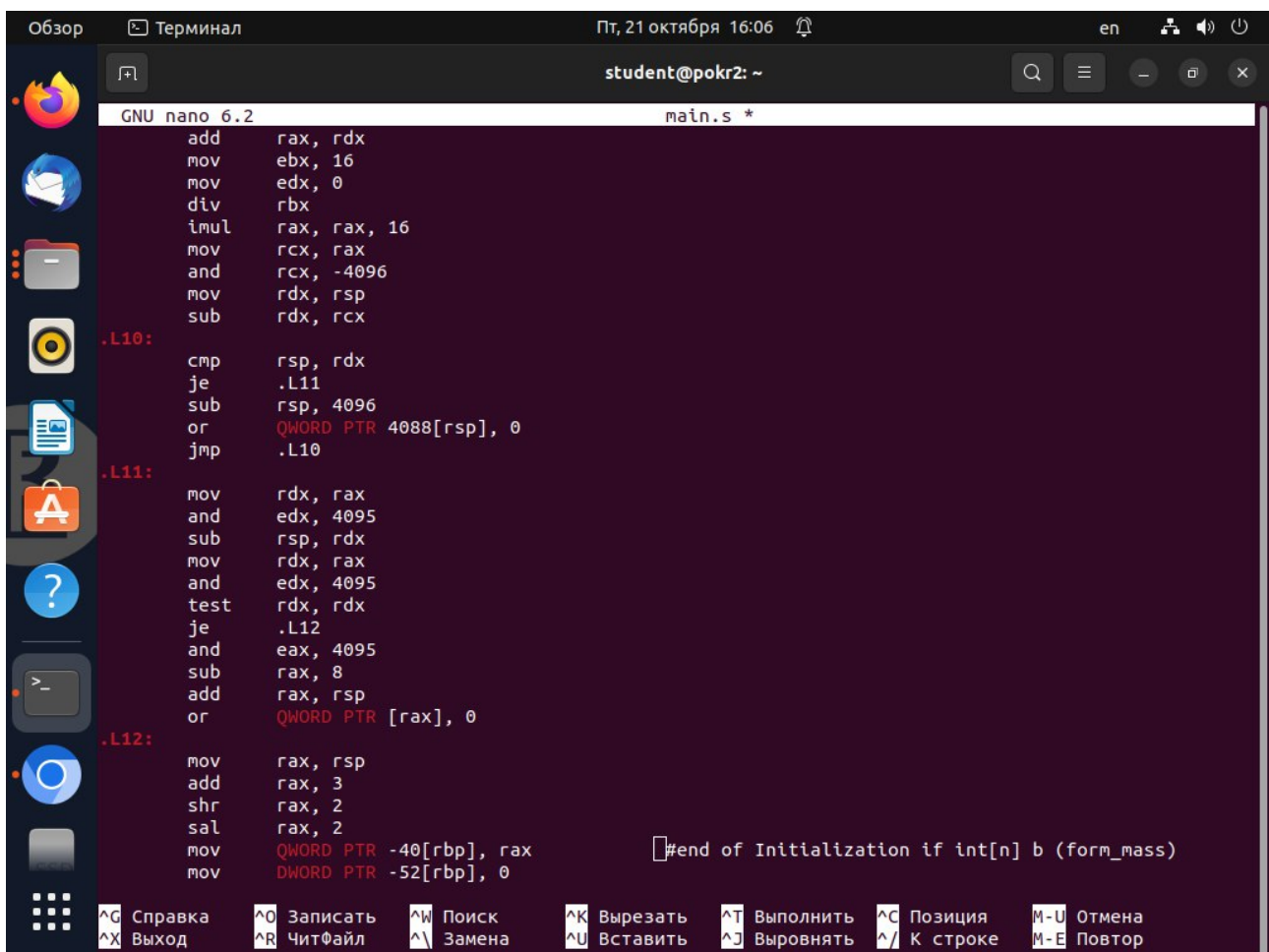
Далее декларация и инициализация `int b[n]` в массиве `form_mass`



```
GNU nano 6.2 main.s *
mov     rax, QWORD PTR fs:40
mov     QWORD PTR -24[rbp], rax
xor     eax, eax
mov     rax, rsp
mov     rsi, rax
mov     eax, DWORD PTR -76[rbp]  [# start of initialization int b[n] (form_mass)
movsx   rdx, eax
sub     rdx, 1
mov     QWORD PTR -48[rbp], rdx
movsx   rdx, eax
mov     r8, rdx
mov     r9d, 0
movsx   rdx, eax
mov     rcx, rdx
mov     ebx, 0
cdq     rdx
lea     rdx, 0[0+rax*4]
mov     eax, 16
sub     rax, 1
add     rax, rdx
mov     ebx, 16
mov     edx, 0
div     rbx
imul    rax, rax, 16
mov     rcx, rax
and     rcx, -4096
mov     rdx, rsp
sub     rdx, rcx

.L10:
cmp     rsp, rdx
je      .L11
sub     rsp, 4096
or      QWORD PTR 4088[rsp], 0
jmp     .L10

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить     ^J Выводить     ^/_ К строке    M-E Повтор
```



```
GNU nano 6.2 main.s *
add     rax, rdx
mov     ebx, 16
mov     edx, 0
div     rbx
imul    rax, rax, 16
mov     rcx, rax
and     rcx, -4096
mov     rdx, rsp
sub     rdx, rcx

.L10:
cmp     rsp, rdx
je      .L11
sub     rsp, 4096
or      QWORD PTR 4088[rsp], 0
jmp     .L10

.L11:
mov     rdx, rax
and     edx, 4095
sub     rsp, rdx
mov     rdx, rax
and     edx, 4095
test    rdx, rdx
je      .L12
and     eax, 4095
sub     rax, 8
add     rax, rsp
or      QWORD PTR [rax], 0

.L12:
mov     rax, rsp
add     rax, 3
shr     rax, 2
sal     rax, 2
mov     QWORD PTR -40[rbp], rax  [#end of Initialization if int[n] b (form_mass)
mov     DWORD PTR -52[rbp], 0

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить     ^J Выводить     ^/_ К строке    M-E Повтор
```


Далее инициализация `int* p` массивом `b` и `return p` (`form_mass`)

Комментарии в функции output поместились все на одну картинку

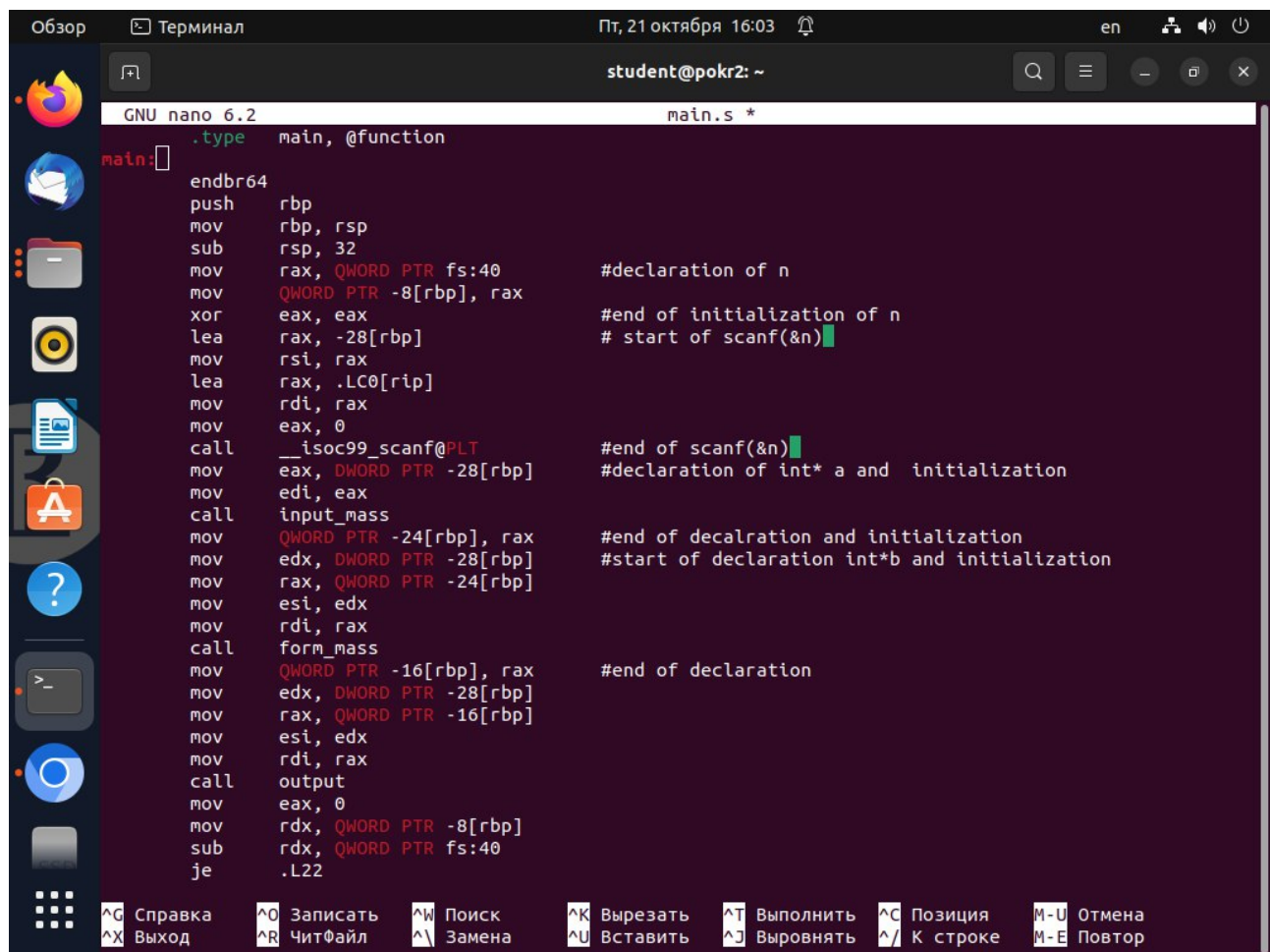
```

GNU nano 6.2                                main.s *
mov     DWORD PTR -28[rbp], esi
mov     DWORD PTR -4[rbp], 0
jmp     .L18
.L19:
mov     eax, DWORD PTR -4[rbp]                # start of printf(b[i])
cdq     rax
lea     rdx, 0[0+rax*4]
mov     rax, QWORD PTR -24[rbp]
add     rax, rdx
mov     eax, DWORD PTR [rax]
mov     esi, eax
lea     rax, .LC1[rip]
mov     rdi, rax
mov     eax, 0
call    printf@PLT                          #end of printf(b[i])
add     DWORD PTR -4[rbp], 1
.L18:
mov     eax, DWORD PTR -4[rbp]
cmp     eax, DWORD PTR -28[rbp]
jl      .L19
nop
nop
leave
ret
.size   output, .-output
.globl  main
.type   main, @function

main:
endbr64
push   rbp
mov     rbp, rsp
sub     rsp, 32
mov     rax, QWORD PTR fs:40                #declaration of n
mov     QWORD PTR -8[rbp], rax

```

Так же, как и в функции main



```
GNU nano 6.2 main.s *
.type main, @function
main:
    endbr64
    push rbp
    mov rbp, rsp
    sub rsp, 32
    mov rax, QWORD PTR fs:40      #declaration of n
    mov QWORD PTR -8[rbp], rax
    xor eax, eax                 #end of initialization of n
    lea rax, -28[rbp]             # start of scanf(&n)
    mov rsi, rax
    lea rax, .LC0[rip]
    mov rdi, rax
    mov eax, 0
    call __isoc99_scanf@PLT      #end of scanf(&n)
    mov eax, DWORD PTR -28[rbp]  #declaration of int* a and initialization
    mov edi, eax
    call input_mass
    mov QWORD PTR -24[rbp], rax   #end of decalration and initialization
    mov edx, DWORD PTR -28[rbp]  #start of declaration int*b and initialization
    mov rax, QWORD PTR -24[rbp]
    mov esi, edx
    mov rdi, rax
    call form_mass
    mov QWORD PTR -16[rbp], rax   #end of declaration
    mov edx, DWORD PTR -28[rbp]
    mov rax, QWORD PTR -16[rbp]
    mov esi, edx
    mov rdi, rax
    call output
    mov eax, 0
    mov rdx, QWORD PTR -8[rbp]
    sub rdx, QWORD PTR fs:40
    je .L22

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить     ^J Выводить     ^_ К строке    M-E Повтор
```

Комментарии писал вручную

Все их вы посмотреть можете в файле

Теперь нужно тестовое покрытие. Ассемблерный файл скомпилировал командами

```
$ as main.s -main.o
$ ld main.o -o source1
```

То есть source и source1 - это исполнимые файлы на языке C и ASM соответственно.

Далее будет представлено тестовое покрытие.

Обзор Терминал Пт, 21 октября 16:37 en

student@pokr2: ~

```
student@pokr2:~$ ./source
5
0 2 0 4000 0
0 4 0 16000000 0 student@pokr2:~$ ./source1
0 2 0 4000 0
student@pokr2:~$ ./source1
5
0 2 0 4000 0
0 4 0 16000000 0 student@pokr2:~$
```

Обзор Терминал Пт, 21 октября 16:36 en

student@pokr2: ~

```
student@pokr2:~$ ./source
10
-1 1 -2 2 -3 3 -4 4 -5 5
1 1 4 4 9 9 16 16 25 25 student@pokr2:~$ ./source1
10
-1 1 -2 2 -3 3 -4 4 -5 5
1 1 4 4 9 9 16 16 25 25 student@pokr2:~$
```

Обзор Терминал Пт, 21 октября 16:34 en

student@pokr2: ~

```
student@pokr2:~$ ./source
10
1 1 1 1 1 2 3 9 10
1
1 1 1 1 1 4 9 81 100 1 student@pokr2:~$ ./source1
10
1 1 1 1 1 2 3 9 10 1
1 1 1 1 1 4 9 81 100 1 student@pokr2:~$
```

Обзор Терминал Пт, 21 октября 16:33 en

student@pokr2: ~

```
student@pokr2:~$ ./source
0
student@pokr2:~$ ./source
0
student@pokr2:~$ ./source1
0
student@pokr2:~$
```


Обзор Терминал Пт, 21 октября 16:32 en

student@pokr2: ~

```
student@pokr2:~$ ./source
4
1 2 3 4
1 4 9 16 student@pokr2:~$ ./source1
4
1 2 3 4
1 4 9 16 student@pokr2:~$
```

Обзор Терминал Пт, 21 октября 16:37 en

student@pokr2: ~

```
student@pokr2:~$ ./source
5
0 2 0 4000 0
0 4 0 16000000 0 student@pokr2:~$ ./source1
0 2 0 4000 0
student@pokr2:~$ ./source1
5
0 2 0 4000 0
0 4 0 16000000 0 student@pokr2:~$
```

На этом все, мог сделать больше, но работал в жестких ограничениях по времени (Пока компьютерный класс на Покровке не закроется, так как работать с ПК пока не могу). В следующий раз надеюсь сделать больше!

