

Отчёт по лабораторной работе №7

**Команды безусловного и условного переходов в
Nasm.Программирование ветвлений.**

Сидорова Арина Валерьевна”

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	1. Реализация переходов в NASM	6
3.2	Изучение структуры файлы листинга	11
3.3	Задание для самостоятельной работы	14
4	Выводы	18

Список иллюстраций

3.1	Создаем каталог и файл	6
3.2	Заполняем файл	7
3.3	Запускаем файл	7
3.4	Изменяем файл	8
3.5	Запускаем файл	8
3.6	Изменяем файл	9
3.7	Проверяем, сошелся ли наш вывод с данным в условии выводом .	9
3.8	Заполняем текст программы	10
3.9	Проверяем разные значения	11
3.10	Создаем файл листинга	11
3.11	Изучаем файл	12
3.12	Удаляем операндум из файла	13
3.13	Транслируем файл	13
3.14	Изучаем файл с ошибкой	14
3.15	Пишем программу	15
3.16	Смотрим на работу программы(всё верно)	15
3.17	Пишем программу	16
3.18	Проверяем работу программы	17
3.19	Проверяем работу программы	17

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Задание

Написать программы для решения системы выражений.

3 Выполнение лабораторной работы

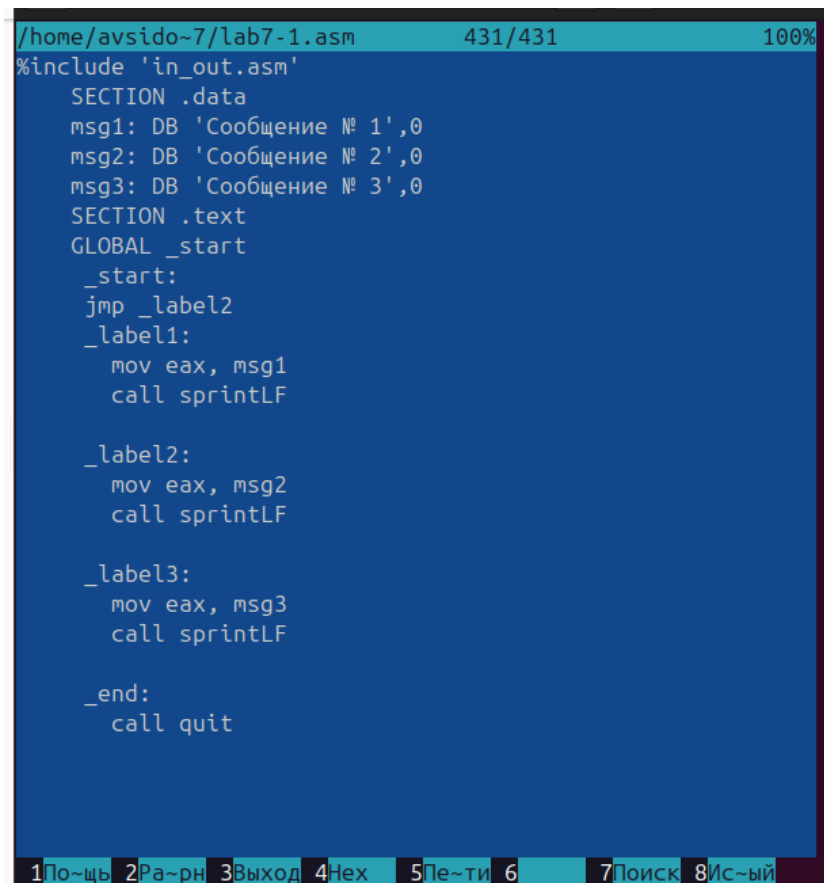
3.1 1. Реализация переходов в NASM

Создадим каталог для программ лабораторной работы № 7, перейдем в него и создадим файл lab7-1.asm(рис. fig. 3.1).

```
avsidorova@avsidorova:~/work/study/2023-2024$ mkdir ~/work/  
arch-pc/lab07  
avsidorova@avsidorova:~/work/study/2023-2024$ cd ~/work/arch-  
pc/lab07  
avsidorova@avsidorova:~/work/arch-pc/lab07$ touch lab7-1.as  
m  
avsidorova@avsidorova:~/work/arch-pc/lab07$ ls  
lab7-1.asm  
avsidorova@avsidorova:~/work/arch-pc/lab07$
```

Рис. 3.1: Создаем каталог и файл

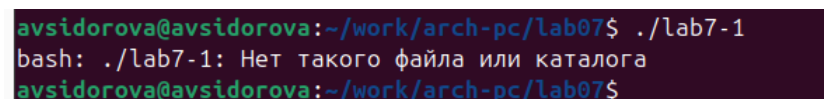
Введите в файл lab7-1.asm текст программы из листинга 7.1. (рис. fig. 3.2).



```
/home/avsidorova~7/lab7-1.asm 431/431 100%
%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1
call sprintf
_label2:
mov eax, msg2
call sprintf
_label3:
mov eax, msg3
call sprintf
_end:
call quit
```

Рис. 3.2: Заполняем файл

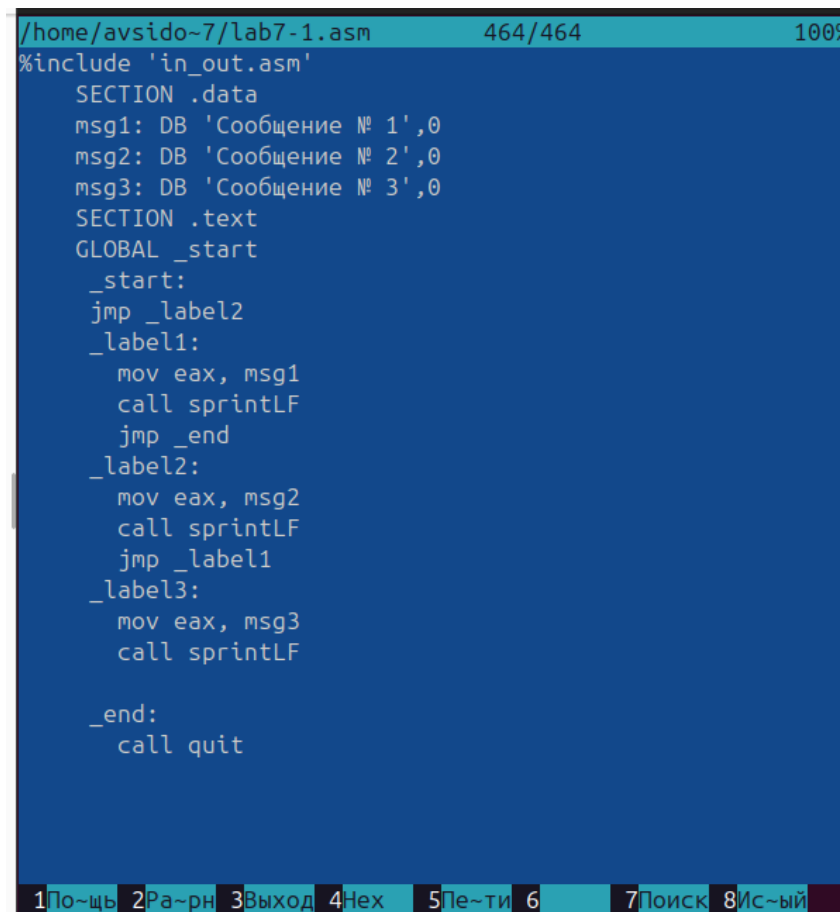
Создаем исполняемый файл и запускаем его (рис. fig. 3.3).



```
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-1
bash: ./lab7-1: Нет такого файла или каталога
avsidorova@avsidorova:~/work/arch-pc/lab07$
```

Рис. 3.3: Запускаем файл

Изменим текст программы в соответствии с листингом 7.2. (рис. fig. 3.4).

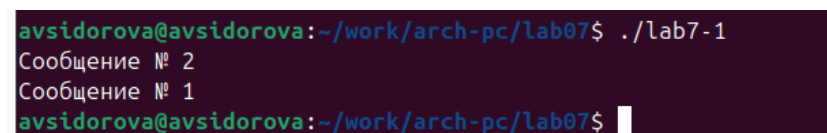


```
/home/avsidorova~7/lab7-1.asm 464/464 100%
#include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1
call sprintLF
jmp _end
_label2:
mov eax, msg2
call sprintLF
jmp _label1
_label3:
mov eax, msg3
call sprintLF

_end:
call quit
```

Рис. 3.4: Изменяем файл

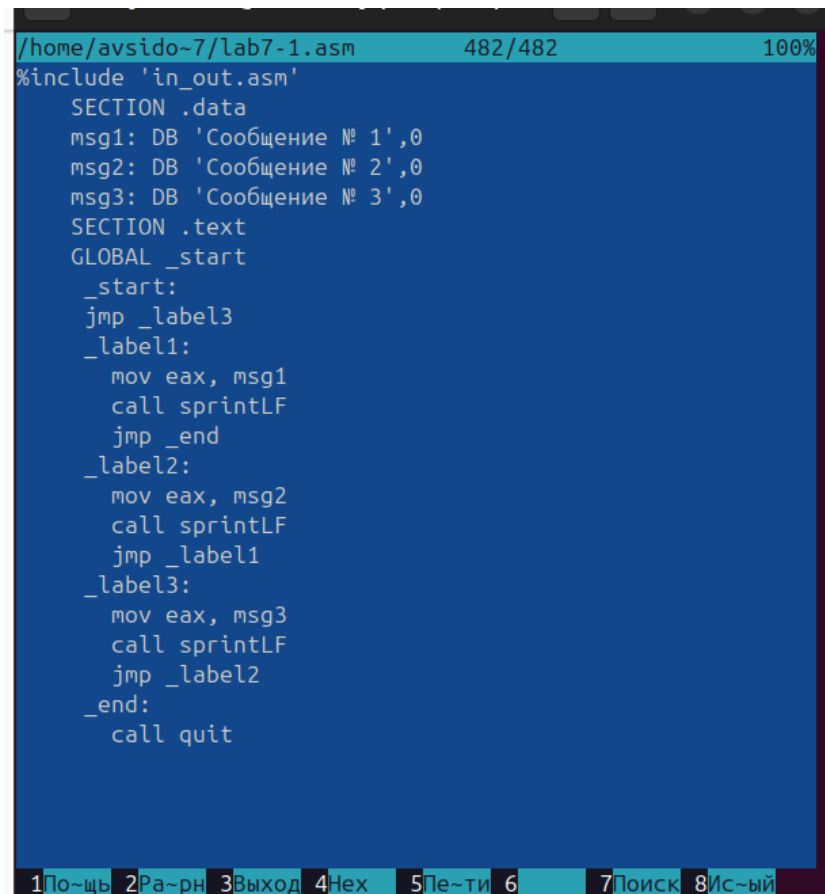
Создаем исполняемый файл и запускаем его (рис. fig. 3.5).



```
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
avsidorova@avsidorova:~/work/arch-pc/lab07$
```

Рис. 3.5: Запускаем файл

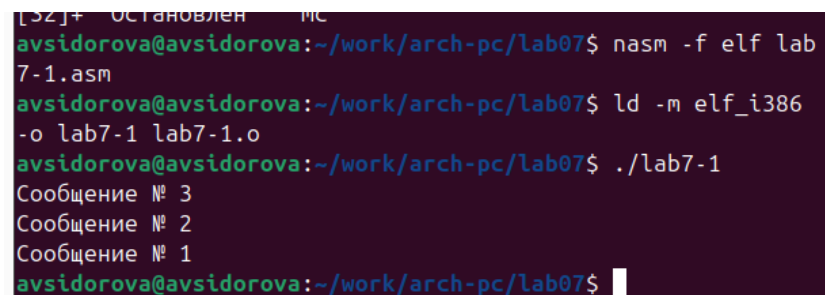
Снова открываем файл для редактирования и изменяем его, чтобы был другой вывод (рис. fig. 3.6).



```
/home/avsidorova~7/lab7-1.asm 482/482 100%
#include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_label1:
mov eax, msg1
call sprintf
jmp _end
_label2:
mov eax, msg2
call sprintf
jmp _label1
_label3:
mov eax, msg3
call sprintf
jmp _label2
_end:
call quit
```

Рис. 3.6: Изменяем файл

Создаем исполняемый файл и запускаем его. Проверяем, сошелся ли наш вывод с данным в условии выводом(рис. fig. 3.7).



```
[32]+ Остановлен
avsidorova@avsidorova:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
avsidorova@avsidorova:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
avsidorova@avsidorova:~/work/arch-pc/lab07$
```

Рис. 3.7: Проверяем, сошелся ли наш вывод с данным в условии выводом

Создаем файл lab7-2 и заполняем программу в соответствии с листингом 7.3 (рис. fig. 3.8).

```
/home/avsidon7/lab7-2.asm 412/560 73%
%include 'in_out.asm'
section .data
    msg1 db ' ',0h
    msg2 db '"',0h
    A dd '20'
    C dd '50'
section .bss
    max resb 10
    B resb 10
section .text
    global _start
_start:
    mov eax,msg1
    call sprint
    mov ecx,B
    mov edx
    call sread
    mov eax,B
    call atoi
    mov [B],eax
    mov ecx,[A]
    mov [max],ecx
check_B:
    mov eax,max
    call atoi
    mov [max],eax
    mov ecx,[max]
```

1По-щъ 2Ра-рн 3Выход 4Hex 5Пе-ти 6 7Поиск 8Ис-ый

Рис. 3.8: Заполняем текст программы

Создаем исполняемый файл и проверяем его работу для разных значений В (рис. fig. 3.9).

```

avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-2
30
30
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-2
10
20
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-2
18
20
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-2
20
20
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./lab7-2
60
60
avsidorova@avsidorova:~/work/arch-pc/lab07$

```

Рис. 3.9: Проверяем разные значения

3.2 Изучение структуры файлы листинга

Создаем файл листинга для программы из файла lab7-2.asm (рис. fig. 3.10).

```

avsidorova@avsidorova:~/work/arch-pc/lab07$ nasm -f elf -l
lab7-1.lst lab7-2.asm

```

Рис. 3.10: Создаем файл листинга

Открываем файл листинга с помощью редактора и изучаем его (рис. fig. 3.11).

```

/home/avsidorova/work/arch-pc/lab07/lab7-2.lst
1                                     %include 'in_out.a>
1                                     <1> ;----->
2                                     <1> ; Функция вычислен>
3                                     <1> slen:>
4 00000000 53                         <1>     push    ebx    >
5 00000001 89C3                       <1>     mov     ebx, e>
6                                     <1>>
7                                     <1> nextchar:>
8 00000003 803800                     <1>     cmp     byte [>
9 00000006 7403                       <1>     jz      finish>
10 00000008 40                        <1>     inc     eax    >
11 00000009 EBF8                      <1>     jmp     nextch>
12                                     <1>>
13                                     <1> finished:>
14 0000000B 29D8                      <1>     sub     eax, e>
15 0000000D 5B                        <1>     pop     ebx    >
16 0000000E C3                       <1>     ret     >
17                                     <1>>
18                                     <1>>
19                                     <1> ;----->
20                                     <1> ; Функция печати с>
21                                     <1> ; входные данные:>
22                                     <1> sprint:>
23 0000000F 52                         <1>     push    edx    >
24 00000010 51                         <1>     push    ecx    >

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить

```

Рис. 3.11: Изучаем файл

Строка 33: 0000001D-адрес в сегменте кода, BB01000000-машинный код, mov ebx,1-присвоение переменной ebx значения 1.

Строка 34: 00000022-адрес в сегменте кода, B804000000-машинный код, mov eax,4-присвоение переменной eax значения 4.

Строка 35 00000027-адрес в сегменте кода, CD80-машинный код, int 80h-вызов ядра.

Открываем файл с программой и удаляем один операндум (рис. fig. 3.12).

```
/home/avsidorova/work/arch-pc/lab07/lab7-2.asm
#include 'in_out.asm'
section .data
    msg1 db '',0h
    msg2 db "",0h
    A dd '20'
    C dd '50'
section .bss
    max resb 10
    B resb 10
section .text
    global _start
_start:
    mov eax,msg1
    call sprint
    mov ecx,B
    mov edx
    call sread
    mov eax,B
    call atoi
    mov [B],eax
    mov ecx,[A]
    mov [max],ecx
check_B:
    mov eax,max
    call atoi
```

[Прочитано 37 строк]

^G Справка	^O Записать	^W Поиск	^K Вырезать
^X Выход	^R ЧитФайл	^_ Замена	^U Вставить

Рис. 3.12: Удаляем операндум из файла

Транслируем с получением файла листинга (рис. fig. 3.13).

```
[40]~$ mv lab7-2.asm lab7-2.o
avsidorova@avsidorova:~/work/arch-pc/lab07$ nasm -f elf -l
lab7-1.lst lab7-2.asm
lab7-2.asm:16: error: invalid combination of opcode and ope
rands
avsidorova@avsidorova:~/work/arch-pc/lab07$ ls
in_out.asm lab7-1.asm lab7-1.o lab7-2.asm
lab7-1 lab7-1.lst lab7-2 lab7-2.lst
avsidorova@avsidorova:~/work/arch-pc/lab07$
```

Рис. 3.13: Транслируем файл

При трансляции файла выдается ошибка, но создаются исполнительные файлы lab7-2 и lab7-2.lst

Снова открываем файл листинга и изучаем его (рис. fig. 3.14).

```

/home/avsidov~7/lab7-2.lst      1006/12705      7%
1                               %include 'in_out.asm'
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
103
```

Рис. 3.14: Изучаем файл с ошибкой

3.3 Задание для самостоятельной работы

ВАРИАНТ-13

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных \boxed{x}, \boxed{y} и z . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.

Создаем новый файл , открываем его и пишем программу, которая выберет наименбшее число из трех(2 числа уже в программе, 3е вводится из консоли) (рис. fig. 3.15).

```

/home/avsidorova/work/arch-pc/lab07/sam1.asm *
%include 'in_out.asm'
section .data
    msg1 db 'Введите B: ',0h
    msg2 db "Наименьшее число: ",0h
    A dd '84'
    C dd '77'
section .bss
    min resb 10
    B resb 10
section .text
    global _start
_start:
    mov eax,msg1
    call sprint
    mov ecx,B
    mov edx,10
    call sread
    mov eax,B
    call atoi
    mov [B],eax
    mov ecx,[A]
    mov [min],ecx
    cmp ecx,[C]
    jl check_B
    mov ecx,[C]

```

^G Справка ^O Записать ^W Поиск ^K Вырезать
 ^X Выход ^R ЧитФайл ^\ Замена ^U Вставить

Рис. 3.15: Пишем программу

Транслируем файл и смотрим на работу программы (рис. fig. 3.16).

```

avsidorova@avsidorova:~/work/arch-pc/lab07$ nasm -f elf sam1.asm
avsidorova@avsidorova:~/work/arch-pc/lab07$ ld -m elf_i386 -o sam1 sam1.o
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./sam1
Введите B: 32
Наименьшее число: 32
avsidorova@avsidorova:~/work/arch-pc/lab07$

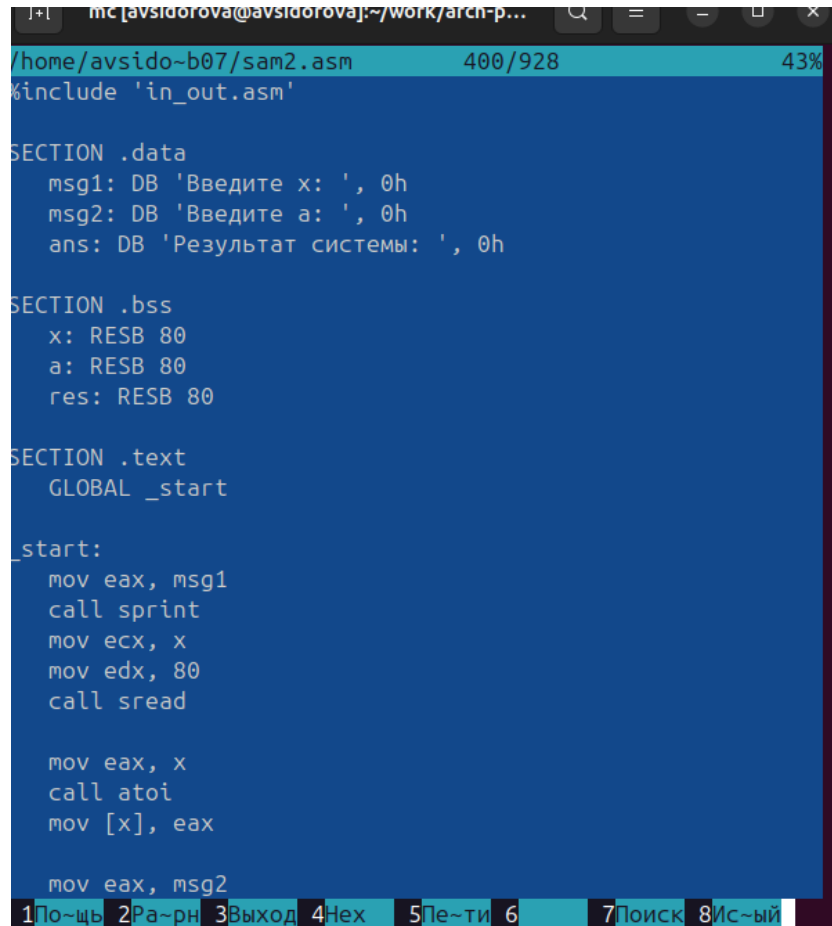
```

Рис. 3.16: Смотрим на работу программы(всё верно)

2. Напишите программу, которая для введенных с клавиатуры значений x и y вычисляет значение заданной функции $f(x, y)$ и выводит результат вычислений. Вид функции $f(x, y)$ выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной ра-

боты № 7. Создайте исполняемый файл и проверьте его работу для значений \times и \times из 7.6.

Создаем новый файл , открываем его и пишем программу, которая решит систему уравнений, при данных, введенных в консоль (рис. fig. 3.17).



```
mc [avsidorova@avsidorova]:~/work/arch-p...
/home/avsidorova/b07/sam2.asm 400/928 43%
#include 'in_out.asm'

SECTION .data
    msg1: DB 'Введите x: ', 0h
    msg2: DB 'Введите a: ', 0h
    ans: DB 'Результат системы: ', 0h

SECTION .bss
    x: RESB 80
    a: RESB 80
    res: RESB 80

SECTION .text
    GLOBAL _start

_start:
    mov eax, msg1
    call sprint
    mov ecx, x
    mov edx, 80
    call sread

    mov eax, x
    call atoi
    mov [x], eax

    mov eax, msg2
```

Рис. 3.17: Пишем программу

Транслируем файл и проверяем его работу при $x=3$ и $a=9$ (рис. fig. 3.18).


```

[47] * Остановлен *
avsidorova@avsidorova:~/work/arch-pc/lab07$ nasm -f elf sam
2.asm
avsidorova@avsidorova:~/work/arch-pc/lab07$ ld -m elf_i386
-o sam2 sam2.o
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./sam2
Введите x: 3
Введите a: 9
Результат системы: 2
avsidorova@avsidorova:~/work/arch-pc/lab07$

```

Рис. 3.18: Проверяем работу программы

Проверяем его работу при $x=6$ и $a=4$ (рис. fig. 3.19).

```

[47] * Остановлен *
avsidorova@avsidorova:~/work/arch-pc/lab07$ ./sam2
Введите x: 6
Введите a: 4
Результат системы: 24
avsidorova@avsidorova:~/work/arch-pc/lab07$

```

Рис. 3.19: Проверяем работу программы

4 Выводы

Изучили команды условного и безусловного переходов. Приобрели навыки написания программ с использованием переходов. Познакомились с назначением и структурой файла листинга.