

ГУАП

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

Н. И. Синев

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

КОСВЕННАЯ АДРЕСАЦИЯ

по курсу:

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ АССЕМБЛЕРА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

1245



01.05.2024

подпись, дата

Г. С. Куранов

инициалы, фамилия

Санкт-Петербург 2024

1. Описание задания

Вариант 9.

19. Переписать из массива A в массив B те элементы массива A, индексы которых совпадают со значениями.

2. Формализация

Создается массив A с некоторыми числами, создается массив B, в которой будут записываться числа из A, если они равны индексам, на которых стоят. Задается размер массива A, чтобы не выйти за границы массива при прохождении через него. Первым проходом проверяется, равно ли число индексу места, на котором это число стоит. Вторым проходом выводятся элементы массива B, в котором -1 – число, значение которого не равно индексу и другие числа, больше или равные 0 (равны индексу)

3. Исходный код

```
.global _main
.align 2

.data
A: .quad 4, 1, 3, 4, 4, 6, 7, 8, 9, 10
B: .quad -1, -1, -1, -1, -1, -1, -1, -1, -1, -1
size: .quad 10

output_str:
.asciz "Output value: %d\n"

.text
_main:
    // загружаем адрес массива A
    adrp X1, A@PAGE
    add X1, X1, A@PAGEOFF

    // загружаем адрес массива B
    adrp X2, B@PAGE
    add X2, X2, B@PAGEOFF

    // загружаем размер массива
    adrp X3, size@PAGE
    add X3, X3, size@PAGEOFF

    ldr x9, [x3]

    // цикл по элементам массива A
    mov x4, 0 // счетчик элементов

loop:
    cmp x4, x9 // сравниваем счетчик с размером массива
    b.ge end // выходим из цикла, если счетчик больше или равен размеру

    // загружаем текущий элемент массива A
    ldr x5, [x1], #8 // смещение X4 * 8 байт

    // проверяем, является ли индекс элемента его значением
    cmp x4, x5
```

```
    add x4, x4, 1
    b.ne loop // переходим к следующему элементу, если индекс не равен
значению
```

```
    //str x5, [sp]
```

```
    //adr x0, output_str //загружаем адрес строки в x0
    //adrp X0, output_str@PAGE
    //add X0, X0, output_str@PAGEOFF
```

```
    //bl _printf //вызываем C-функцию вывода
    // mov x0, #32
```

```
    // копируем текущий элемент массива A в массив B
    str x5, [x2], #8 // смещение X4 * 8 байт
    bl loop
```

```
end:
mov x6, 0
```

```
loop_print:
```

```
adrp X5, B@PAGE
add X5, X5, B@PAGEOFF
```

```
str x6, [sp, #-16]!
```

```
ldr x3, [x5, x6, lsl 3]
```

```
//stp x5, x6, [sp, #-16]
//stp x29, x30, [sp, #-16]
//sub sp, sp, #16
```

```
str x3, [sp, #-16]!
```

```
adrp X0, output_str@PAGE
add X0, X0, output_str@PAGEOFF
```

```
bl _printf //вызываем C-функцию вывода
```

```
//add sp, sp, #16
//ldp x29, x30, [sp], #16
```

```
ldr x6, [sp, #16]
```

```
//stp x29, x30, [sp, #16]
```

```
adrp X3, size@PAGE
add X3, X3, size@PAGEOFF
ldr x9, [x3]
```

```

add x6, x6, #1
//add x5, x5, #8

cmp x9, x6
b.ne loop_print

end_print:
// выводим перевод строки

mov X16, #1
svc #0x80

```

Тестирование

А) ручное

	Входной массив А	Выходной массив В
Тест 1	4, 1, 3, 4, 4, 6, 7, 8, 9, 10	-1, 1, -1, -1, 4, -1, -1, -1, -1, -1
Тест 2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1
Тест 3	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1, 2, 3, 4, 5, 6, 7, 8, 9

1) 4, 1, 3, 4, 4, 6, 7, 8, 9, 10

в данном массиве элементы равны индексу только 1 на 1-ом индексе и 4 на 4-ом
Поэтому -1, 1, -1, -1, 4, -1, -1, -1, -1, -1

2) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Все элементы не равны индексам мест -> -1, -1, -1, -1, -1, -1, -1, -1, -1, -1

3) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Все элементы равны индексам мест -> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Б) С помощью кода

lab_2_singed

lab_3_if

lab_4_adr

lab_5

lab_4

Assembler > lab_4_adr > No Selection

```
1 .global _main
2 .align 2
3
4 .data
5     A: .quad 4, 1, 3, 4, 4, 6, 7, 8, 9, 10
6     B: .quad -1, -1, -1, -1, -1, -1, -1, -1, -1, -1
7     size: .quad 10
8
9 output_str:
10     .asciz "Output value: %d\n"
11
12 .text
13 _main:
14     // загружаем адрес массива A
15     adrp X1, A@PAGE
16     add X1, X1, A@PAGEOFF
17
18     // загружаем адрес массива B
19     adrp X2, B@PAGE
20     add X2, X2, B@PAGEOFF
21
22     // загружаем размер массива
23     adrp X3, size@PAGE
24     add X3, X3, size@PAGEOFF
25
26     ldr x9, [x3]
27
28     // цикл по элементам массива A
29     mov x4, 0 // счетчик элементов
30
31 loop:
32     cmp x4, x9 // сравниваем счетчик с размером массива
33     bge loop_end
34     ldr x0, [A, x4, lsl #3]
35     str x0, [output_str, x4, lsl #1]
36     inc x4
37     b loop
38 loop_end:
39     mov x0, 0
40     mov x1, 0
41     mov x2, 0
42     mov x3, 0
43     mov x4, 0
44     mov x5, 0
45     mov x6, 0
46     mov x7, 0
47     mov x8, 0
48     mov x9, 0
49     mov x10, 0
50     mov x11, 0
51     mov x12, 0
52     mov x13, 0
53     mov x14, 0
54     mov x15, 0
55     mov x16, 0
56     mov x17, 0
57     mov x18, 0
58     mov x19, 0
59     mov x20, 0
60     mov x21, 0
61     mov x22, 0
62     mov x23, 0
63     mov x24, 0
64     mov x25, 0
65     mov x26, 0
66     mov x27, 0
67     mov x28, 0
68     mov x29, 0
69     mov x30, 0
70     mov x31, 0
71     mov x32, 0
72     mov x33, 0
73     mov x34, 0
74     mov x35, 0
75     mov x36, 0
76     mov x37, 0
77     mov x38, 0
78     mov x39, 0
79     mov x40, 0
80     mov x41, 0
81     mov x42, 0
82     mov x43, 0
83     mov x44, 0
84     mov x45, 0
85     mov x46, 0
86     mov x47, 0
87     mov x48, 0
88     mov x49, 0
89     mov x50, 0
90     mov x51, 0
91     mov x52, 0
92     mov x53, 0
93     mov x54, 0
94     mov x55, 0
95     mov x56, 0
96     mov x57, 0
97     mov x58, 0
98     mov x59, 0
99     mov x60, 0
100    mov x61, 0
101    mov x62, 0
102    mov x63, 0
103    mov x64, 0
104    mov x65, 0
105    mov x66, 0
106    mov x67, 0
107    mov x68, 0
108    mov x69, 0
109    mov x70, 0
110    mov x71, 0
111    mov x72, 0
112    mov x73, 0
113    mov x74, 0
114    mov x75, 0
115    mov x76, 0
116    mov x77, 0
117    mov x78, 0
118    mov x79, 0
119    mov x80, 0
120    mov x81, 0
121    mov x82, 0
122    mov x83, 0
123    mov x84, 0
124    mov x85, 0
125    mov x86, 0
126    mov x87, 0
127    mov x88, 0
128    mov x89, 0
129    mov x90, 0
130    mov x91, 0
131    mov x92, 0
132    mov x93, 0
133    mov x94, 0
134    mov x95, 0
135    mov x96, 0
136    mov x97, 0
137    mov x98, 0
138    mov x99, 0
139    mov x100, 0
140    mov x101, 0
141    mov x102, 0
142    mov x103, 0
143    mov x104, 0
144    mov x105, 0
145    mov x106, 0
146    mov x107, 0
147    mov x108, 0
148    mov x109, 0
149    mov x110, 0
150    mov x111, 0
151    mov x112, 0
152    mov x113, 0
153    mov x114, 0
154    mov x115, 0
155    mov x116, 0
156    mov x117, 0
157    mov x118, 0
158    mov x119, 0
159    mov x120, 0
160    mov x121, 0
161    mov x122, 0
162    mov x123, 0
163    mov x124, 0
164    mov x125, 0
165    mov x126, 0
166    mov x127, 0
167    mov x128, 0
168    mov x129, 0
169    mov x130, 0
170    mov x131, 0
171    mov x132, 0
172    mov x133, 0
173    mov x134, 0
174    mov x135, 0
175    mov x136, 0
176    mov x137, 0
177    mov x138, 0
178    mov x139, 0
179    mov x140, 0
180    mov x141, 0
181    mov x142, 0
182    mov x143, 0
183    mov x144, 0
184    mov x145, 0
185    mov x146, 0
186    mov x147, 0
187    mov x148, 0
188    mov x149, 0
189    mov x150, 0
190    mov x151, 0
191    mov x152, 0
192    mov x153, 0
193    mov x154, 0
194    mov x155, 0
195    mov x156, 0
196    mov x157, 0
197    mov x158, 0
198    mov x159, 0
199    mov x160, 0
200    mov x161, 0
201    mov x162, 0
202    mov x163, 0
203    mov x164, 0
204    mov x165, 0
205    mov x166, 0
206    mov x167, 0
207    mov x168, 0
208    mov x169, 0
209    mov x170, 0
210    mov x171, 0
211    mov x172, 0
212    mov x173, 0
213    mov x174, 0
214    mov x175, 0
215    mov x176, 0
216    mov x177, 0
217    mov x178, 0
218    mov x179, 0
219    mov x180, 0
220    mov x181, 0
221    mov x182, 0
222    mov x183, 0
223    mov x184, 0
224    mov x185, 0
225    mov x186, 0
226    mov x187, 0
227    mov x188, 0
228    mov x189, 0
229    mov x190, 0
230    mov x191, 0
231    mov x192, 0
232    mov x193, 0
233    mov x194, 0
234    mov x195, 0
235    mov x196, 0
236    mov x197, 0
237    mov x198, 0
238    mov x199, 0
239    mov x200, 0
240    mov x201, 0
241    mov x202, 0
242    mov x203, 0
243    mov x204, 0
244    mov x205, 0
245    mov x206, 0
246    mov x207, 0
247    mov x208, 0
248    mov x209, 0
249    mov x210, 0
250    mov x211, 0
251    mov x212, 0
252    mov x213, 0
253    mov x214, 0
254    mov x215, 0
255    mov x216, 0
256    mov x217, 0
257    mov x218, 0
258    mov x219, 0
259    mov x220, 0
260    mov x221, 0
261    mov x222, 0
262    mov x223, 0
263    mov x224, 0
264    mov x225, 0
265    mov x226, 0
266    mov x227, 0
267    mov x228, 0
268    mov x229, 0
269    mov x230, 0
270    mov x231, 0
271    mov x232, 0
272    mov x233, 0
273    mov x234, 0
274    mov x235, 0
275    mov x236, 0
276    mov x237, 0
277    mov x238, 0
278    mov x239, 0
279    mov x240, 0
280    mov x241, 0
281    mov x242, 0
282    mov x243, 0
283    mov x244, 0
284    mov x245, 0
285    mov x246, 0
286    mov x247, 0
287    mov x248, 0
288    mov x249, 0
289    mov x250, 0
290    mov x251, 0
291    mov x252, 0
292    mov x253, 0
293    mov x254, 0
294    mov x255, 0
295    mov x256, 0
296    mov x257, 0
297    mov x258, 0
298    mov x259, 0
299    mov x260, 0
300    mov x261, 0
301    mov x262, 0
302    mov x263, 0
303    mov x264, 0
304    mov x265, 0
305    mov x266, 0
306    mov x267, 0
307    mov x268, 0
308    mov x269, 0
309    mov x270, 0
310    mov x271, 0
311    mov x272, 0
312    mov x273, 0
313    mov x274, 0
314    mov x275, 0
315    mov x276, 0
316    mov x277, 0
317    mov x278, 0
318    mov x279, 0
319    mov x280, 0
320    mov x281, 0
321    mov x282, 0
322    mov x283, 0
323    mov x284, 0
324    mov x285, 0
325    mov x286, 0
326    mov x287, 0
327    mov x288, 0
328    mov x289, 0

```

Рисунок 1 – результат работы программы с массивом $A = \{ \}$

lab_2_singedlab_3_iflab_4_adrlab_5lab_4

Assembler > lab_4_adr > No Selection

```
1 .global _main
2 .align 2
3
4 .data
5 A: .quad 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
6 B: .quad -1, -1, -1, -1, -1, -1, -1, -1, -1, -1
7 size: .quad 10
8
9 output_str:
10 .asciz "Output value: %d\n"
11
12 .text
13 _main:
14 // загружаем адрес массива A
15 adrp X1, A@PAGE
16 add X1, X1, A@PAGEOFF
17
18 // загружаем адрес массива B
19 adrp X2, B@PAGE
20 add X2, X2, B@PAGEOFF
21
22 // загружаем размер массива
23 adrp X3, size@PAGE
24 add X3, X3, size@PAGEOFF
25
26 ldr x9, [x3]
27
28 // цикл по элементам массива A
29 mov x4, 0 // счетчик элементов
30
31 loop:
32 cmp x4, x9 // сравниваем счетчик с размером массива
33
34 // выводим значение элемента
35 ldr x10, [x1, x4, lsl #3]
36 mov x10, x10, ror #31
37 str x10, [x2, x4, lsl #3]
38
39 // увеличиваем счетчик
40 add x4, x4, #1
41
42 // выходим из цикла
43 cmp x4, x9
44 bne loop
45
46 // выходим из программы
47 mov x0, #17
48 svc #0
```

Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Output value: -1
Program ended with exit code: 17

Рисунок 2 – результат работы программы с числом 69

