**ЛАБОРАТОРНАЯ РАБОТА №4**

**ВВЕДЕНИЕ В АРХИТЕКТУРУ ARM**

**Цели работы:**

1. Знакомство с архитектурой ARM
2. Анализ ассемблерного листинга программы для архитектуры ARM

Листинг программы на Си:

1. #include <stdlib.h>
2. void swap(int \*a, int \*b) {
3. int tmp = \*a;
4. \*a = \*b;
5. \*b = tmp;
6. }
7. int\* FillArray(long long n) {
8. int \*array = calloc(n, sizeof(int));
9. for (int i = 0; i < n; i++) {
10. array[i] = rand() % 1000;
11. }
12. return array;
13. }
14. int main() {
15. long long n = 140000;
16. int \*array = FillArray(n);
17. for (int i = 0; i < n; i++) {
18. for (int j = 0 ; j < n - 1; j++) {
19. if (array[j] > array[j + 1]) {
20. swap(&array[j], &array[j + 1]);
21. }
22. }
23. }
24. return 0;
25. }

Листинг программы на ассемблере (уровень O0):

1. swap:
2. sub sp, sp, #32
3. str x0, [sp, 8]
4. str x1, [sp]
5. ldr x0, [sp, 8]
6. ldr w0, [x0]
7. str w0, [sp, 28]
8. ldr x0, [sp]
9. ldr w1, [x0]
10. ldr x0, [sp, 8]
11. str w1, [x0]
12. ldr x0, [sp]
13. ldr w1, [sp, 28]
14. str w1, [x0]
15. nop
16. add sp, sp, 32
17. ret
18. FillArray:
19. stp x29, x30, [sp, -48]!
20. mov x29, sp
21. str x0, [sp, 24]
22. ldr x0, [sp, 24]
23. mov x1, 4
24. bl calloc
25. str x0, [sp, 32]
26. str wzr, [sp, 44]
27. b .L3
28. .L4:
29. bl rand
30. mov w1, w0
31. ldrsw x0, [sp, 44]
32. lsl x0, x0, 2
33. ldr x2, [sp, 32]
34. add x2, x2, x0
35. mov w0, 19923
36. movk w0, 0x1062, lsl 16
37. smull x0, w1, w0
38. lsr x0, x0, 32
39. asr w3, w0, 6
40. asr w0, w1, 31
41. sub w0, w3, w0
42. mov w3, 1000
43. mul w0, w0, w3
44. sub w0, w1, w0
45. str w0, [x2]
46. ldr w0, [sp, 44]
47. add w0, w0, 1
48. str w0, [sp, 44]
49. .L3:
50. ldrsw x0, [sp, 44]
51. ldr x1, [sp, 24]
52. cmp x1, x0
53. bgt .L4
54. ldr x0, [sp, 32]
55. ldp x29, x30, [sp], 48
56. ret
57. main:
58. stp x29, x30, [sp, -48]!
59. mov x29, sp
60. mov x0, 8928
61. movk x0, 0x2, lsl 16
62. str x0, [sp, 32]
63. ldr x0, [sp, 32]
64. bl FillArray
65. str x0, [sp, 24]
66. str wzr, [sp, 44]
67. b .L7
68. .L11:
69. str wzr, [sp, 40]
70. b .L8
71. .L10:
72. ldrsw x0, [sp, 40]
73. lsl x0, x0, 2
74. ldr x1, [sp, 24]
75. add x0, x1, x0
76. ldr w1, [x0]
77. ldrsw x0, [sp, 40]
78. add x0, x0, 1
79. lsl x0, x0, 2
80. ldr x2, [sp, 24]
81. add x0, x2, x0
82. ldr w0, [x0]
83. cmp w1, w0
84. ble .L9
85. ldrsw x0, [sp, 40]
86. lsl x0, x0, 2
87. ldr x1, [sp, 24]
88. add x2, x1, x0
89. ldrsw x0, [sp, 40]
90. add x0, x0, 1
91. lsl x0, x0, 2
92. ldr x1, [sp, 24]
93. add x0, x1, x0
94. mov x1, x0
95. mov x0, x2
96. bl swap
97. .L9:
98. ldr w0, [sp, 40]
99. add w0, w0, 1
100. str w0, [sp, 40]
101. .L8:
102. ldrsw x1, [sp, 40]
103. ldr x0, [sp, 32]
104. sub x0, x0, #1
105. cmp x1, x0
106. blt .L10
107. ldr w0, [sp, 44]
108. add w0, w0, 1
109. str w0, [sp, 44]
110. .L7:
111. ldrsw x0, [sp, 44]
112. ldr x1, [sp, 32]
113. cmp x1, x0
114. bgt .L11
115. mov w0, 0
116. ldp x29, x30, [sp], 48
117. ret

Листинг программы на ассемблере (уровень O2):

1. swap:
2. ldr w3, [x1]
3. ldr w2, [x0]
4. str w3, [x0]
5. str w2, [x1]
6. ret
7. FillArray:
8. stp x29, x30, [sp, -64]!
9. mov x1, 4
10. mov x29, sp
11. stp x19, x20, [sp, 16]
12. mov x20, x0
13. bl calloc
14. mov x19, x0
15. cmp x20, 0
16. ble .L3
17. stp x21, x22, [sp, 32]
18. mov w22, 19923
19. movk w22, 0x1062, lsl 16
20. mov w21, 1000
21. str x23, [sp, 48]
22. mov x23, 0
23. .L5:
24. bl rand
25. smull x1, w0, w22
26. asr x1, x1, 38
27. sub w1, w1, w0, asr 31
28. msub w0, w1, w21, w0
29. str w0, [x19, x23, lsl 2]
30. add x23, x23, 1
31. cmp x20, x23
32. bne .L5
33. ldp x21, x22, [sp, 32]
34. ldr x23, [sp, 48]
35. .L3:
36. mov x0, x19
37. ldp x19, x20, [sp, 16]
38. ldp x29, x30, [sp], 64
39. ret
40. main:
41. stp x29, x30, [sp, -16]!
42. mov x0, 8928
43. movk x0, 0x2, lsl 16
44. mov x29, sp
45. bl FillArray
46. add x3, x0, 557056
47. mov w4, 8928
48. mov x5, x0
49. add x3, x3, 2940
50. movk w4, 0x2, lsl 16
51. .L9:
52. mov x0, x5
53. .L11:
54. ldp w1, w2, [x0]
55. cmp w1, w2
56. ble .L10
57. stp w2, w1, [x0]
58. .L10:
59. add x0, x0, 4
60. cmp x0, x3
61. bne .L11
62. subs w4, w4, #1
63. bne .L9
64. mov w0, 0
65. ldp x29, x30, [sp], 16
66. ret

**Вывод:**

По результатам выполнения лабораторной работы мы ознакомились с программной архитектурой ARM, научились анализировать ассемблерные листинги программ на данной архитектуре.