**Лабораторная работа 7.**

Задание 1.

Рассмотрим массив A элементов типа Type

1. sizeof(Type), (sizeof(int) = 4 (у меня, но он может быть и другой, зависит от компилятора и платформы), sizeof(char) = 1)
2. Адреса соседних элементов отличаются на sizeof(Type)
3. Одинаково
4. sizeof(Type) \* кол-во элементов массива
5. Назовем массив array, требуемы элемент N

Адрес элемента в массиве вычисляется как =

&array[0] + sizeof(Type) \* N

Задание 2.

int main() {

    int N;

    scanf("%d", &N);

    int\* data = (int\*) malloc(sizeof(int) \* N);

    for (int i = 0; i < N; ++i)

        data[i] = rand() % 50;

    printf("array values is: \n");

    for (int i = 0; i < N; ++i)

        printf("%d: %d\n", i, data[i]);

    printf("sum of array is %d\n", summ(N, data));

free(data);

return 0;

}

.text

.globl sum

.type sum, @function

sum:

testl %edi, %edi

jle returnifzero

movq %rsi, %rdx

leal -1(%rdi), %eax

leaq 4(%rsi,%rax,4), %rcx

movl $0, %eax

for:

addl (%rdx), %eax

addq $4, %rdx

cmpq %rcx, %rdx

jne for

rep ret

returnifzero:

movl $0, %eax

ret

10

array values is:

0: 33

1: 36

2: 27

3: 15

4: 43

5: 35

6: 36

7: 42

8: 49

9: 21

sum of array is 337

2

array values is:

0: 33

1: 36

sum of array is 69

Задние 3.

Рассмотрим структуру KeyValue;

struct KeyValue {

int key;

double value;

};

sizeof(KeyValue) = 16, хотя sizeof(int) + sizeof(double) == 12;

Это происходит из-за выравнивания

1. sizeof(KeyValue) = 16
2. Адреса соседних элементов отличаются на

sizeof(int) = 4

sizeof(double) = 8

1. alignof(KeyValue) = 8
2. sizeof(KeyValue) \* кол-во элементов массива
3. Назовем массив array, требуемы элемент N, поле field

Адрес элемента в массиве вычисляется как =

&array[0] + sizeof(KeyValue) \* N + &array[0].field - &array[0] =

&array[0].field + sizeof(KeyValue) \* N

Задание 4.

struct KeyValue {

    int key;

    double value;

};

extern "C" bool sum(int key, int size, KeyValue\* data, double\* result);

int main() {

    srand(time(NULL));

    int N, k;

    scanf("%d %d", &N, &k);

    KeyValue\* data = (KeyValue\*) malloc(sizeof(KeyValue) \* N);

    for (int i = 0; i < N; ++i) {

        data[i].key = rand() % 20;

        data[i].value = rand() % 100;

    }

    for (int i = 0; i < N; ++i) {

        printf("%d: %le\n", data[i].key, data[i].value);

    }

    double result;

    if (!sum(k, N, data, &result)) {

        printf("Elements with key %d didn't exist\n", k);

    } else {

        printf("Sum of elemtnst is: %le\n", result);

    }

free(data);

return 0;

}

.text

.globl sum

.type sum, @function

sum:

movq $0x000000000, (%rcx)

testl %esi, %esi

jle returnifzero

movq %rdx, %r8

leal -1(%rsi), %eax

salq $4, %rax

leaq 16(%rdx,%rax), %rdx

movl $0, %eax

movl $1, %esi

jmp forbody

forhead:

addq $16, %r8

cmpq %rdx, %r8

je forend

forbody:

cmpl %edi, (%r8)

jne forhead

movsd (%rcx), %xmm0

addsd 8(%r8), %xmm0

movsd %xmm0, (%rcx)

movl %esi, %eax

jmp forhead

forend:

rep ret

returnifzero:

movl $0, %eax

ret

10 0

3: 8.600000e+01

17: 1.500000e+01

13: 3.500000e+01

6: 9.200000e+01

9: 2.100000e+01

2: 2.700000e+01

10: 5.900000e+01

3: 2.600000e+01

0: 2.600000e+01

12: 3.600000e+01

Sum of elemtnst is: 2.600000e+01

10 5

3: 8.600000e+01

17: 1.500000e+01

13: 3.500000e+01

6: 9.200000e+01

9: 2.100000e+01

2: 2.700000e+01

10: 5.900000e+01

3: 2.600000e+01

0: 2.600000e+01

12: 3.600000e+01

Elements with key 5 didn't exist

Задание 5.

#include <stdio.h>

#include <math.h>

extern "C" void fib(int N, int\* result);

int main() {

    int x;

    scanf("%d", &x);

    int\* fibs = (int\*) malloc((x + 1) \* sizeof(int));

    fib(x, fibs);

    for (int i = 0; i < (x + 1); ++i) {

        printf("fib(%d) = %d\n", i, fibs[i]);

    }

    free(fibs);

return 0;

}

.text

.globl fib

.type fib, @function

fib:

movl $1, (%rsi)

movl $1, 4(%rsi)

leal 1(%rdi), %eax

cmpl $2, %eax

jle end

movq %rsi, %rax

leal -2(%rdi), %edx

leaq 4(%rsi,%rdx,4), %rcx

for:

movl (%rax), %edx

addl 4(%rax), %edx

movl %edx, 8(%rax)

addq $4, %rax

cmpq %rcx, %rax

jne for

end:

rep ret

6

fib(0) = 1

fib(1) = 1

fib(2) = 2

fib(3) = 3

fib(4) = 5

fib(5) = 8

fib(6) = 13

Задание 6.

#include <stdio.h>

#include <math.h>

#include <time.h>

struct KeyValue {

    int key;

    double value;

};

extern "C" void form(int key, int size, KeyValue\* data);

int main() {

    srand(time(NULL));

    int N, k;

    scanf("%d %d", &N, &k);

    KeyValue\* data = (KeyValue\*) malloc(sizeof(KeyValue) \* N);

    for (int i = 0; i < N; ++i) {

        data[i].key = rand() % 20;

        data[i].value = rand() % 100;

    }

    for (int i = 0; i < N; ++i) {

        printf("%d: %le\n", data[i].key, data[i].value);

    }

    form(k, N, data);

    printf("\n\n");

    for (int i = 0; i < N; ++i) {

        printf("%d: %le\n", data[i].key, data[i].value);

    }

    free(data);

    return 0;

}

.text

.globl form

.type form, @function

form:

testl %esi, %esi

jle returnifzero

movq %rdx, %rcx

leal -1(%rsi), %eax

salq $4, %rax

leaq 16(%rdx,%rax), %r8

movl $0, %eax

counter:

cmpl %edi, (%rdx)

sete %sil

movzbl %sil, %esi

addl %esi, %eax

addq $16, %rdx

cmpq %r8, %rdx

jne counter

testl %eax, %eax

je returnifzero

pxor %xmm0, %xmm0

cvtsi2sd %eax, %xmm0

movsd one(%rip), %xmm1

divsd %xmm0, %xmm1

jmp forbody

for:

addq $16, %rcx

cmpq %rdx, %rcx

je returnifzero

forbody:

cmpl %edi, (%rcx)

jne for

movsd %xmm1, 8(%rcx)

jmp for

returnifzero:

rep ret

one:

.long 0

.long 1072693248