

Правительство Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н. Тихонова

ОТЧЕТ
О ПРАКТИЧЕСКОЙ РАБОТЕ № 3
по дисциплине «Системное программирование»
КОМБИНИРОВАНИЕ РАЗНОЯЗЫКОВЫХ МОДУЛЕЙ

Студент гр. БИБ 211

_____ В.Ф. Санников

«___» _____ 2023 г.

Руководитель

Старший преподаватель

_____ В.И. Морозов

«___» _____ 2023 г.

Москва

СОДЕРЖАНИЕ

1 Задание на практическую работу.....	3
2 Ход работы.....	4
2.1 Программная реализация на Assembly MASM x64.....	4
2.2 Программная реализация на языке C.....	7
2.3 Примеры выполнения исполняемого файла в IDE Visual Studio 2022..	9
3 Выводы о проделанной работе.....	11
ПРИЛОЖЕНИЕ А.....	11
ПРИЛОЖЕНИЕ Б.....	13

1 Задание на практическую работу

Целью работы является изучение языка описания процессорных инструкций “Ассемблер” и языка С, а также написание программной реализации, которая удовлетворяет следующему заданию: “Напишите программу, в которой создается одномерный числовой массив. После заполнения значениями (случайными числами) массива в нем нужно выполнить циклическую перестановку элементов. Количество позиций для циклической перестановки вводится пользователем с клавиатуры.”

Для выполнения работы необходимо:

- 1) Повторно ознакомиться с теоретическим материалом семинара 4. Внимание! Все настройки, действия и манипуляции, проводимые для выполнения заданий из последующих пунктов, необходимо фиксировать в отчёте в виде снимков экрана или листингов вводимых команд.
- 2) С помощью языков программирования С и Ассемблер, используя любой синтаксис (Intel или AT&T) на свой выбор, необходимо написать программу, решающую задачу из варианта.

а) Входные данные для задач, основная часть которых подразумевает обработку чисел, а не их создание, должны генерироваться автоматически с помощью ГПСЧ языка Си. Сгенерированные исходные данные должны быть выведены в консоль.

б) При случайной генерации чисел пользователю должна предоставляться возможность ограничить диапазон значений этих чисел.

в) Решением задачи является вывод в консоль верного результата при любых допустимых входных данных.

г) Программа должна корректно завершаться, не вызывая аварийный останов.

д) В программе на языке С должен быть реализован только следующий функционал: выделение памяти под исходные данные (и их инициализация в случаях, когда она должна быть произведена с помощью ГПСЧ, или ввод данных), вызов подпрограммы, 2 выполняющей основные вычисления, вывод результата вычислений в консоль. Все основные вычисления должны быть оформлены в виде ассемблерного модуля (не ассемблерной вставки в код на Си, а именно отдельного файла на языке Assembler).

е) Для выполнения повторяющихся однотипных действий необходимо использовать соответствующие средства языка. Копирование и вставка одного и того же кода нужное количество раз с незначительными изменениями без необходимости является ошибкой.

ж) Все программы должны работать с любыми размерами входных данных. Размер входных данных (даже если сами данные генерируются случайно) должен задаваться пользователем посредством ввода с командной строки. Запрещается ограничивать максимальный размер входных данных.

2 Ход работы

Для выполнения данной лабораторной работы необходимо выбрать задание по варианту. Мой номер в группе = 19, соответственно рассмотрим 19 вариант:

“Напишите программу, в которой создается одномерный числовой массив. После заполнения значениями (случайными числами) массива в нем нужно выполнить циклическую перестановку элементов. Количество позиций для циклической перестановки вводится пользователем с клавиатуры.”

2.1 Программная реализация на Assembly MASM x64

```
CODE
f PROC
    dec rdx
zxc:
    xor ebx, ebx
    xor rsi, rsi
    mov eax,[r8]
shifter:
    inc rsi
    mov ebx, [r8+(rsi)*4]
    dec rsi
    mov [r8+(rsi)*4], ebx
    inc rsi
    cmp rsi, rdx
```

```
jl shifter
mov [r8+(rsi)*4],eax
loop zxc
ret
f ENDP
END
```

Листинг 1 – Программная реализация на языке ассемблер.

Давай разберем его по частям.

1. Сигнатура процедуры:

```
f PROC
```

2. Уменьшение значения регистра rdx на 1:

```
dec rdx
```

3. Метка zxc:

```
zxc:
```

4. Обнуление регистров ebx и rsi:

```
xor ebx, ebx
```

```
xor rsi, rsi
```

5. Загрузка значения, на которое указывает r8, в регистр eax:

```
mov eax, [r8]
```

6. Метка shifter:

```
shifter:
```

7. Инкремент регистра rsi:

```
inc rsi
```

8. Загрузка значения, на которое указывает r8 с учетом смещения $((rsi)*4)$, в регистр ebx:

```
mov ebx, [r8+(rsi)*4]
```

9. Декремент регистра rsi:

```
dec rsi
```

10. Запись значения из регистра ebx по адресу, на который указывает r8 с учетом смещения $((rsi)*4)$:

```
mov [r8+(rsi)*4], ebx
```

11. Инкремент регистра rsi:

```
inc rsi
```

12. Сравнение регистра rsi с rdx, после чего выполнение перехода к метке shifter, если rsi меньше rdx:

```
cmp rsi, rdx
```

```
jl shifter
```

13. Запись значения из регистра eax по адресу, на который указывает r8 с учетом смещения $((rsi)*4)$:

```
mov [r8+(rsi)*4], eax
```

14. Операция loop с меткой zxc, которая имеет то же самое действие, что и инструкция dec rdx:

```
loop zxc
```

15. Завершение процедуры:

```
ret
```

Данный код принимает на вход массив и выполняет его циклический сдвиг на заданное количество позиций.

2.2 Программная реализация на языке C

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

extern void f(int32_t shift, int32_t len, int32_t* mass);

int main() {
    SetConsoleOutputCP(1251);
    int32_t n;
    int32_t shift;
    printf("Введите размер массива\n");
#pragma warning(suppress : 4996)
    scanf("%d", &n);
    if (n <= 0) {
        printf("длина меньше 0");
        return 1;
    }
    printf("Введите количество позиций для циклического сдвига\n");
#pragma warning(suppress : 4996)
    scanf("%d", &shift);
    int* mass = (int32_t*)malloc(n * sizeof(int32_t));
    for (int i = 0; i < n; i++) {
        mass[i] = rand() % 1000;
    }
    printf("Сгенерированный массив\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", mass[i]);
    }
    f(shift, n, mass);
    printf("\n");
    printf("Итоговый массив\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", mass[i]);
    }
    free(mass);
}
```

Листинг 2 – Программная реализация на языке C.

Этот код на C имеет довольно явную структуру, и его функциональность заключается в работе с массивом, выполняя циклический сдвиг элементов массива на заданное количество позиций. Давай разберем пошагово.

1. Сначала код включает несколько заголовочных файлов:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
```

В первых двух заголовочных файлах содержатся общие функции для работы с вводом/выводом и выделением памяти. Заголовочный файл `stdint.h` содержит определения для фиксированных ширин целых чисел.

2. Объявляется неявное внешнее ссылочное объявление на функцию `f`, которая написана на языке ассемблера, и должна принимать три аргумента:

```
extern void f(int32_t shift, int32_t len, int32_t* mass);
```

3. В функции `main` идет инициализация переменных и ввод данных с клавиатуры:

```
int main() {
    SetConsoleOutputCP(1251); // Это вызов функции, которая, вероятно, устанавливает
    кодировку вывода консоли
    int32_t n;
    int32_t shift;
    printf("Введите размер массива\n");
    scanf("%d", &n);
    // ... (проверка n и выделение памяти для массива)
    printf("Введите количество позиций для циклического сдвига\n");
    scanf("%d", &shift);
    // ... (генерация случайных чисел, вывод массива)
    // ... (вызов функции f для выполнения циклического сдвига)
    // ... (вывод итогового массива)
    // ... (освобождение памяти)
}
```

4. Затем пользователю предлагается ввести размер массива и количество позиций для циклического сдвига. Если введенный размер массива отрицателен или равен нулю, то программа выведет сообщение об ошибке и завершит свою работу.

5. Затем программа выделяет память для массива размером `n` целочисленных элементов.

6. Создается массив, заполняемый случайными числами от 0 до 999:

```
for (int i = 0; i < n; i++) {
    mass[i] = rand() % 1000;
}
```

7. Печатается сгенерированный массив.

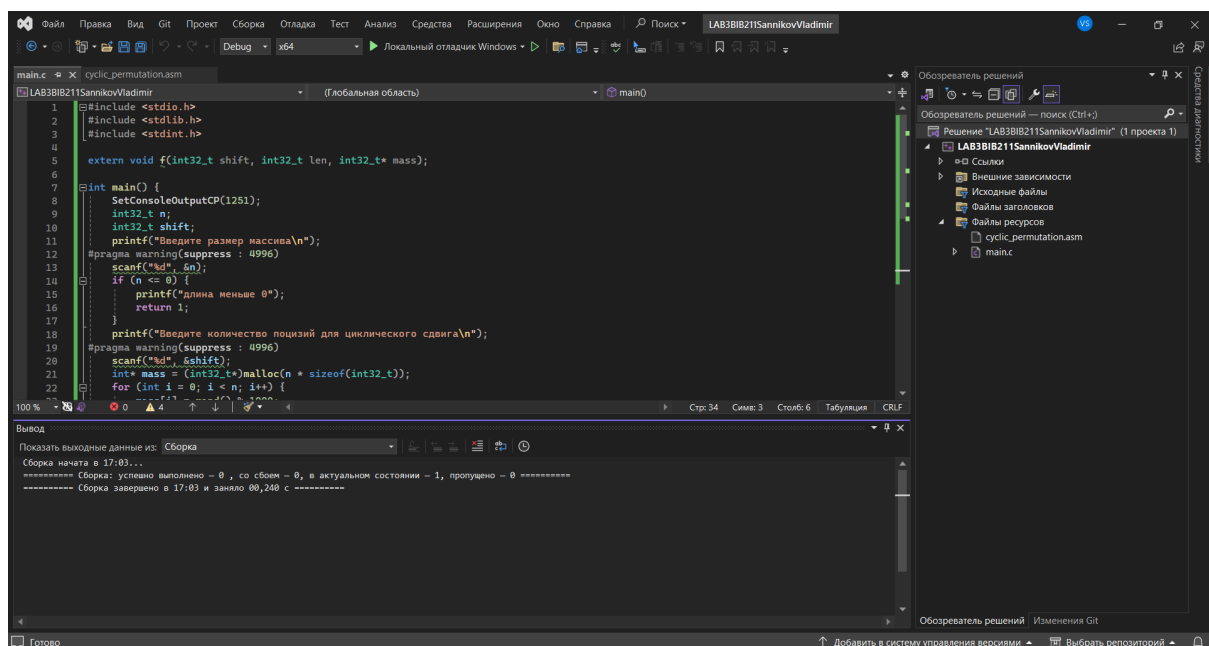
8. Вызывается функция `f`, которая выполняет циклический сдвиг элементов массива на заданное количество позиций.

9. Печатается итоговый массив.

10. Память, выделенная для массива, освобождается с помощью функции `free()`.

Таким образом, этот код позволяет пользователю задать размер массива, заполнить его случайными числами, выполнить циклический сдвиг элементов массива на указанное количество позиций и затем вывести итоговый результат.

2.3 Примеры выполнения исполняемого файла в IDE Visual Studio 2022



Скриншот 1 – Среда разработки.

Как видно на Скриншот 1, сборка проекта произошла успешно, также можно обратить внимание на файловую структуру проекта. Теперь прогоним различные тестовые данные через программу и посмотрим на вывод.

```
Консоль отладки Microsoft V x + v
Введите размер массива
10
Введите количество позиций для циклического сдвига
5
Сгенерированный массив
41 467 334 500 169 724 478 358 962 464
Итоговый массив
724 478 358 962 464 41 467 334 500 169
C:\Users\vladi\source\repos\LAB3BIB211SannikovVladimir\
работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отл
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Скриншот 2 – Небольшие входные данные.

```
Консоль отладки Microsoft V x + v - □ x
Введите размер массива
100
Введите количество позиций для циклического сдвига
250
Сгенерированный массив
41 467 334 500 169 724 478 358 962 464 705 145 281 827 961 491 995 942 827 436 391 604 902 153 292 382 421 716 718 895 4
47 726 771 538 869 912 667 299 35 894 703 811 322 333 673 664 141 711 253 868 547 644 662 757 37 859 723 741 529 778 316
35 190 842 288 106 40 942 264 648 446 805 890 729 370 350 6 101 393 548 629 623 84 954 756 840 966 376 931 308 944 439
626 323 537 538 118 82 929 541
Итоговый массив
547 644 662 757 37 859 723 741 529 778 316 35 190 842 288 106 40 942 264 648 446 805 890 729 370 350 6 101 393 548 629 6
23 84 954 756 840 966 376 931 308 944 439 626 323 537 538 118 82 929 541 41 467 334 500 169 724 478 358 962 464 705 145
281 827 961 491 995 942 827 436 391 604 902 153 292 382 421 716 718 895 447 726 771 538 869 912 667 299 35 894 703 811 3
22 333 673 664 141 711 253 868
C:\Users\vladi\source\repos\LAB3BIB211SannikovVladimir\x64\Debug\LAB3BIB211SannikovVladimir.exe (процесс 29636) завершил
работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Скриншот 3 – Входные данные среднего размера.

```
Консоль отладки Microsoft \ x
Введите размер массива
500
Введите количество позиций для циклического сдвига
2
Сгенерированный массив
41 467 334 500 169 724 478 358 962 464 705 145 281 827 961 491 995 942 827 436 391 604 902 153 292 382 421 716 718 895 447 726 771 538 869 912 667 299 35 89
4 703 811 322 333 673 664 141 711 253 868 547 644 662 757 37 859 723 741 529 778 316 35 190 842 288 106 40 942 264 648 446 805 890 729 370 350 6 101 393 548
629 623 84 954 756 840 966 376 931 308 944 439 626 323 537 538 118 82 929 541 833 115 639 658 704 930 977 306 673 386 21 745 924 72 270 829 777 573 97 512
986 290 161 636 355 767 655 574 31 52 350 150 941 724 966 439 107 191 7 337 457 287 753 383 945 989 209 758 221 588 422 946 506 30 413 168 900 591 762 655 4
10 359 624 537 548 483 595 41 602 350 291 836 374 20 596 21 348 190 660 404 281 734 53 999 412 938 900 788 127 467 728 893 648 483 807 421 310 617 813 514 3
09 616 935 451 600 249 519 556 798 303 224 8 844 609 089 702 195 485 93 243 523 587 314 503 448 200 458 618 580 796 798 281 589 798 9 157 472 622 538 292 30
179 190 657 958 191 815 888 156 511 202 634 272 55 328 646 362 886 875 433 869 142 844 416 881 998 322 651 21 699 557 476 892 389 75 712 600 510 3 869 861
688 401 789 255 423 2 585 182 285 88 426 617 757 832 932 169 154 721 189 976 329 368 692 425 555 434 549 441 512 145 60 718 753 139 423 279 996 687 529 549
437 866 949 193 195 297 416 286 105 488 282 455 734 114 701 316 671 786 263 313 355 185 53 912 808 832 945 313 756 321 558 646 982 481 144 196 222 129 161 5
35 450 173 466 44 659 292 439 253 24 154 510 745 649 186 313 474 22 168 18 787 905 958 391 202 625 477 414 314 824 334 874 372 159 833 70 487 297 518 177 77
3 270 763 668 192 985 102 480 213 627 802 99 527 625 543 924 23 972 61 181 3 432 505 593 725 31 492 142 222 286 64 900 187 360 413 974 270 170 235 833 711 7
60 896 667 285 550 140 694 695 624 19 125 576 694 658 302 371 466 678 593 851 484 18 464 119 152 800 87 60 926 10 757 170 315 576 227 43 758 164 109 882 86
565 487 577 474 625 627 629 928 423 520 902 962 123 596 737 261 195 525
Итоговый массив
334 500 169 724 478 358 962 464 705 145 281 827 961 491 995 942 827 436 391 604 902 153 292 382 421 716 718 895 447 726 771 538 869 912 667 299 35 894 703 8
11 322 333 673 664 141 711 253 868 547 644 662 757 37 859 723 741 529 778 316 35 190 842 288 106 40 942 264 648 446 805 890 729 370 350 6 101 393 548 629 62
3 84 954 756 840 966 376 931 308 944 439 626 323 537 538 118 82 929 541 833 115 639 658 704 930 977 306 673 386 21 745 924 72 270 829 777 573 97 512 986 290
161 636 355 767 655 574 31 52 350 150 941 724 966 430 107 191 7 337 457 287 753 383 945 909 209 758 221 588 422 946 506 30 413 168 900 591 762 655 410 359
624 537 548 483 595 41 602 350 291 836 374 20 596 21 348 199 668 484 281 734 53 999 418 938 900 788 127 467 728 893 648 483 807 421 310 617 813 514 309 616
935 451 600 249 519 556 798 303 224 8 844 609 089 702 195 485 93 343 523 587 314 503 448 200 458 618 580 796 798 281 589 798 9 157 472 622 538 292 30 179 19
0 657 958 101 815 888 156 511 202 634 272 55 328 646 362 886 875 433 869 142 844 416 881 998 322 651 21 699 557 476 892 389 75 712 600 510 3 869 861 688 401
789 255 423 2 585 182 285 88 426 617 757 832 932 169 154 721 189 976 329 368 692 425 555 434 549 441 512 145 60 718 753 139 423 279 996 687 529 549 437 866
949 193 195 297 416 286 105 488 282 455 734 114 701 316 671 786 263 313 355 185 53 912 808 832 945 313 756 321 558 646 982 481 144 196 222 129 161 535 450
173 466 44 659 292 439 253 24 154 510 745 649 186 313 474 22 168 18 787 905 958 391 202 625 477 414 314 824 334 874 372 159 833 70 487 297 518 177 773 270 7
63 668 192 985 102 480 213 627 802 99 527 625 543 924 23 972 61 181 3 432 505 593 725 31 492 142 222 286 64 900 187 360 413 974 270 170 235 833 711 760 896
667 285 550 140 694 695 624 19 125 576 694 658 302 371 466 678 593 851 484 18 464 119 152 800 87 60 926 10 757 170 315 576 227 43 758 164 109 882 86 565 487
577 474 625 627 629 928 423 520 902 962 123 596 737 261 195 525 41 467
C:\Users\vladi\source\repos\LAB3BIB211SannikovVladimir\src\Debug\LAB3BIB211SannikovVladimir.exe (процесс 26352) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрывать консоль при оста
новке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Скриншот 4 – Большие входные данные.

Как мы можем заметить, программа выполняется правильно. Более того, не происходит переполнения. Также стоит заметить, что сдвиг происходит справа налево.

3 Выводы о проделанной работе

В результате выполнения данной лабораторной работы мне удалось успешно написать кодовую реализацию для поставленного задания, определить область входных и выходных данных, в которой может работать программа и проверить её на работоспособность и сбои. В итоге был написан код на Assembler MASM x64 и C, причём генерация данных и вызов функции для сдвига, а также вывод нового массива. Я углубил свои познания в теоретической части и стал лучше понимать изучаемую тему.

ПРИЛОЖЕНИЕ А

Листинг полного кода на C.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
```

```
extern void f(int32_t shift, int32_t len, int32_t* mass);
```

```

int main() {
    SetConsoleOutputCP(1251);
    int32_t n;
    int32_t shift;
    printf("Введите размер массива\n");
#pragma warning(suppress : 4996)
    scanf("%d", &n);
    if (n <= 0) {
        printf("длина меньше 0");
        return 1;
    }
    printf("Введите количество позиций для циклического сдвига\n");
#pragma warning(suppress : 4996)
    scanf("%d", &shift);
    int* mass = (int32_t*)malloc(n * sizeof(int32_t));
    for (int i = 0; i < n; i++) {
        mass[i] = rand() % 1000;
    }
    printf("Сгенерированный массив\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", mass[i]);
    }
    f(shift, n, mass);
    printf("\n");
    printf("Итоговый массив\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", mass[i]);
    }
    free(mass);
}

```

ПРИЛОЖЕНИЕ Б

Полный код ассемблерной функции на языке MASM x64:

```
CODE
f PROC
    dec rdx
zxc:
    xor ebx, ebx
    xor rsi, rsi
    mov eax,[r8]
shifter:
    inc rsi
    mov ebx, [r8+(rsi)*4]
    dec rsi
    mov [r8+(rsi)*4], ebx
    inc rsi
    cmp rsi, rdx
    jl shifter
    mov [r8+(rsi)*4],eax
    loop zxc
    ret
f ENDP
END
```