

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 2

Дисциплина: Низкоуровневое программирование

Тема: программирование EDSAC

Вариант: 17

Выполнил студент гр. 3530901/90002 _____ А.И. Юрченко
(подпись)

Принял старший преподаватель _____ Д. С. Степанов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург
2021

Формулировка задачи

1. Разработать программу для EDSAC, реализующую определенную вариантом задания функциональность, и предполагающую загрузчик Initial Orders 1. Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.

2. Выделить определенную вариантом задания функциональность в замкнутую (closed) подпрограмму, разработать вызывающую ее тестовую программу.

Использовать возможности загрузчика Initial Orders 2. Адрес обрабатываемого массива данных и другие параметры передавать через ячейки памяти с фиксированными адресами.

Вариант задания

По варианту номер 17 необходимо реализовать реверс массива. Алгоритм состоит из одного прохода по массиву. За этот проход элемент с индексом [1] меняется местами с элементом с индексом [n], элемент с индексом [2] меняется местами с элементом с индексом [n-1] и так далее. Где n – длина массива.

Initial Orders 1

В данном алгоритме массив, который получается в результате программы, выводится не на месте старого, а следом за ним.

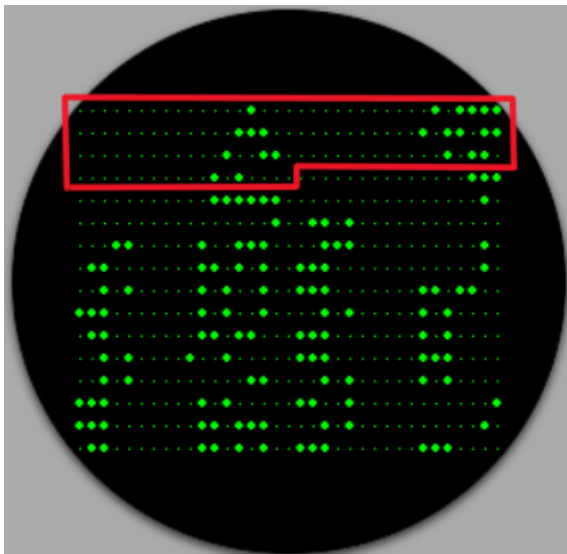
Алгоритм следующий:

- Берём адрес последнего и записываем его в начало нового массива, дальше модифицируем в цикле загрузку и запись нового числа, уменьшая текущий адрес элемента массива и увеличивая адрес ячейки записи.
- Чтобы совершить нужное количество итераций заведём для этого счётчик, который будем уменьшать каждый раз, как совершим итерацию.

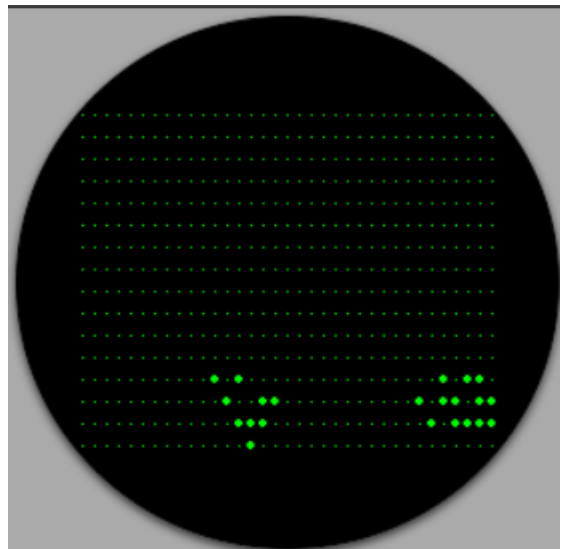
В конце программы записаны константы числа один, два, адрес последнего элемента массива, длины массива и сам массив.

Код с комментариями представлен в приложении 1.

Проведём тестирование программы:



Массив изначально



Массив после реверса

Initial Orders 2 и руководство программиста

Мной была усовершенствована программа из Initial Orders 1 и переписана с учётом директив Initial Orders 2. При помощи параметров я определил ячейки, где должны храниться длина и адрес массива, а также использовал фиксированные адреса «2» инструкции «P 1 F» и «3» – «U 2 F», которые заносятся загрузчиком Initial Orders 2, что позволяет сделать закрытую подпрограмму практичней для использования.

Изменился алгоритм реверса массива:

- Теперь результат реверса выводится на месте изначального массива.
- Счётчик итераций не нужен, вместо этого в конце подпрограммы проверяем больше ли изменённый второй адрес элемента первого, если нет, значит подпрограмма закончила реверс.

Руководство программиста:

Раздел [addresses] отвечает за установку адресов параметров для подпрограммы.

[sub] – подпрограмма сортировки обмёнами. Далее приведу пояснения по коду:

[0-1] Запись адреса возврата для выхода из подпрограммы.

[2-6] Запись параметров в выделенные для этого ячейки.

[7-19] Изменяем код, вставляя туда нужные адреса.

[20-25] Меняем местами элементы массива.

[26-31] Адрес первого элемента (adr1) увеличили, адрес второго элемента (adr2) уменьшили.

[32-35] Проверка, вычитаем adr1 из adr2, если результат положительный, продолжаем реверс, иначе – выходим из подпрограммы.

[36-41] Шаблоны для инструкций и переменные подпрограммы.

[test] – тестовая программа, которая вызывает замкнутую подпрограмму.

Вызывается два раза на два разных массива для проверки работы подпрограммы при неоднократном запуске.

Приложение 1

Листинг программы для загрузчика Initial Orders 1:

```
T [end]64 S
A [length]56 S [загружаем в акк длину массива]
S [one]53 S [длина массива - 1]
T 1 S [оставшиеся итерации]
A [last adr]55 S [загрузка в акк адрес последнего числа]
L 0 L
A [load]40 S [прибавляем в акк инструкцию из строки 40]
T [load]40 S [выгружаем в строку 40 адрес последнего числа]
[loop] T 3 S [очищаем акк]
[load] A 0 S [загрузка в акк текущий элемент массива]
[write]T [end+1]65 S [выгружаем элемент массива]
    A [load]40 S [загружаем инструкцию из 40 строки]
    S [two]54 S [вычитаем двойку]
    T [load]40 S [выгружаем в строку 40 адрес следующего числа]
    A [write]41 S [загружаем инструкцию из 41 строки]
    A [two]54 S [прибавляем двойку]
    T [write]41 S [выгружаем в строку 41 адрес для записи
следующего числа]
        A 1 S [загружаем в акк число оставшихся итераций]
        S [one]53 S [вычитаем один]
        U 1 S [выгружаем обратно]
        E [loop]39 S [если число итераций >=0, переходим в строку 39]
Z 0 S
[one] P 0 L
[two] P 1 S
[last adr] P 31 L [адрес последнего числа в массиве]
[length] P 3 L [длина массива]
```

P 20 S [40]

P 11 S [22]

P 9 L [19]

P 45 L [91]

P 7 S [14]

P 23 L [47]

[last] P 2 S [4]

[end]

Приложение 2

Листинг программы для загрузчика Initial Orders 1:

T 56 K [Адрес загрузки]

[addresses]

G K [Директива, фиксация начального адреса]

T 45 K [Установка адреса для параметров далее]

P 256 F [45 = H, тут будет адрес первого элемента массива]

P 255 F [46 = N, длина массива]

TZ [восстановления адреса целевой ячейки]

[sub]

G K [Директива, фиксация начального адреса]

[далее в квадратных скобах адреса @]

[0] A 3 F [Пролог: формирование кода инструкции возврата]

[1] T [return]35 @ [Пролог: запись инструкции возврата]

[2] A 0 H [adr = H]

[3] U [adr1]38 @

[4] A 0 N

[5] S [two]2 F

[6] T [adr2]39 @

[7] [START] T 41[trash] @ [self-modified code]

[8] A [AOF]36 @

[9] A [adr1]38 @

[10] T [A1]20 @

[11] A [AOF]36 @

[12] A [adr2]39 @

[13] T [A2]22 @

[14] A [TOF]37 @


```

[15]    A [adr1]38 @
[16]    T [T1]23 @
[17]    A [TOF]37 @
[18]    A [adr2]39 @
[19]    T [T2]25 @    [/self-modified code]
[20]    [A1] A 0 F [swapping]
[21]        T [temp]40 @
[22]    [A2] A 0 F
[23]    [T1] T 0 F
[24]        A [temp]40 @
[25]    [T2] T 0 F    [/swapping]
[26]    A [adr1]38 @ [меняем адрес первого числа]
[27]    A [two]2 F
[28]    T [adr1]38 @
[29]    A [adr2]39 @ [меняем адрес второго числа]
[30]    S [two]2 F
[31]    U [adr2]39 @
[32]    S [adr1]38 @
[33]    E [START]7 @
[34] [EXIT] T [trash]41 @
[35] [return] E 0 F [Эпилог: возвращение из функции]
[CONSTS]
[36] [A0F] A 0 F
[37] [TOF] T 0 F
[38] [adr1] P 0 F
[39] [adr2] P 0 F
[40] [temp] P 0 F
[41] [trash] P 0 F

```

[test]
G K
[0] X 0 F
[1] A [adr]8 @
[2] T 256 F [Записываем адрес первого элемента в H]
[3] A [length]9 @
[4] T 255 F [Записываем длину массива в N]
[5] A 5 @
[6] G [sub]56 F [subprogram]
[7] E 23 @ [Переходим во второй тест]
[8] [adr] P 13 @
[9] [length] P 10 F
[10] [space] P 0 F
P 0 F
P 0 F
[13] [array] P 1023 F
P 511 F
P 255 F
P 127 F
P 63 F
P 31 F
P 15 F
P 7 F
P 3 F
P 1 F
[23] A [adr2]30 @
[24] T 256 F [Записываем адрес первого элемента в H]
[25] A [length2]31 @
[26] T 255 F [Записываем длину массива в N]

[27] A 27 @

[28] G [sub] 56 F

[29] Z 0 F [STOP]

[30] [adr2] P 33 @

[31] [length2] P 8 F

P 0 F [space]

P 7 F [array]

P 31 F

P 1 F

P 255 F

P 127 F

P 1023 F

P 15 F

P 4095 F

EZ PF