

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**"Южно-Уральский государственный университет
(национальный исследовательский университет)"**

**Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»**

Отчет по лабораторной работе № 3
на тему «Анализ командной реализации программы»
Вариант № 12

Авторы работы:

студенты группы КЭ-303

_____ / Д.В. Старостенок

« ____ » _____ 2023 г.

_____ / Е.В. Елисеев

« ____ » _____ 2023 г.

Проверил

_____ / И.Л. Кафтанников

« ____ » _____ 2023 г.

Челябинск 2023 г.

ОГЛАВЛЕНИЕ

ЗАДАНИЕ	3
КОД АССЕМБЛЕРА ДЛЯ PDP-11	4
КОД АССЕМБЛЕРА ДЛЯ МП-580.....	5
КОД АССЕМБЛЕРА ДЛЯ IA32	6
СРАВНЕНИЕ ПРОГРАММ ПО КОЛИЧЕСТВУ КОМАНД.....	7
ПОДСЧЕТ КОЛИЧЕСТВА КОМАНД ПО ГРУППАМ.....	7
ПРОЦЕНТНОЕ СООТНОШЕНИЕ КОМАНД ПРЕОБРАЗОВАНИЯ ДАННЫХ И ВСПОМОГАТЕЛЬНЫХ СООТВЕТСТВЕННО АРХИТЕКТУРЕ ПРОЦЕССОРА	7
ВЫВОДЫ.....	7
СПИСОК ЛИТЕРАТУРЫ.....	9

ЗАДАНИЕ

Согласно полученному варианту разработать программу решения задачи на языке Assembler, используя следующие системы команд:

- PDP–11;
- МП 580;
- IA32.

Этапы выполнения задания:

- Реализация программ на языке Assembler для предложенных систем команд, синтаксис которых описан в справочниках, приложенных к лабораторной работе;

- Сравнение синтезированных программ по количеству команд;
- Подсчет количества команд каждой группы:

1. Арифметические;
2. Логические;
3. Сдвига;
4. Управления;
5. Пересылки;
6. Ввода–вывода;
7. Прочие.

Цель работы:

Практическое применение навыков разработки программного обеспечения на низкоуровневом языке программирования Assembler, получение опыта анализа разработанной программы по критерию количества команд каждой функциональной группы языка.

В соответствии с вариантом 12.

Задача: Число, чаще всего встречающееся в массиве

Определить, какое число в массиве встречается чаще всего.

КОД АССЕМБЛЕРА ДЛЯ PDP-11

<pre> .org 1000 mov #array, r0 ; указатель на начало массива mov #end_of_array, r1 ; указатель на конец массива mov #0, r2 ; текущий наиболее часто встречающийся элемент mov #0, r3 ; текущее количество вхождений наиболее часто встречающегося элемента loop: cmp r0, r1 ; если достигнут конец массива, выходим из цикла beq end_loop mov (r0)+, r4 ; загружаем текущий элемент массива mov #array, r5 ; указатель на начало массива mov #0, r6 ; текущее количество вхождений текущего элемента inner_loop: cmp r5, r1 ; если достигнут конец массива, выходим из цикла beq end_inner_loop cmp (r5)+, r4 ; сравниваем текущий элемент с очередным элементом массива bne continue ; если элементы не равны, продолжаем цикл inc r6 ; увеличиваем счетчик вхождений текущего элемента continue: </pre>	<pre> bra inner_loop ; переходим к следующему элементу массива end_inner_loop: cmp r6, r3 ; если текущий элемент встретился чаще, чем наиболее частый элемент, обновляем значения ble continue_loop mov r4, r2 ; обновляем наиболее часто встречающийся элемент mov r6, r3 ; обновляем текущее количество вхождений наиболее часто встречающегося элемента continue_loop: bra loop ; переходим к следующему элементу массива end_loop: ; выводим результат на экран mov r2, r0 jsr pc, print_r0 halt </pre>
---	---

КОД АССЕМБЛЕРА ДЛЯ МП-580

```
.org 0h

; инициализация указателя на начало массива
lxi h, array

; инициализация переменных
mvi b, 0 ; счетчик наиболее часто встречающегося числа
mvi c, 0 ; наиболее часто встречающееся число
mvi d, 0 ; временный счетчик
mvi e, 0 ; временное значение числа

; перебор элементов массива
next_element:
    ldax h ; загрузка значения элемента в регистр A
    mov e, a ; сохранение значения в регистре E
    mvi a, 0 ; обнуление регистра A
    mvi d, 0 ; обнуление счетчика

    ; подсчет количества вхождений числа в массиве
    count_occurrences:
        cmp e ; сравнение регистра A с значением числа
        jnz next_byte ; если не совпадает, переход к следующему элементу
        inr d ; увеличение счетчика
    next_byte:
        inx h ; сдвиг указателя на следующий элемент
        mov a, h ; сохранение указателя в регистре A
        cpi end_of_array - 1 ; сравнение с концом массива
        jnz count_occurrences ; если не конец массива, продолжение подсчета

    ; проверка на наиболее часто встречающееся число
    cmp b ; сравнение наиболее частого числа с текущим
    jnc next_element ; если текущее число встречается реже, переход к
    следующему элементу
    mov b, d ; сохранение счетчика в регистре B
    mov c, e ; сохранение числа в регистре C
    jmp next_element ; переход к следующему элементу

; вывод результата
result:
    ; вывод наиболее часто встречающегося числа и количества вхождений
    ; внимание: это пример вывода на экран, конкретная реализация может
    отличаться
    mvi a, c ; загрузка числа в регистр A
    call print_number ; вывод числа на экран
    mvi a, ' ' ; загрузка пробела в регистр A
    call print_char ; вывод пробела на экран
    mvi a, '(' ; загрузка открывающей скобки в регистр A
    call print_char ; вывод открывающей скобки на экран
    mvi a, b ; загрузка количества вхождений в регистр A
```

КОД АССЕМБЛЕРА ДЛЯ IA32

<pre> org 100h mov ecx, 0 ; инициализируем переменную счетчика на 0 mov edx, 0 ; инициализируем переменную счетчика максимального повторения на 0 mov esi, array ; загрузка адреса массива в регистр esi mov ebx, end_of_array - array ; количество элементов массива mov eax, [esi] ; загрузка первого элемента массива в регистр eax check_array: mov edi, esi ; копируем адрес текущего элемента в регистр edi mov ecx, 0 ; обнуляем счетчик повторений текущего элемента mov edx, 0 ; обнуляем счетчик максимальных повторений mov ebx, end_of_array - array ; количество элементов массива cmp ebx, 0 ; проверка, что мы не дошли до конца массива je done ; переход к завершению, если достигнут конец массива loop_array: cmp [edi], eax ; сравнение текущего элемента с загруженным в eax jne skip ; переход к пропуску, если элемент не равен текущему inc ecx ; увеличиваем счетчик повторений текущего элемента </pre>	<pre> skip: add edi, 4 ; увеличиваем указатель на следующий элемент массива dec ebx ; уменьшаем счетчик оставшихся элементов cmp ebx, 0 ; проверяем, достигнут ли конец массива jne loop_array ; если нет, переходим к следующему элементу cmp ecx, edx ; сравнение счетчика повторений текущего элемента с максимальным jle continue ; если текущий элемент повторился меньше или столько же раз, ; то пропускаем его mov edx, ecx ; иначе, обновляем максимальный счетчик повторений mov ebx, eax ; и сохраняем текущий элемент в регистре ebx continue: add esi, 4 ; переходим к следующему элементу массива mov eax, [esi] ; загружаем следующий элемент массива в регистр eax dec end_of_array - array ; уменьшаем количество оставшихся элементов массива jnz check_array ; если не достигнут конец массива, продолжаем цикл done: mov eax, ebx ; сохраняем максимальный элемент в регистре eax call print_eax ; выводим его на экран int 20h ; завершаем программу </pre>
--	---

СРАВНЕНИЕ ПРОГРАММ ПО КОЛИЧЕСТВУ КОМАНД

PDP-11	МП-580	IA32
38	59	39

ПОДСЧЕТ КОЛИЧЕСТВА КОМАНД ПО ГРУППАМ

Тип	PDP-11	МП-580	IA32
Арифметические	0	6	2
Логические	2	4	3
Сдвига	0	2	0
Управления	12	7	9
Пересылки	8	14	5
Ввода вывода	2	3	1
Прочие	14	23	19

ПРОЦЕНТНОЕ СООТНОШЕНИЕ КОМАНД ПРЕОБРАЗОВАНИЯ ДАННЫХ И ВСПОМОГАТЕЛЬНЫХ СООТВЕТСТВЕННО АРХИТЕКТУРЕ ПРОЦЕССОРА

	PDP-11	МП-580	IA32
Преобразования	26,3%	24,1%	23,07%
Вспомогательные	73,7%	75,9%	76,92%

ВЫВОДЫ

В рамках данного сравнения были проанализированы три архитектуры процессоров: PDP-11, МП-580 и IA32. При этом было выявлено, что МП-580 содержит наибольшее количество команд 59, в то время как PDP-11 имеет наименьшее количество команд 38. Количество команд для IA32 составляет 39.

Для каждой из архитектур процессоров было проанализировано количество команд по группам: арифметические, логические, сдвига, управления, пересылки, ввода-вывода и прочие. В результате анализа было выявлено следующее:

Для всех трех архитектур процентное соотношение команд преобразования данных составляет примерно 23-26%. Наибольшее

процентное соотношение команд преобразования данных у PDP-11 (26,3%), а наименьшее у IA32 (23,07%).

Вспомогательные команды составляют примерно 74-77% от общего числа команд для каждой из архитектур. Наибольшее процентное соотношение вспомогательных команд у IA32 (76,92%), а наименьшее у PDP-11 (73,7%).

СПИСОК ЛИТЕРАТУРЫ

- 1) Справочник Форматы команд, директив и пр.
- 2) Справочник Архитектура однокристалльного микропроцессора K580BM80A
- 3) Негода В.Н., Никищенков И.А. Функциональная организация микро-эвм и микроконтроллеров часть 1: pdp-11