
Задание для лабораторной работы N 3.
Язык Ассемблера для IBM PC
гр. 350 - 354

Процедуры и модульное программирование.

Цель работы: ознакомиться с правилами построения процедур, способов обращения к ним и передачи параметров, модульного программирования.

Задание: Написать два исходных модуля на языке Ассемблер первый из которых содержит главную процедуру, а второй - вспомогательную.

Главная процедура подготавливает исходные данные для вспомогательной процедуры и вызывает ее. Все необходимые для работы переменные описываются в модуле главной процедуры и являются внешними для вспомогательной процедуры. Вспомогательная процедура осуществляет преобразование данных, переданных ей главной процедурой в соответствии с вариантом задания. Передача параметров во вспомогательную процедуру организуется через стек. Все необходимые для вспомогательной процедуры переменные должны быть организованы как локальные и размещены в стеке. Вспомогательная процедура использует стек главной процедуры и не имеет собственного сегмента данных.

Действия вспомогательной процедуры, количество и вид передаваемых ей параметров выбираются из таблицы 1 в соответствии с вариантом задания.

Лабораторная работа выполняется в два этапа: сначала подготавливается вспомогательная процедура и осуществляется ее отладка. После этого вспомогательная процедура в соответствии с заданием помещается в отдельный модуль, подготавливается главная процедура, осуществляется их раздельная компиляция и редактирование связей полученных объектных модулей. Описанная в данном абзаце последовательность действий не обязательная, а рекомендуемая.

Вспомогательная процедура должна сохранять все используемые регистры, и возвращать в регистре AX нулевое значение если преобразование прошло успешно и ненулевое если при преобразовании была обнаружена ошибка (например, значение заданного в символической форме числа не может быть размещено в требуемом количестве разрядов и т.п.).

Таблица 1.

N Вар.	Исходное представление	Вид результата	Способ передачи параметра		Максималь- ная длина строки символов.
			1-й пар.	2-й пар.	
1	1.1	2.6	п/з	п/а	5
2	1.2	2.3	п/з	п/а	5
3	1.3	2.5	п/а	п/а	8
4	1.4	2.1	п/а	п/а	6
5	1.5	2.2	п/а	п/а	8
6	1.2	2.4	п/з	п/а	4
7	1.2	2.5	п/з	п/а	4
9	1.2	2.6	п/з	п/а	10
10	1.2	2.7	п/з	п/а	12
11	1.3	2.3	п/з	п/а	8
12	1.3	2.4	п/з	п/а	8
13	1.4	2.2	п/а	п/а	10
14	1.5	2.1	п/а	п/а	6
15	1.6	2.1	п/а	п/а	5

16	1.6	2.2	n/a	n/a	9
17	1.7	2.1	n/a	n/a	5
18	1.7	2.2	n/a	n/a	9
19	1.8	2.1	n/a	n/a	6
20	1.8	2.2	n/a	n/a	12
21	1.1	2.8	n/a	n/a	6
22	1.1	2.9	n/a	n/a	7
23	1.2	2.8	n/з	n/a	7
24	1.2	2.9	n/з	n/a	8
25	1.3	2.8	n/з	n/a	11
26	1.3	2.9	n/з	n/a	12
27	1.9	2.1	n/a	n/a	10
28	1.9	2.1	n/a	n/a	14
29	1.8	2.3	n/a	n/a	10
30	1.8	2.4	n/a	n/a	10

Вид исходных данных:

- 1.1. Содержимое переменной размером в байт.
 - 1.2. Содержимое переменной размером в слово.
 - 1.3. Содержимое переменной размером в двойное слово.
- Строка символов, содержащая символьное представление
- 1.4. Десятичного числа без знака;
 - 1.5. Десятичного числа со знаком;
 - 1.6. Шестнадцатеричного числа со знаком;
 - 1.7. Шестнадцатеричного числа без знака;
 - 1.8. Восьмеричного числа без знака.
 - 1.9. Восьмеричного числа со знаком

=====
Исходные данные необходимо преобразовать в

- 2.1. Шестнадцатеричное число в формате WORD.
- 2.2. Шестнадцатеричное число в формате DWORD.
- 2.3. Символьное представление в десятичном формате со знаком.
- 2.4. Символьное представление в десятичном формате без знака.
- 2.5. Символьное представление в шестнадцатеричном формате.
- 2.6. Символьное представление в троичном формате со знаком.
- 2.7. Символьное представление в троичном формате без знака.
- 2.8. Символьное представление в восьмеричном формате со знаком.
- 2.9. Символьное представление в восьмеричном формате без знака.

=====
Способ передачи параметров:

- п/а - по адресу;
- п/з - по значению.