

Лабораторная работа No8.

Программирование цикла. Обработка аргументов командной строки.

Бурыктна Софья Дмитриевна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	13
5	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Создание	7
3.2	Создала файл	8
3.3	Результат работы	8
3.4	Текст программы	9
3.5	Работа программы	10
3.6	Текст програмы	11
3.7	Работа программы	11
3.8	Работа программы	12
3.9	Работа программы	12
3.10	Работа программы	12
4.1	Работа программы	14
4.2	Работа программы	14

Список таблиц

1 Цель работы

Написания программ с использованием циклов и обработкой аргументов командной строки.

2 Теоретическое введение

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров. На рис. 8.1 показана схема организации стека в процессоре. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается. Для стека существует две основные операции: • добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop) |

Более подробно об Unix см. в [1–6].

3 Выполнение лабораторной работы

Создала каталог и создала файл (рис. 3.1).

```
sdburikhkina@dk4n65 ~ $ mkdir ~/work/arch-pc/lab08  
sdburikhkina@dk4n65 ~ $ cd ~/work/arch-pc/lab08  
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ touch lab8-1.asm
```

Рис. 3.1: Создание

Ввела в файл lab8-1.asm текст программы из листинга 8.1 (рис. 3.2).

```

lab8-1.asm      [----] 12 L:[ 1+26 27/ 28] *(609 / 670b) 0032 0x
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ----- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения 'N'
loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
; ----- ; переход на 'label'
call quit

```

Рис. 3.2: Создала файл

Проверила работу программы (рис. 3.3).

```

sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab6-1 lab6-1.o
ld: невозможно найти lab6-1.o: Нет такого файла или каталога
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 5
5
4
3
2
1
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $

```

Рис. 3.3: Результат работы

Изменила текст программы (рис. 3.4).


```

lab8-1.asm      [----] 10 L:[ 1+26 27/ 28] *(531
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ----- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
loop label

```

Рис. 3.4: Текст программы

Проверила его работу. Принимает разные значения. Не соответствует вводу с клавиатуры, ошибка (рис. 3.5).

```
sdburikhina@dk5n59 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 10
9
7
5
3
1
Ошибка сегментирования (стек памяти сброшен на диск)
sdburikhina@dk5n59 ~/work/arch-pc/lab08 $
```

Рис. 3.5: Работа программы

Изменила текст программы (рис. 3.6).

```

/afs/.dk.sci.pfu.edu.ru/home/s/d/sdburikhina/work/arch-
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ----- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
push ecx ; добавление значения ecx в стек
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx ; извлечение значения ecx из стека
loop label

```

Рис. 3.6: Текст программы

Проверила роботу (рис. 3.7).

```

sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ mc
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ./lab6-1
bash: ./lab6-1: Нет такого файла или каталога
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 3
2
1
0
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ █

```

Рис. 3.7: Работа программы

Создала файл исполняемый файл и запустила его указав аргументы, три аргу-

мента были обработаны программой (рис. 3.8).

```
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ./lab8-2 1 2 3
1
2
3
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $
```

Рис. 3.8: Работа программы

Создала исполняемый файл и запустила его, указав аргументы из примера (рис. 3.9).

```
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 47
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $
```

Рис. 3.9: Работа программы

Изменила текст файла для вычисления произведения аргументов командной строки и запустила его (рис. 3.10).

```
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 54600
sdburikhina@dk4n65 ~/work/arch-pc/lab08 $
```

Рис. 3.10: Работа программы

4 Самостоятельная работа

Написала программу для вычисления варианта 7(рис. 4.1). проверила правильную работу программы (рис. 4.2).

```

/afs/.dk.sci.pfu.edu.ru/home/s/d/sdburikhkina/work/arch-pc/
%include 'in_out.asm'
SECTION .data
msg1 db "Функция:3(x+2) ",0
msg2 db "Результат: ",0

SECTION .text
global _start
_start:
por ecx ; Извлекаем из стека в 'ecx' количество
; аргументов (первое значение в стеке)
por edx ; Извлекаем из стека в 'edx' имя программы
; (второе значение в стеке)
sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
; аргументов без названия программы)
mov esi, 0 ; Используем 'esi' для хранения
; промежуточных сумм
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку '_end')
por eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число

mov ebx, 2
add eax,ebx
mov ebx, 3
mul ebx

add esi,eax
loop next ; переход к обработке следующего аргумента
_end:
mov eax, msg1 ; вывод сообщение "Функция"
mov eax, msg2 ; вывод сообщения "Результат: "
call sprint
mov eax, esi ; записываем сумму в регистр 'eax'
call iprintLF ; печать результата
call quit ; завершение программы

```

Рис. 4.1: Работа программы

```

sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ nasm -f elf lab8-4.asm
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $ ./lab8-4 4 1
Результат: 27
sdburikhkina@dk4n65 ~/work/arch-pc/lab08 $

```

Рис. 4.2: Работа программы

5 Выводы

Были пролучены навыки по организации и работе сос теком на языке NASM.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.