ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №9

Дисциплина: Архитектура компьютера

Скрипникова София Дмитриевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

2 Задание

Приобрести навыки написания программ с использованием циклов и обработки аргументов командной строки.

3 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.

Команда push размещает значение в стеке, т.е. помещает значение в ячейку памяти, на которую указывает регистр esp, после этого значение регистра esp увеличивается на 4. Данная команда имеет один операнд — значение, которое необходимо поместить в стек.

Команда pop извлекает значение из стека, т.е. извлекает значение из ячейки памяти, на которую указывает регистр esp, после этого уменьшает значение регистра esp на 4. У этой команды также один операнд, который может быть регистром или переменной в памяти.

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop.

4 Выполнение лабораторной работы

1. Создала каталог для программ лабораторной работы №9, перешла в него и создала файл lab9-1.asm. (рис. 4.1)

```
[sdskripnikova@fedora ~]$ mkdir ~/work/arch-pc/lab09
[sdskripnikova@fedora ~]$ cd ~/work/arch-pc/lab09
[sdskripnikova@fedora lab09]$ touch lab9-1.asm
```

Рис. 4.1: Создание, переход в lab09

2. Ввела в файл lab9-1 нужный текст программы из листинга 9.1., создала исполняемый файл и вывела результат. (рис. 4.2; рис. 4.3)

```
label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения `N`
loop label ; `ecx=ecx-1` и если `ecx` не '0'
; переход на `label`
```

Рис. 4.2: Ввод текста

```
[sdskripnikova@fedora lab09]$ nasm -f elf lab9-1.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[sdskripnikova@fedora lab09]$ ./lab9-1
Введите N: 5
5
4
3
2
1
[sdskripnikova@fedora lab09]$
```

Рис. 4.3: Результат программы

3. Изменила текст программы, добавив изменения значения регистра ecx в цикле. Цикл закольцевался и стал бесконечным. (рис. 4.4; рис. 4.5)

```
label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ;Вывод значения `N`
pop ecx
loop label ; `ecx=ecx-1` и если `ecx` не '0'
; переход на `label`
call quit
```

Рис. 4.4: Изменения текста

```
[sdskripnikova@fedora lab09]$ nasm -f elf lab9-1.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[sdskripnikova@fedora lab09]$ ./lab9-1
Введите N: 6
5
3
1
[sdskripnikova@fedora lab09]$
```

Рис. 4.5: Результат изменений

4. Изменила текст программы, добавив команды push и pop (добавление строк и извлечение из стека) для сохранения значения счетчика цикла loop. После изменения программы, число проходок циклов стал соответствовать числу введенному с клавиатуры. (рис. 4.6; рис. 4.7)

```
%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
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ;Вывод значения `N`
pop ecx
loop label ; `ecx=ecx-1` и если `ecx` не '0'
; переход на `label`
call quit
```

Рис. 4.6: Изменила программу

```
[sdskripnikova@fedora lab09]$ nasm -f elf lab9-1.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[sdskripnikova@fedora lab09]$ ./lab9-1
Введите N: 7
6
5
4
3
2
1
0
[sdskripnikova@fedora lab09]$
```

Рис. 4.7: Вывела результат

5. Создала файл lab9-2.asm в нужном каталоге, ввела нужный текст и вывела результат. Программа выводит все аргументы, введенные при запуске программы. (рис. 4.8; рис. 4.9)

```
%include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в `ecx` количество; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в `edx` имя программы; (второе значение в стеке)
sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество; аргументов без названия программы)
next:
cmp ecx, 0 ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла; (переход на метку `_end`)
pop eax ; иначе извлекаем аргумент из стека
call sprintLF ; вызываем функцию печати
loop next ; переход к обработке следующего; аргумента (переход на метку `next`)
_end:
call quit
```

Рис. 4.8: Ввела нужный текст

```
[sdskripnikova@fedora lab09]$ nasm -f elf lab9-2.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[sdskripnikova@fedora lab09]$ ./lab9-2
[sdskripnikova@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
[sdskripnikova@fedora lab09]$
```

Рис. 4.9: Вывод результата

6. Создала файл lab9-3.asm, ввела в него нужный текст и вывела результат. (рис. 4.10;)

```
[sdskripnikova@fedora lab09]$ touch lab9-3.asm
[sdskripnikova@fedora lab09]$ nasm -f elf lab9-3.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[sdskripnikova@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
[sdskripnikova@fedora lab09]$
```

Рис. 4.10: Результат программы

7. Изменила текст программы для вычисления произведения аргументов командной строки и вывела результат. (рис. 4.11; рис. 4.12)

```
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,1
mov eax,1

next:
cmp ecx,0
jz _end
pop eax
call atoi
mov ebx,eax
mov eax,esi
mul ebx
mov esi,eax
loop next

_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 4.11: Текст программы

```
[sdskripnikova@fedora lab09]$ nasm -f elf lab9-3.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[sdskripnikova@fedora lab09]$ ./lab9-3 1 2 3 4
Результат: 24
```

Рис. 4.12: Результат программы

4.1 Задания для самостоятельной работы

1. Я написала программу, которая находит сумму значений функции f(x) для x = x1, x2, …, xn, т.е. программа должна выводить значение f(x1) + f(x2)+…+f(xn). Значения xi передаются как аргументы. Вариант задания: №6. (рис. 4.13; рис. 4.14)

```
%include 'in_out.asm'

SECTION .data
prim DB 'f(x)=2(x-1)',0
otv DB 'Результат: ',0
SECTION .text
GLOBAL _start
_start:

pop ecx

pop edx

sub ecx,1

mov esi,0

mov eax,prim
call sprintLF
next:
cmp ecx,0
jz _end

mov ebx,1
pop eax
call atoi
sub eax, ebx

mov ebx,2
mul ebx


add esi,eax

loop next

_end:
mov eax,otv
call sprint
mov eax,esi
call iprintLF
call quit
```

Рис. 4.13: Текст программы

```
[sdskripnikova@fedora lab09]$ nasm -f elf sr1.asm
[sdskripnikova@fedora lab09]$ ld -m elf_i386 -o sr1 sr1.o
[sdskripnikova@fedora lab09]$ ./sr1 2 3 4 5
f(x)=2(x-1)
Результат: 20
[sdskripnikova@fedora lab09]$ ./sr1 2 3 4 5 6
f(x)=2(x-1)
Результат: 30
[sdskripnikova@fedora lab09]$
```

Рис. 4.14: Результат

Данные изменения можно проверить по ссылке: https://github.com/sdskripnikova/study_2022-2023_arh-pc/tree/master/labs/lab09

5 Выводы

Приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки.

Список литературы

1. https://esystem.rudn.ru/pluginfile.php/1584393/mod_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%969.pdf