Лабораторная работа 3.

**Модули и функции на ассемблере**

**Цель работы:** изучить процесс компиляции программы на C++; научиться

включать в проекты на языке C++ ассемблерные модули; научиться

описывать функции и вызывать из программы на языке C++.

**Л3.1. Задание на лабораторную работу**

**Задание 1.** Разработайте ассемблерную функцию, вычисляющую целое

выражение от целого аргумента (в соответствии с вариантом), а также

головную программу на языке C++, использующую разработанную функцию.

extern "C" int foo(int x);

\_\_asm\_\_

(

"foo:**\n**"

" movl 8**(**%**esp)**, %**eax\n**"

" **imul** **$**3,%**eax\n**"

" **add** **$**1,%**eax\n**"

" **ret\n**"

);

int main(void)

{

int x = 1;

int y = foo(x);

}

**Задание 2.** Разработайте программу, целиком написанную на ассемблере,

вычисляющую значение для и выводящую полученное

значение на стандартный вывод с использованием библиотеки stdlib (в частности,

функции printf).

**.data:**

printf\_format:

.string "%d\n"

**.globl main**

main:

movl **$**13, %**eax**

**imul** **$**3,%**eax**

**add** **$**1,%**eax**

pushl %**eax**

pushl **$**printf\_format

**call** printf

addl **$**8, %**esp**

movl **$**0, %**eax**

**ret**

**Задание 3.** Опишите функцию на произвольном языке высокого уровня

(включая C/C++) и вызовите её из ассемблерной функции.

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

using namespace std;

extern "C" int fooASM();

extern "C" int foo();

\_\_asm\_\_

(

"fooASM:\n"

" call foo\n"

" ret\n"

);

int foo()

{

int minX, maxX;

printf("Enter range(upper and lower bounds): ");

scanf("%d", &minX);

scanf("%d", &maxX);

int x = rand()%(maxX-minX)+minX;

return x;

}

int main(void)

{

srand(time(NULL));

cout << fooASM();

}

**Задание 4.** Бонус (+2 балла). Опишите на ассемблере одну подпрограмму

с параметрами и результатами и и вызовите её из другой

ассемблерной программы.

.data

printf\_format:

.string "%d\n"

a:

.int 1

b:

.int 2

c:

.int 3

x :

.int 0

y:

.int 0

calc:

movl a, %eax

movl b, %ebx

movl c, %ecx

imul %ecx, %ebx

add %ebx, %eax

movl %eax, x

movl a, %eax

movl b, %ebx

imul %ecx, %ebx

sub %ebx, %eax

movl %eax, y

ret

.globl main

main:

call calc

movl x, %eax

pushl %eax

pushl $printf\_format

call printf

addl $8, %esp

movl $0, %eax

movl y, %eax

pushl %eax

pushl $printf\_format

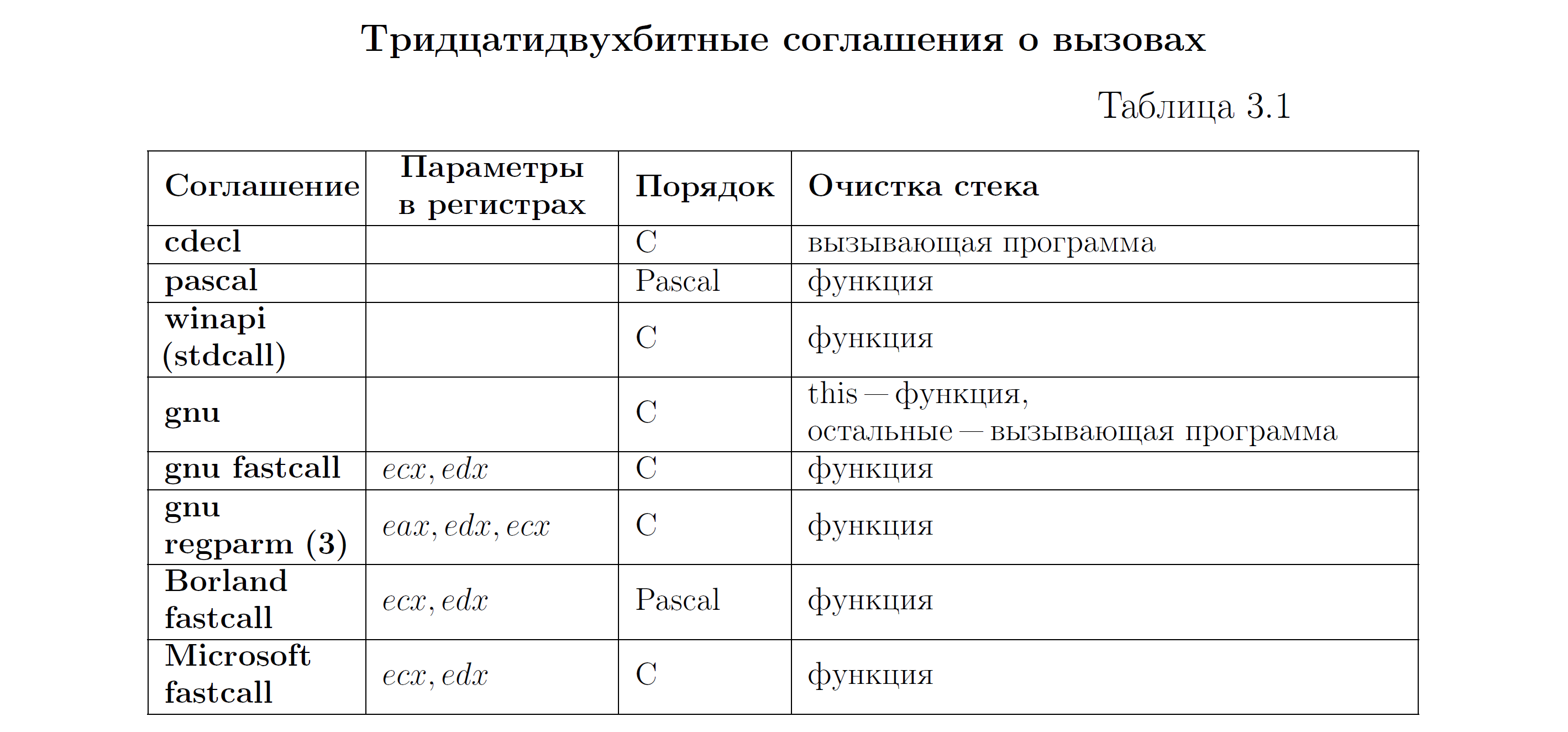
call printf

addl $8, %esp

movl $0, %eax

ret

**Вопросы:**

*1. Какие вы знаете соглашения о вызове?*  


*2. Какая команда передаёт управление подпрограмме?*

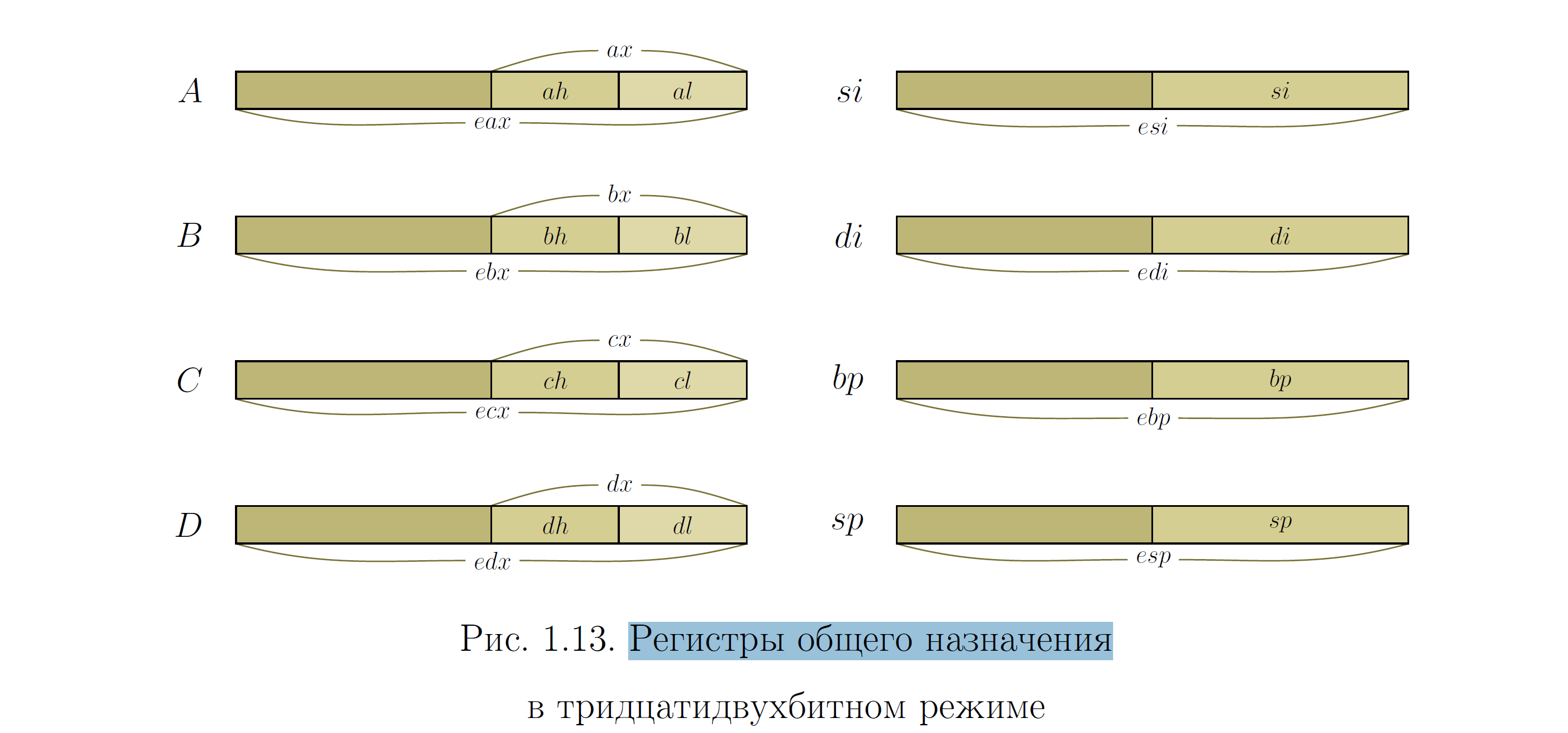
*call*

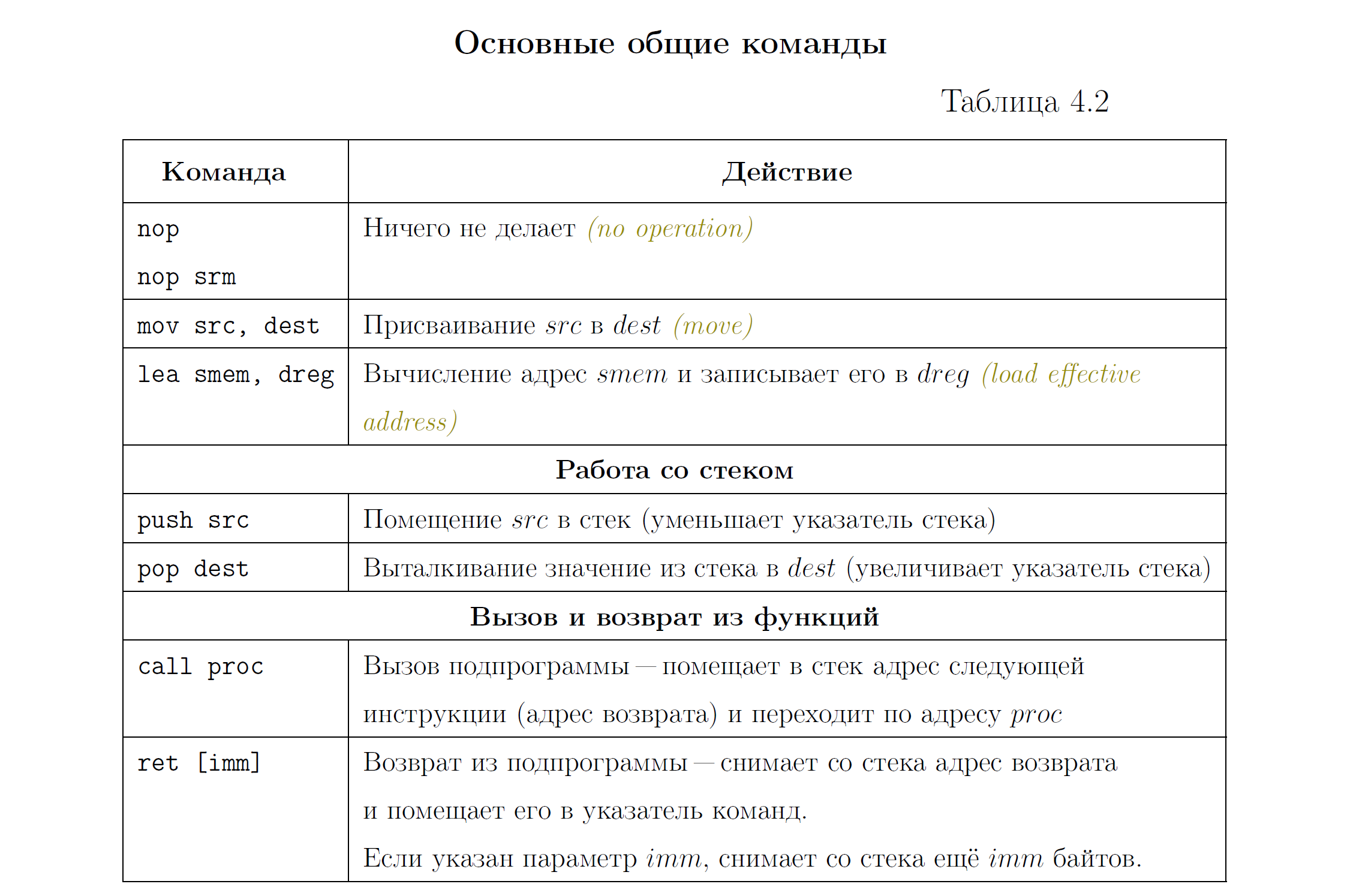
*3. Какая команда возвращает управление вызывающей программе?  
ret*

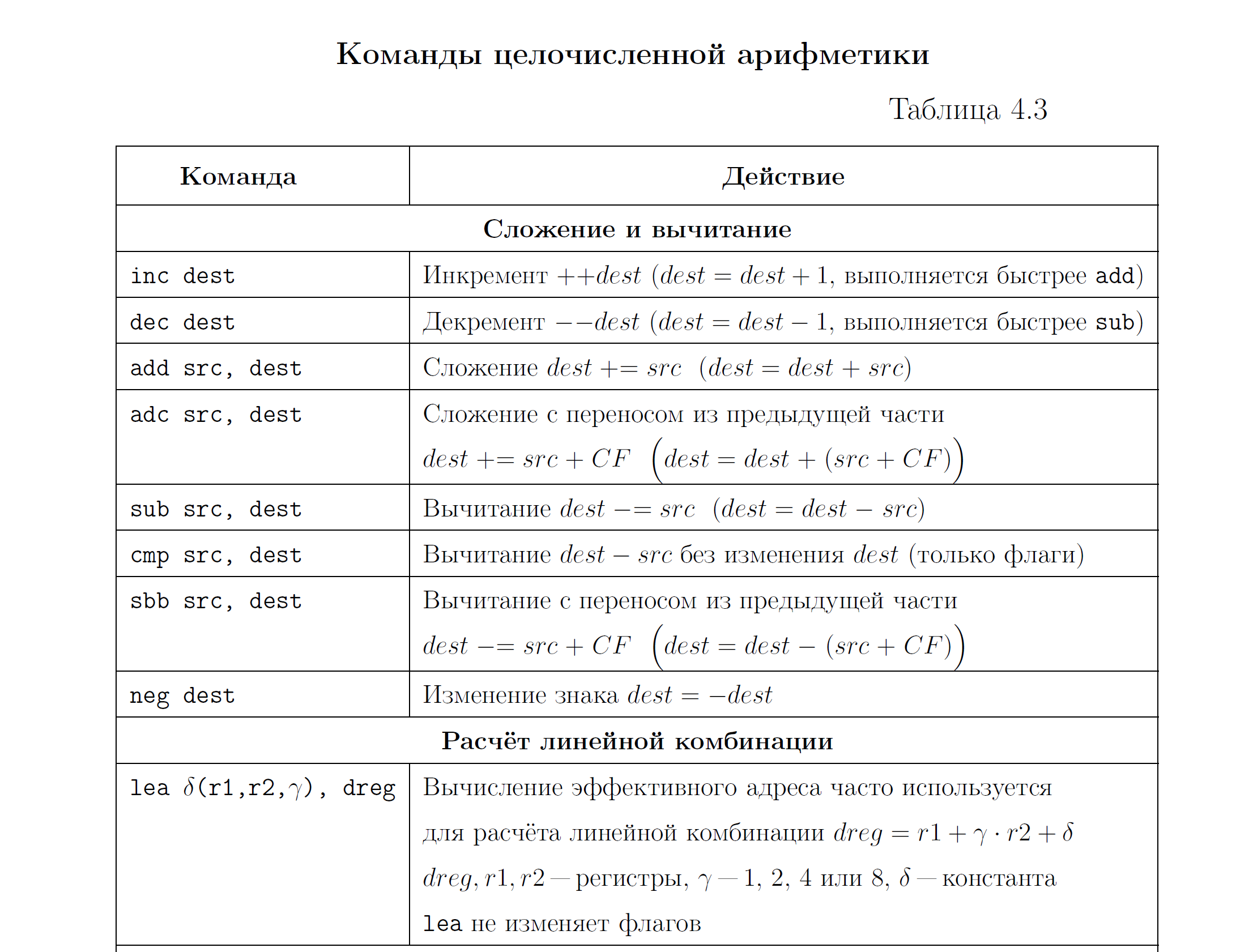
*4. Что такое адрес возврата?*Пусть следующая команда, расположенная по адресу — call f.

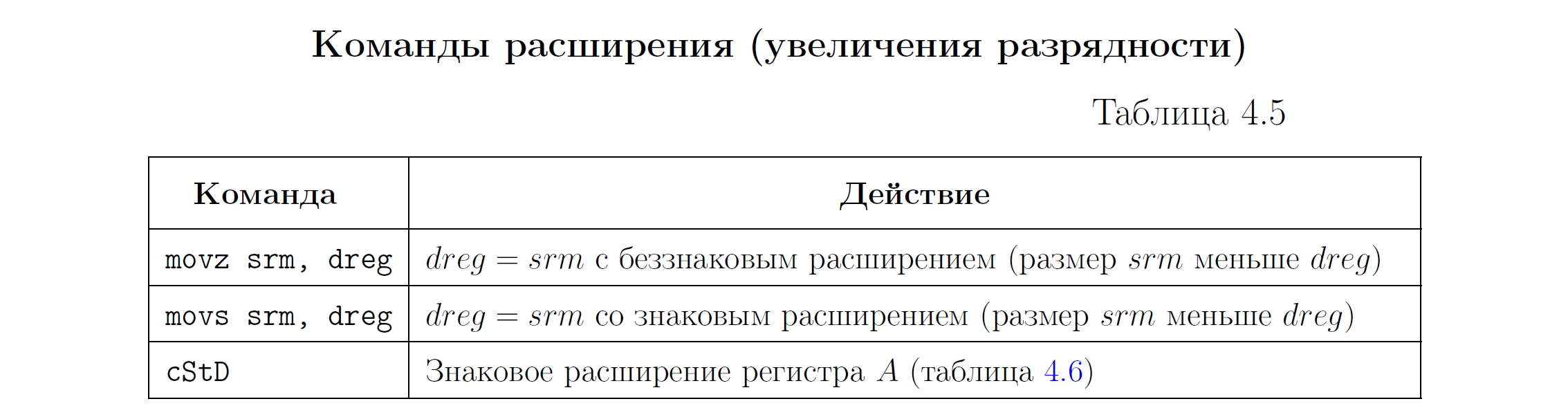
Команда call помещает в стек адрес следующей по порядку команды —

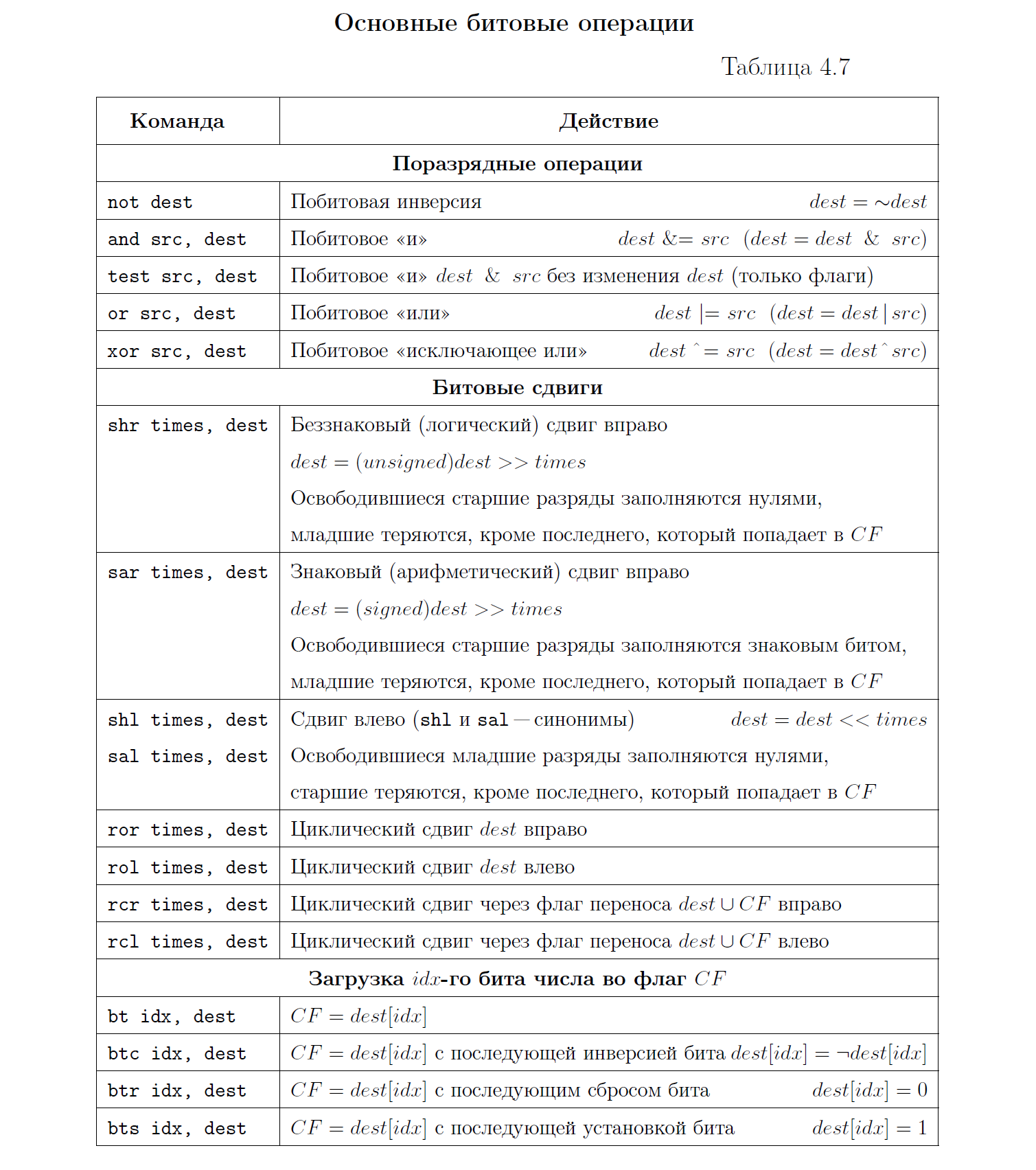
адрес возврата

*5. Какие вы знаете регистры общего назначения?*

*6. Какие вы знаете команды ассемблера x86?*







*7. Какие вы знаете флаги?*

