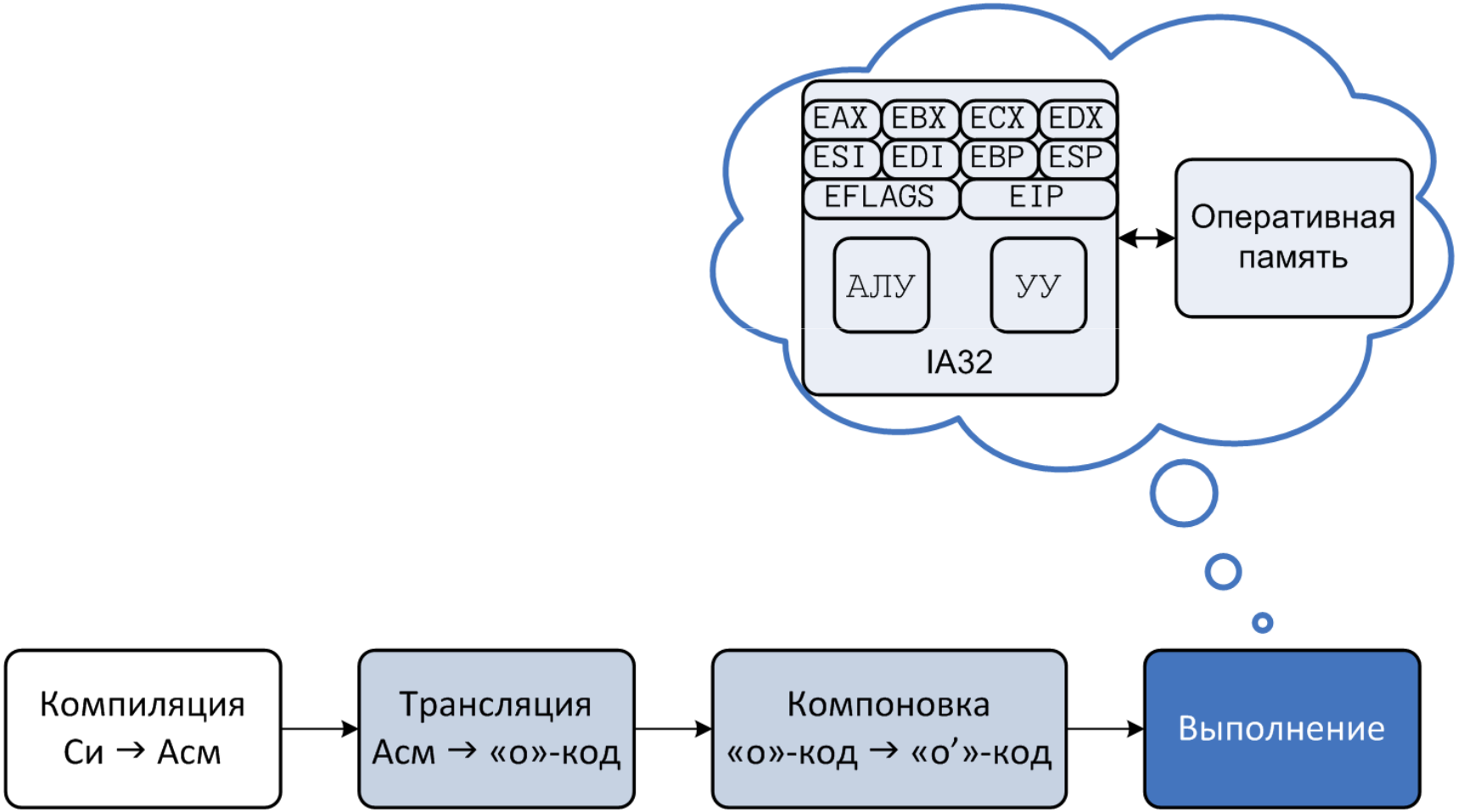
Лекция 5

22 февраля

Промежуточные итоги



Промежуточные итоги

- *Общего назначения*
- *x87 FPU*
- MMX
- SSE
- IA-32e команды 64-разрядного режима работы
- Системные инструкции
- Аппаратная виртуализация

- *Передача данных*
- **Команды двоичной арифметики**
- Команды двоично-десятичной арифметики
- **Логические команды**
- **Сдвиги и вращения**
- **Битовые и байтовые команды**
- *Передача управления*
- **Строковые**
- *Ввод/Вывод*
- *Явное управление EFLAGS*
- *Вспомогательные*

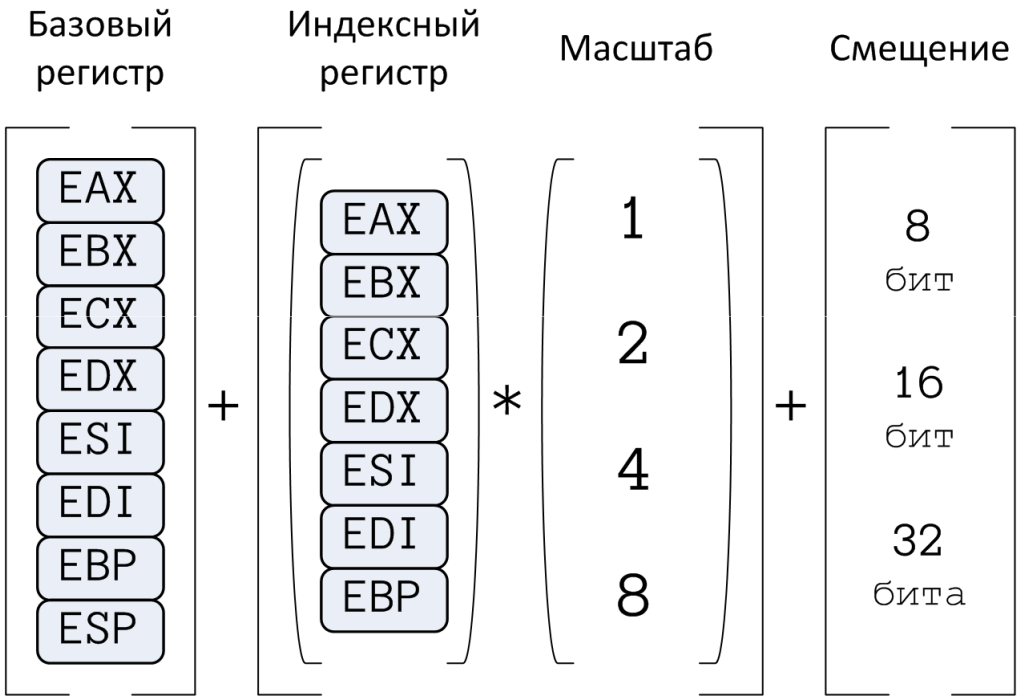
Промежуточные итоги

- MOV
- **XCHG**
- **BSWAP**
- ADD
- SUB
- NEG
- IMUL
- MUL
- IDIV
- DIV
- **INC**
- **DEC**
- MOVSX
- MOVZX
- CDQ
- CWD
- CBW
- JMP
- Jcc
- CALL
- RET
- PUSH
- POP
- **LEA**

Минимальный набор ассемблерных инструкций (на данный момент)

Общий вид адресного кода при обращении к памяти

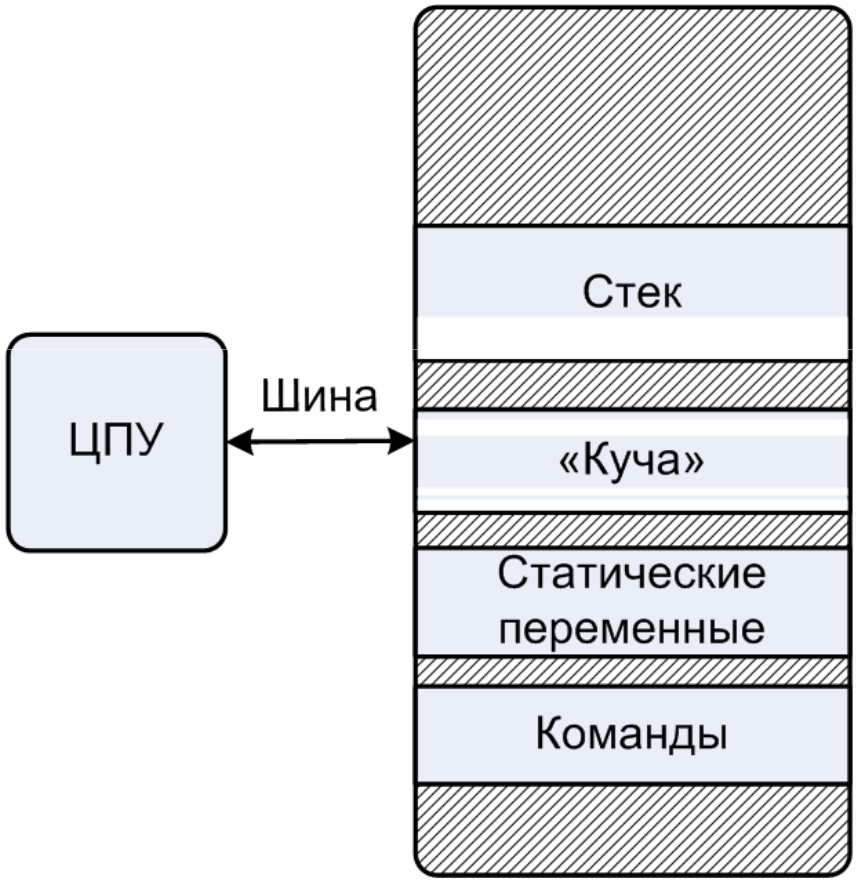
- LEA
  - r32, m



Исполнительный адрес = База + (Масштаб * Индекс) + Смещение

Распределение памяти

- Модели памяти в языке Си
  - Автоматическая
  - Статическая
  - Динамическая
- Секции кода и данных в языке ассемблера
  - .text
  - .data, .bss
  - .rodata



```
-bash-2.05b$ ./build_asm.sh hello.asm
-bash-2.05b$ objdump -h hello
```

```
%include "io.inc“
section .rodata
  msg db `Hello, world!\n`, 0
section .text
  global CMAIN
CMAIN:
  PRINT_STRING msg
  xor eax, eax
  ret
```

```
hello:     file format elf32-i386

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
...
 11 .text         000001cc  08048310  08048310  00000310  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
...
 13 .rodata       00000017  080484f8  080484f8  000004f8  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
...
 15 .data         00000008  08049514  08049514  00000514  2**2
                  CONTENTS, ALLOC, LOAD, DATA
...
 21 .bss          0000000c  08049610  08049610  00000610  2**2
                  ALLOC
...
```

Типы данных языка Си

- `char`
- Стандартные знаковые целочисленные типы
  - `signed char`
  - `short int`
  - `int`
  - `long int`
  - `long long int`
- Стандартные беззнаковые целочисленные типы
  - `_Bool`
- Перечисление
- Типы чисел с плавающей точкой
  - `float`
  - `double`
  - `long double`
  - `_Complex`
- Производные типы
  - Массивы
  - Структуры
  - Объединения
  - Указатели
  - Указатели на функции

Регистры и типы данных

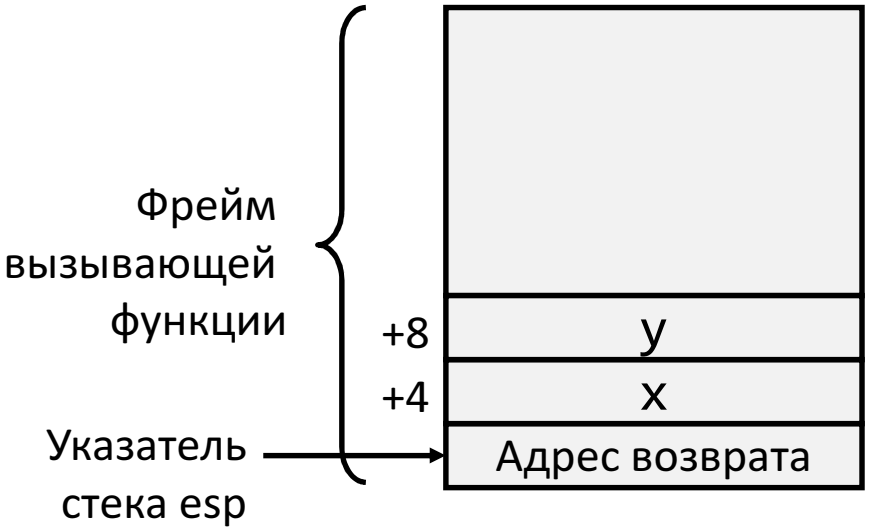
- Целые числа
  - Размещаются и обрабатываются в регистрах общего назначения
  - Знаковые/беззнаковые числа

Intel	ASM	Bytes	C
byte	**b**	1	**[unsigned] char**
word	**w**	2	**[unsigned] short**
double word	**d**	4	**[unsigned] int**
quad word	**q**	8	**[unsigned] long long int**

- Указатели
- Числа с плавающей точкой
  - Размещаются и обрабатываются в специализированных регистрах для чисел с плавающей точкой

Intel	ASM	Bytes	C
Single	**d**	4	**float**
Double	**q**	8	**double**

Пример: обмен значениями с использованием указателей




```
void exchange(int *x, int *y) {
   int tmp = *x;
   *x = *y;
   *y = tmp;
}
```

Фрейм намеренно не создается

```
exchange:
   mov eax, [esp+4] ; eax ← x
   mov edx, [eax]   ; edx ← *x
   mov ecx, [esp+8] ; ecx ← y
   mov ecx, [ecx]   ; ecx ← *y
   mov [eax], ecx   ; *x  ← ecx
   mov eax, [esp+8] ; eax ← y
   mov [eax], edx   ; *y  ← edx
```