



# Архитектура ЭВМ и Язык Ассемблера

## Семинар #2:

1. Изучение кодогенерации с помощью дизассемблера.
2. Регистры общего назначения в i386.
3. Арифметические операции: +, -, \*, /.
4. Сдвиги и знаковое расширение.
5. Задачи для самопроверки.

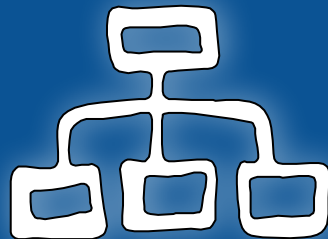


# Изучение кодогенерации с помощью дизассемблера





# Регистры общего назначения в i386



# Регистры общего назначения



## General-Purpose Registers

31	16 15	8 7	0	16-bit	32-bit
	AH	AL		AX	EAX
	BH	BL		BX	EBX
	CH	CL		CX	ECX
	DH	DL		DX	EDX
	BP				EBP
	SI				ESI
	DI				EDI
	SP				ESP

## General-Purpose Registers

31	0
	EAX
	EBX
	ECX
	EDX
	ESI
	EDI
	EBP
	ESP

## Segment Registers

15	0
	CS
	DS
	SS
	ES
	FS
	GS

## Program Status and Control Register

31	0
	EFLAGS

## Instruction Pointer

31	0
	EIP

# Назначение регистров общего назначения



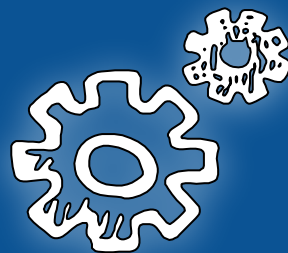
- **EAX** — Accumulator for operands and results data.
- **EBX** — Pointer to data in the DS segment.
- **ECX** — Counter for string and loop operations.
- **EDX** — I/O pointer.
- **ESI** — Pointer to data in the segment pointed to by the DS register; source pointer for string operations.
- **EDI** — Pointer to data (or destination) in the segment pointed to by the ES register; destination pointer for string operations.
- **ESP** — Stack pointer (in the SS segment).
- **EBP** — Pointer to data on the stack (in the SS segment).

Договорённость между разработчиками аппаратуры и  
разработчиками компиляторов.



# Арифметические операции:

$+$ ,  $-$ ,  $*$ ,  $/$



## MUL—Unsigned Multiply

Opcode	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
F6 /4	MUL r/m8	M	Valid	Valid	Unsigned multiply (AX := AL * r/m8).
REX + F6 /4	MUL r/m8 <sup>1</sup>	M	Valid	N.E.	Unsigned multiply (AX := AL * r/m8).
F7 /4	MUL r/m16	M	Valid	Valid	Unsigned multiply (DX:AX := AX * r/m16).
F7 /4	MUL r/m32	M	Valid	Valid	Unsigned multiply (EDX:EAX := EAX * r/m32).
REX.W + F7 /4	MUL r/m64	M	Valid	N.E.	Unsigned multiply (RDX:RAX := RAX * r/m64).

**Table 4-9. MUL Results**

Operand Size	Source 1	Source 2	Destination
Byte	AL	r/m8	AX
Word	AX	r/m16	DX:AX
Doubleword	EAX	r/m32	EDX:EAX
Quadword	RAX	r/m64	RDX:RAX

## DIV—Unsigned Divide

Opcode	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
F6 /6	DIV r/m8	M	Valid	Valid	Unsigned divide AX by r/m8, with result stored in AL := Quotient, AH := Remainder.
REX + F6 /6	DIV r/m8 <sup>1</sup>	M	Valid	N.E.	Unsigned divide AX by r/m8, with result stored in AL := Quotient, AH := Remainder.
F7 /6	DIV r/m16	M	Valid	Valid	Unsigned divide DX:AX by r/m16, with result stored in AX := Quotient, DX := Remainder.
F7 /6	DIV r/m32	M	Valid	Valid	Unsigned divide EDX:EAX by r/m32, with result stored in EAX := Quotient, EDX := Remainder.

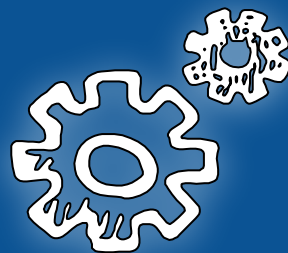
**Table 3-15. DIV Action**

Operand Size	Dividend	Divisor	Quotient	Remainder	Maximum Quotient
Word/byte	AX	r/m8	AL	AH	255
Doubleword/word	DX:AX	r/m16	AX	DX	65,535
Quadword/doubleword	EDX:EAX	r/m32	EAX	EDX	$2^{32} - 1$
Doublequadword/quadword	RDX:RAX	r/m64	RAX	RDX	$2^{64} - 1$



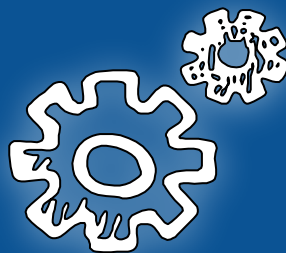


# Сдвиги и знаковое расширение





# Задачи для самопроверки



# Вопросы?

