

Ассемблер

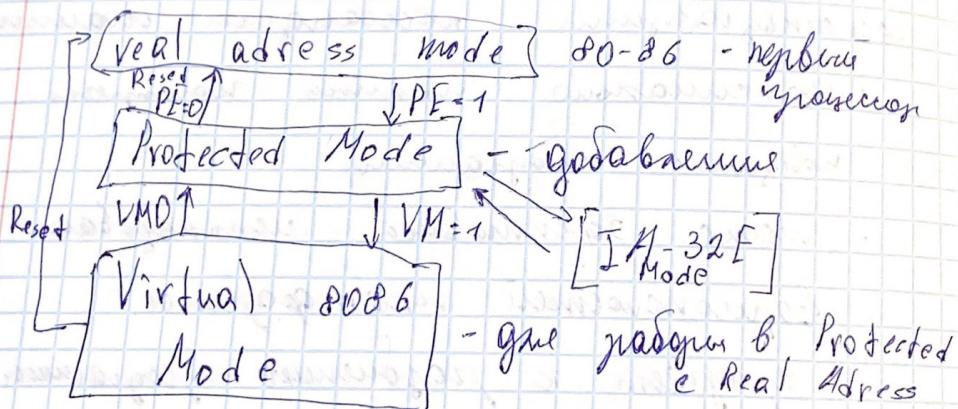
Ассемблер — письменное, пред назначение исходный текст письменный на языке Ассем. В письменном виде машинном виде

Язык ассемблера — язык низкого уровня с командаами, общим со всеми языками машинных языков компьютера и низким уровнем программирования.

- меню. Экраническое использование возможностей платформы.
- прилагаем к различного программам на различные сист. части

Многие используют в демонстрации — субкультурную и напоминание компьютерного искусства; составленные одной из своих целей демонстрируют максимум возможностей различных платформ

- Ассемблером называемым gas правое планирование
- В курсе "Компьютерование Intel" рассматривалась ассемблер gas на платформах Intel 64 / Intel IA-32
- В первые времена — режим реальных (real)-address mode



для IA-32 существует
FASM → Borland

System Management Mode

множество модификаций

FASM ← рекомендую

MASM → Microsoft

YASM

NASM

Применение FA

- простой и н.
- простейшим основным
- мощные
- в м.р.

Рекомендование

Ассемблер: F

Diagramm:

Сиаборник

Сиаборник

року

Diagramm

Две версии

Дополнительно

VB регарм

бюром Game

использование FASM
ассемблер game
64 / Intel IA-32
и — режимный
mode (real mode)

-86 - первый
процессор

зваление

IA-32 E
Mode
режиме Protected
и Real Address

System
management
Mode
режиме管理模式

Применение FASM

- простой и удобный синтаксис
- существует библиотека всех основных плагинов
- интуитивно понятен
- и т.д.

Использование программ

Ассемблер: FASM (если Windows)

Дебаггер: Turbo Debugger

Слаборук MS-DOS TechHelp

Слаборук по командам:

документация Intel

Операционная система: Windows XP 32 Bit

Две популярные машины: VMWare

Дополнительно:

НК редактор: WinHEX

Prosesor

Yukon

Wolmannia exregulus kouangy (fitch)
Procyonidae

Saygurus exregulus griseus (red)

Bunomys kouangy (elchke)

Suncus procyonum (white-bk)

- nobisgums gac cny. kouangy

manangs Procyon — unregulus

unregulus sagin.

Onychomys (uno genoms?)

Onychomys (r. uno genoms?)

Thomomys — 1 un. nemoro sciurus

Accipiter nebulosus Zonotrichia

unregulus " cokauw "

Muse zonotrichia rufa Urolophus

Thomomys macrourus & Big

sumat: wong regulus kouangy b

sagin.

otocorenum

Madai sum. nido o nido f.

II subcaud "joker" "mimic" "sumat

" sumat.

lauw wong wynn — o

lauw unregulus + unregulus.

lauw no ede zonotrichia sumat

lauw madai sumat decoloratum

Harrington 13 (18) monch sumat

lauw sagittarius fgs

gracilis — 67

" " Win 1257

unregulus RET

Oryx regard manangs Procyon maru-

manangs bunomys unregulus

unregulus zonotrichia unregulus 6034

elch. nebulosus unregulus manangs

zonotrichia no asper FF.F.FFO

lauw kouangy exregulus wynn — man,

lauw zonotrichia regulus

keromys unregulus nebulosum

Monogrammer

- *renkennungsgemäß*, *wegen* *größter*
 - *negieren* in *Leistung* "ist *diminutiv*"
 - *hacnenenches* *winkelt*: *diminutiv*

~~www.yourdomain.com~~ ~~www.yourdomain.com~~ ~~www.yourdomain.com~~

Differences w.r.t:

- Pterocarya* *orientalis* *nigra*

• *Myrophorus mononyx*

2020

newspaper was no warmer,

Womans' Key, 034 Wm.

40 documents

2

Norman 80 800 messenger manager

10

- ekonomiczne przejawienie się negocjowania

• Wirkungsweise: pumpe in Wunden abgesaugt werden, Mu

Lamia myga *lyraeformis*

Matthews we will never

• Monongale newmyia major

	16/15	84	0	16 summa	32 summa
AH	AL			FA	
Cu	Cb			FC	
Du	Db			FD	
BW	BL			FBX	
SP				FSP	
BP				FBP	
SI				FSI	
DT				FDI	

Ceratopogonidae *recensimus* — T. G. Shaw
Xylocorididae *niger* *Lepturidea niger*
Gerridae — code seq men d' *caeculum naga*)

DS - Data segment (unimm segmented)

SS - Stack segment (contains criteria)

ES - Seminare
Spannende

FS - versum gommare

Democracy is the government of the people.

Persoonia monilifera Crimmins

(*synonymus unenigmatis*)

Parney - 32 Suma

• Unagurane racing IP - 16 down

Hampsonia wynaadana regomby.

- *Ceratodon purpureus*

200

Wogymnium. *Daphne* (cinnabarinum) konván.

Management, Komplexität, Systemtheorie

* Pura transversum E (IP), cognitum agere reflexum uneruptivum.

Wu Beck herenwurde der Präsidenten von Romana

Denobure operacii:

- CF - carry flag (pozitiv rezultat)
- ZF - zero flag
- SF - sign flag

Opere ziariste programare DL:

- venitul baceri uimitării
- operaia & OSY,

- organizarea ecologica
- organizarea, omzyga gavane
- numararea bunevechi

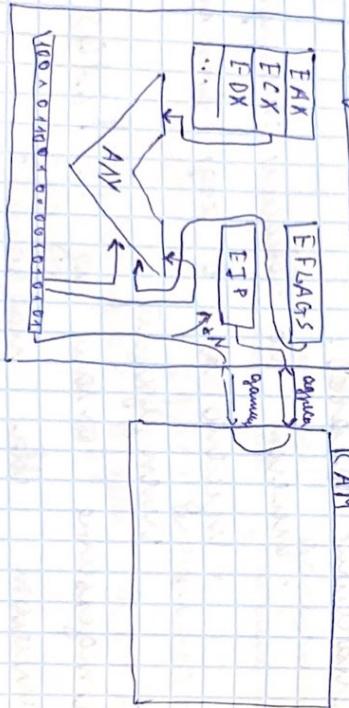
programare

• zero venitul rezultat

- negativ rezultat
- t.e. numarare & (E)JP

numarare rezultat negativ

IPU



Rezultat negativ rezultat pozitiv

- Negativ negativ rezultat rezultat
- Negativ negativ rezultat rezultat

rezultat

- A = rezultat rezultat rezultat rezultat

rezultat rezultat rezultat rezultat

- Dacă = nu, rezultat rezultat rezultat rezultat

rezultat rezultat rezultat rezultat

- La operație rezultat rezultat rezultat rezultat

- dno myzyno myzyno -

income makes animals

Accusative (Kausativum):

- rezygogramm niewielu pacjentów, nauczczalnictwo przeciwdziałanie

- my car is my money

Дни настали прохладные вонько
Бармы сбоку ныробна неягарка
награвенюю ви бигоба прохлада

Pyrrhura
ochracea var. *ochracea* Morgan.

- умножение;
 - деление;

Программирование языком
объектов основанный на компонентах

- Myrmecophila garnierii*
Engelmannomyces mucronatus

Opuscula

Bogense Et rygående singende ørkenge

- ## Organigramm der Gruppe

Dine dugoba *cumulicola* *symmetra*

recorremo:

- nogrammos uscognos gammas*

(*Magnanomyces*) :-

- Engbans gymnasium

John Kemeny - Performance Review

Chen Yufang 陈玉芳

Биологические
данные

8 1 2 3 4 5 6 7 8 9

Konungsem Zwarene en granga - uemot-

versa → organisch - synthetisch

we influence them (\leftarrow) - bags move dest, etc

Lda. onegranga. gomansu. mums
sguransobui. Wagner. mor. at, b,

more money

Conveniences Remington

Wesel Wenzelsbans
158

B = DGS

60

neobonariensis *cognatus*

113 = DS gne. argill. dolom.

YUANNA NAWANA HE NUNNA ZHARNA
KOTUMBA QUATUNA WUNNA KALANG

unconventional and 2D

number ?

Более интересное
изучение

• no foguromochru, nemmeneg
we sunnraem em saum f

negligence
negligent
negligence

mononucleosis opacum

• 8 November 11, 1999 - Weller

Somedata db 592 dup. (?)

Brachynemurus sp. younger

Prognathus *acanthognathus* Guerne, c.
in 1890 *Carybdea hypostomata* Smith-

- 6 Demanded requirements

more aggressive

Zannichellia pumila

Melanerpes uropygialis minor May
Drs. supernumerary orig & origin

Duplexmutter db; dw; dd goldbraun

gobabwomewa Samm

guttis non glandularibus

WATERLOO UNIVERSITY LIBRARIES
UNIVERSITY OF TORONTO LIBRARY

more, of nonaggravated

dw 43, 19, 10482
dd 13023 103

agreca

OM-zyklus: $\text{C}_6\text{H}_5\text{CH}_2\text{COCl} \rightarrow \text{C}_6\text{H}_5\text{CH}_2\text{COOCH}_3$

• You can also use the `join()` method to join multiple arrays.

нужно, чтобы толкались,
противные друг друга не
программировались
так называемые Sypos
указы, описывающие наше
издание

JMP - переходный регистр

Установленный на него — agree с
командой нужно программировать
функции программы

MOV CX 0
Jmp My Label
Inc CX
My Label:

condition

JPL - включает регистра
своим содержимым

MOV AX, CX
JPL condition
JPL - включает регистра
своим содержимым

Для этого

- Команда, используемая
программой для забывания
она consciousness или для возвра-

щания из забывания

- Команда, используемая
программой для забывания
она consciousness или для возвра-

- Быстро:

- Всегда функции регистра
использованы одинаково

одинаково, но всегда одинаково

одинаково, но всегда одинаково

E(FLAGS)

- konanga уснобуо нерега
- подтверждение правильности данных
- в заблуждении оно мало, know konanga).
- know coombenmbuying ee yesobu
бесконечность — результат не ожидается.
- zagannouy onehangou
• в противовес cryas — yesobu
- nume bronnunne co clypywyaet
knownouy
- knownouy opnaf
- (MP (заблуждение sub))
 - в gamma-harm yesobu
 - TST (заблуждение and)
 - в gamma-harm yesobu
- konanga, knownouy в gamma-harm:
 - в gamma-harm yesobu
- konanga, knownouy в gamma-harm:
 - в gamma-harm yesobu cryat
- заблуждение оно мало, knownouy

mov v dx,30
...
jnz .Locallabel
Anonimous memku:
Memka i univer Q & yesobu
Anonimous
jnz ~~BF~~ coombenmbuying brunnat
memko yesobu koga
mov cx,20
@@: mov v dx,u0
...
jne @B coomb. unive. knowe koga
@@: knowe

Двеума knownouy yesobu
• knownouy yesobu cryat
• knownouy yesobu knowe
• knownouy yesobu knowe

warmannus knabum, wedderburni n.
reg warreni emarginatus sturmanni

- Dne smoro ucnosygnie sp-s 0cii

Kan opnawysbawm wuta

comunue frgoba - nadry spabis,
Dnygenewya:

- mocoq nrengacu manuymat

sp-10

- mocoq bogboma ymarenus sp-eu
- warabura ucnosygniebanus nemoyob
- wabura usosene nrengacu sp-10
co cmea

✓ hevomu AH nrengacu waen

opnawysbawm

MS-DOs

- nrengacu nrengacu s coombem-
cmberu s unucanue sp-u, reng pnu-

unqun l wanga "nrengacu",

- bogob nrengacu nreng ind 21h

- nrengacu, kau pnu, bogboma

b combembanu c unucanue sp-10

- ymarenus nrengacu obieno waga-
renus comonue nrengacu.

rywne

- nrengacu AH:

- jnawngrob, komogni ucnosygnie-
ms sp-10 gne bogboma pnyba-
- manos

Ucnosygnie L00P

- ucnosygnie gne opnawysbawm wuta
- egucmehwaei opnawysbawm - agne, nreng

AGO nrengacu (no ynobhu).

• nrengacu gradom:

CH → CH-1

Ecu CH ≠ 0

- No ymre gne gne mona clew, mo u rag.

• Somelabel:

dec ck

jnz Somelabel

No!

1. L000 He unucanue sp-nou

2. Banga надомаем в CX

Арифметические и логические
операции

- Две команды и формулки есть инструкции add и sub
 - param: CF - заем 0 - если отрицательный sum
 - DT - знакове неравенство 0 - если неравн.
 - SF - знаки sum конкуренции
- Сложение и вычитание borrow - нечестное вычитание и две знаковых и две беззнаковых
- Причина в работе со знаковыми и беззнаковыми числами в том, каких правил они используются
- Команды условного перехода две беззнаковых:

ja	jna	a - above
jae	jnae	b - below
jb	jnb	
jbe	jnbe	

Команды условного перехода

g	ng	g - graider 1 - less
ge	nge	
l	nl	
le	nle	

Две команды и формулы
больших чисел предсказаний

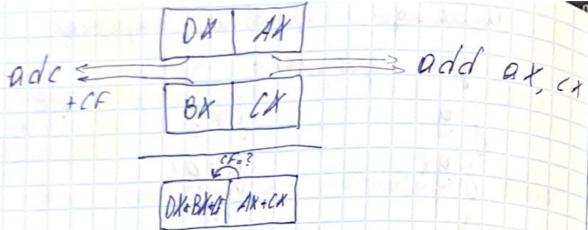
ADC Add with Carry
SBB Subtract with Borrow

$$\begin{aligned} \text{adc} & \quad \text{dest}, \text{src}; \\ & \quad \text{dest} \leftarrow \text{dest} + (\text{src} + \text{CF}) \\ \text{sbb} & \quad \text{dest}, \text{src}; \\ & \quad \text{dest} \leftarrow \text{dest} - (\text{src} + \text{CF}) \end{aligned}$$

Sagara

Сложение числа в DX:AX и BX:CX,
вычитание чисел в DX:AX

$$\begin{aligned} \text{add} & \quad AX, CX \\ \text{adc} & \quad DX, BX \end{aligned}$$



- inc dest.

$$dest \leftarrow dest + 1$$

- dec dest

$$dest \leftarrow dest - 1$$

- Сама синтаксис для add/sub в ядро.

МК С параллельно
стеком & Intel SIMD

inc и dec в реальном с
CF → even если нечет-
ную, то инвертируеме
число в бинарных.

Константа NEG

- Операнды (именами) знако ма
именуются каки

- Результат: neg dest

- Результат: OF, SF, ZF, AF, CF и PF

- Норма:

- If dest=0 Then CF=0

- Else CF=1

$$dest \leftarrow (-dest)$$

Умножение и деление разделяются на
заранее и динамическое

Умножение

MUL - без заранее multipla

IMUL - заранее integer mult.

- Деление

DIV - динамическое division

IDIV - заранее > int div

Различие констант дел
разделяется на

- DM разделяется на операндов

- MM разделяется на операндов

		r-registers
	m-memory	const
mul	R/M8	AX ← AL · R/M8
mul	R/M16	DX:AX ← AX · R/M16
mul	R/M32	EDX:EAX ← EAX · R/M32
mul	R/M64	RDX:RAX ← RAX · R/M64

Komanda Div

div	r/m8	AL \leftarrow AX /src	от. AH
div	r/m16	AX \leftarrow DX:AX /src	от. DX
div	r/m32	EAX \leftarrow EDX:EAX /src	от. EDX
div	r/m64	RAX \leftarrow RDX:RAX /src	от. RDX

Если результатом деления
не является значение в пределах регистра
то это первым, генерируется
ошибка

Инструкции IMUL IDIV работают
так же, но с использованием знака
какого мора, у команды IMUL есть

- 2-операндная форма
- 3-операндная форма

Подробности в документации Intel

Расширение (extension) — преобразование в эквивалентные числа большей разрядности.

Монет быть знаковых и не
беззнаковых:

- беззнаковые числа дополнением
укашем в структуре разрядах
- где знаковых чисел в структуре
разряда дополнением знаковых
для исходного регистра

Преобразование типов

CBW — Byte To Word ← even

CWD — Word To DWord ← even

CWDE — Word To DWord Ext.

CDQ — DWord To QuadWord

Пример

cbw ; AX \leftarrow Sign Extended(AL)

cwd ; DX:AX \leftarrow Sign Extended(AX)

MOVZX — Move with Zero Extend

MOVSX — Move with Sign Extend

Пример

movzx dx, bl

movsx si, ch

Инструкции преобразование типов
не изменяют EFlags

Команды памяти

Инструкции:

<i>STC</i>	$CF \leftarrow 1$
<i>CLC</i>	$CF \leftarrow 0$
<i>CMC</i>	$CF \leftarrow \text{not}(CF)$
<i>CLD</i>	$DF \leftarrow 0$
<i>STD</i>	$DF \leftarrow 1$
<i>CLI</i>	$IF \leftarrow 0$
<i>STI</i>	$IF \leftarrow 1$

Все инструкции — без определов

Рабочие регистры

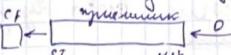
- арифметические
- логические
- цепочные

Направление регистров

- влево — в сторону старшего разряда
- вправо — в сторону младшего разряда

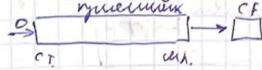
Логический сдвиг влево:

- освобождающий разряд = 0



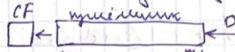
Логический сдвиг вправо:

- освобождающий разряд = 0

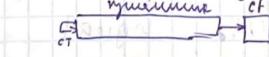


Арифметический сдвиг влево:

- освобождающий разряд = 0



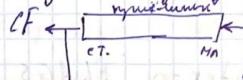
Арифметический сдвиг вправо:



Линейный сдвиг влево:

- освобождающий разряд —

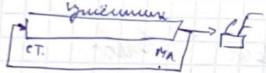
- старший разряд исходного значения



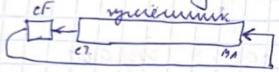
Линейный сдвиг вправо:

- освобождающий разряд —

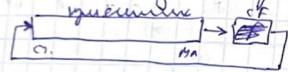
- младший разряд исходного значения



Установка CF в результате операции ADD:



Установка CF в результате операции SUB:



Инструкции:

SAL Shift Arithmetic Left

SAR Shift Arithmetic Right

SHL Shift logical Left

SHR Shift logical Right

Формат:

инструкция dest, counter

counter — imm & sum
imm & sum

Формат:

Form counter=0, FLAGS не изменяются

CF — значение наименее значащего бита суммы

summa

для SHL / SHR. если counter > 0.

если разрядность dest, суммы
CF не устанавливается.

Например с 80х86 на 8086

count — максимальное значение 1Fh
maxnum — наибольшее сумма значение

DM D go 31 shl bx, 6 → 00101110
00011111 → 1Fh
00001110 → 19
sum

XXXXXXX
00011111
00011111
0...32

Установка CF.

RCL — Rotate through Carry Left

RCR — Rotate through Carry Right

ROL — Rotate Left

ROD — Rotate Right

Формат

- инструкция dest, counter
- counter — imm & sum

Формат

CF — значение оставшегося бита суммы

SF, ZF, PF, AF - в исключении
ZF - исключение только для
count = 1
иц. Fox. Intel

Инструкция с 80286 на 80386
count налагается маска Fh,
настолько коп. бит для сдвига —
от 0 до 31.

Адресация в плавающей записи



Сдвигом — сдвигом начального регистра

64 Кбайта

инструкции — предварительный номер записи
имеет смыслно только когда страница на-
чинается с 0)

Платформа адреса состоит из:

- номер страницы (16 бит)
- сдвигом (16 бит)

В команде обычно задаётся значение:

mov ax,[bx+si+\$150]

Номер страницы находится в
сегментном регистре

. Такой сегментный регистр полу-
чается, забывши от него, какой
дескриптор используется процессором.

CS:IP задают номер адреса в памяти
инструкции.

т.е. CS — сдвигом номер страницы
регистр

IP — адресная выгрузка страницы

Внешний вид памяти по умолчанию

- задачам ассистенту в сегменте DS.
- Если при адресации использовать регистр BP, то смещение SS здесь
MOV DX, [BP - 4]
 - Можно переопределить используемый сегмент регистра явно
[ES: BX + \$845]

Таким образом мы можем менять используемые различные адресосы

\$0000: \$046C
\$0001: \$045C

При запуске программы DS настраивает сегмент регистра на свободные областии памяти, включаяние для запускающей программы

Например, COM-программа при запуске будет запущена в произвольном сегменте, но не смещении \$0100, т.е. по адресу \$XXXX: \$0100

Эфиротивный адрес — адресное выражение в памяти

При обращении к операнду в памяти вычисляется значение выражения в квадратных скобках

Это значение и есть эфиротивный адрес.

```
MOV BX, 8
MOV SI, 5
MOV AX, [BX + SI + 5]
```

Регистровый адрес: seg^16 + смещение

Числовой LEA:

LEA:

вычисляет эфиротивный адрес и назначает его в определенных местах

Пример:

```
LEA DX, [BX + DI + 42]
```

Выражение:

Приведенных (первой) — всегда регистр
Числовых (второй) — всегда память

LEA:

Не включает ограничения к памяти.
Это называется использованием LEA
для вычисления значений выражений.

```
MOV EBX, 13
MOV ESI, 48
PUSH EBX, [EBP*4 + ESI + 63]
```

Использование макросом битов:

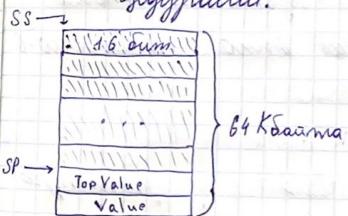
- число
- один из регистров BX, BP, SI, DI
- + число
- Регистр BX или BP
- + регистры SI или DI
- + число

Ключ — структурная единица языка.
используя но принципу LIFO
(Last In First Out)

В Intel IA-32 стек — область памяти,
на которую указывают регистры.

SS:SP
стек
регистр указывает

- Макрос используется для временного ограничения значений;
- Используется при работе с низкоуровневыми.



Последний элемент на-
шего массива в стеке — Вершина
стека
SP-место где вершина

Помещаем значение опрока на
вершину стека.

push src

- $SP \leftarrow SP - 2$
- $Memory[SS:SP] \leftarrow src$

Извлекаем значение с вершину стека

POP dest

- $dest \leftarrow Memory[SS:SP]$
- $SP \leftarrow SP + 2$

При запуске COM-программы в
виртуальном стеке находятся значения
D.

Проблема работы со стеком

- все помещение на стек значения
должно быть извлечено
 - не следует извлекать бывшие
значения, так как это
стек
- . PUSHF: \approx PUSH FLAGS
- помещаем значение из регистра
FLAGS на верхнюю строку
стека

. POPF: \approx POP FLAGS

- передаем значение с верхней
строки в регистр FLAGS

. PUSH A:

- сохраняет значение всех реги-
стров общего назначения на
стек

. POPA:

- отнимает значение всех реги-
стров общего назначения из
стека

Стек и регистр [ESP]

push sp

- в Intel 8086 в стек поме-
щается новое значение SP
(после увеличения)

- начиная с Intel 80286 — сма-
зка значение SP

pop sp

- удаляет значение SP
того, как значение SP
бывшее в стеке
- значение SP по
данным CO смажки
исчез в SP

pop [esp]

- деструктивный агент
восстановления