Лекция 3

15 февраля

Преобразования кода

Си

```
int x = 2;
```

Операторы и выражения

Компилятор

Ассемблер (язык)

```
mov     dword [ebp-12], 2
```

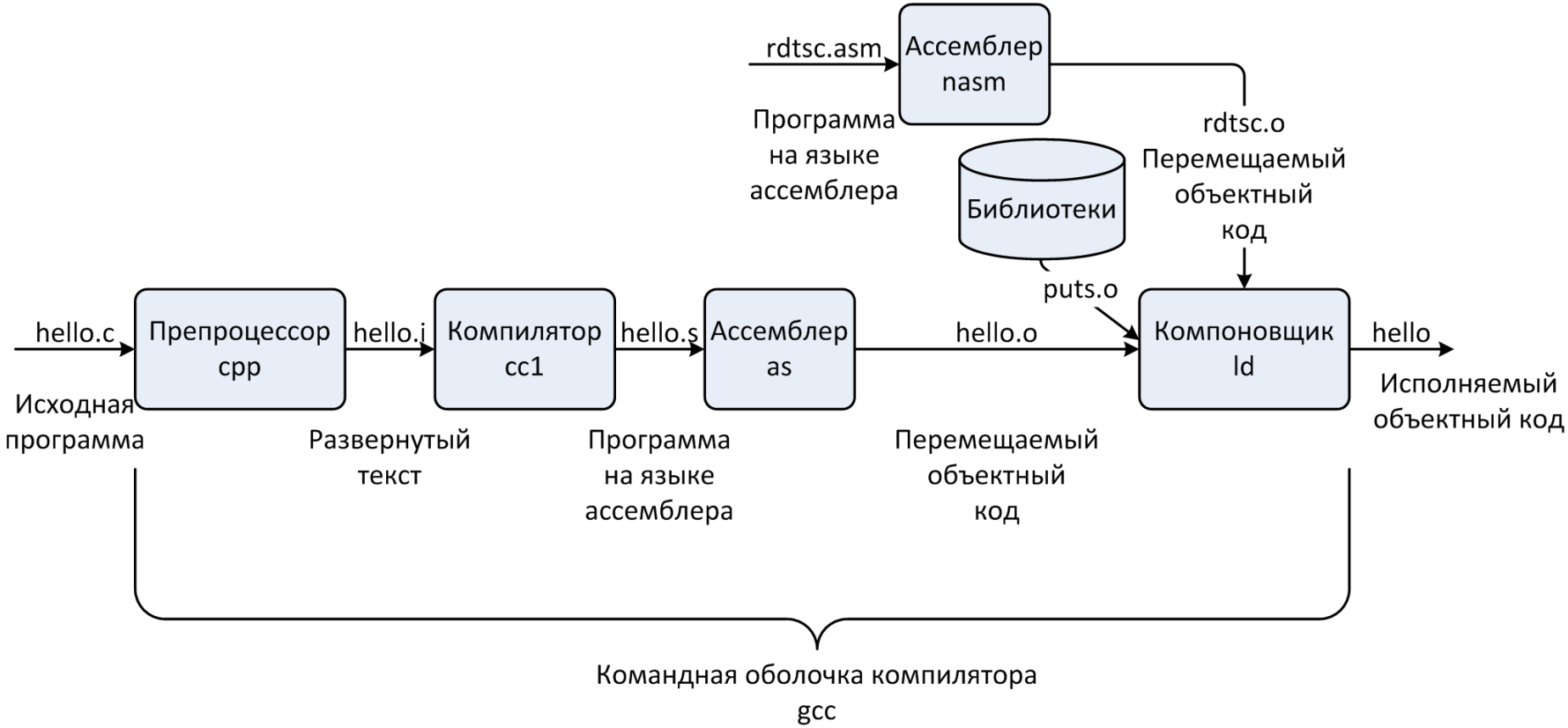
Ассемблерные инструкции

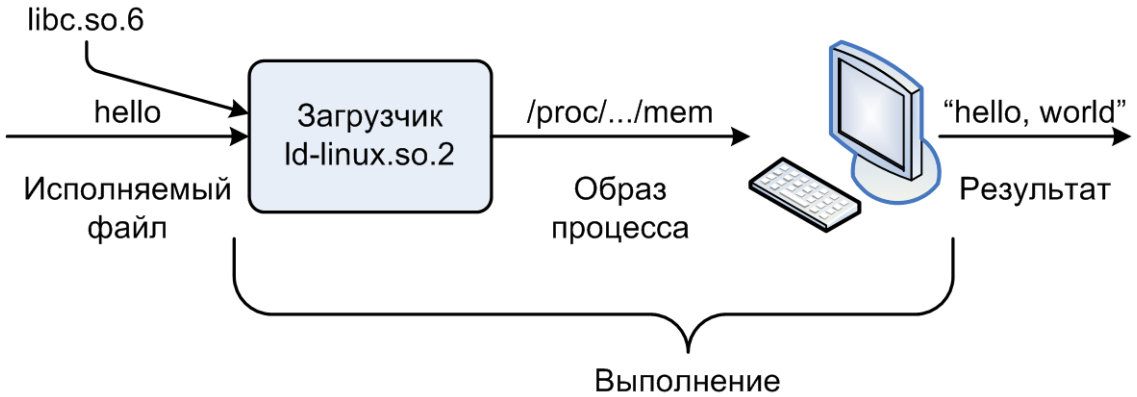
Ассемблер (программа-транслятор)

Машинный код

```
c7 45 f4 02 00 00 00
```

Машинные команды





Обратный путь – от исполняемого кода к читаемому тексту

snoop@echidna: ~/asm

```
snoop@echidna:~/asm$ nasm -f elf -o sample1.o sample1.asm
snoop@echidna:~/asm$ objdump -d -M intel sample1.o

sample1.o:     file format elf32-i386


Disassembly of section .text:

00000000 <f>:
   0:   55                      push   ebp
   1:   89 e5                   mov    ebp,esp
   3:   83 ec 10                sub    esp,0x10
   6:   c7 45 f4 02 00 00 00    mov    DWORD PTR [ebp-0xc],0x2
   d:   c7 45 f8 01 00 00 00    mov    DWORD PTR [ebp-0x8],0x1
  14:   c7 45 fc 00 00 00 00    mov    DWORD PTR [ebp-0x4],0x0
  1b:   66 c7 45 f2 64 00       mov    WORD PTR [ebp-0xe],0x64
  21:   c6 45 f1 0a             mov    BYTE PTR [ebp-0xf],0xa
  25:   83 05 00 00 00 00 01    add    DWORD PTR ds:0x0,0x1
  2c:   0f be 45 f1             movsx  eax,BYTE PTR [ebp-0xf]
  30:   0f b7 55 f2             movzx  edx,WORD PTR [ebp-0xe]
  34:   0f af c2                imul   eax,edx
  37:   0f af 45 f8             imul   eax,DWORD PTR [ebp-0x8]
  3b:   2b 45 f4                sub    eax,DWORD PTR [ebp-0xc]
  3e:   29 d0                   sub    eax,edx
  40:   89 45 fc                mov    DWORD PTR [ebp-0x4],eax
  43:   c9                      leave
  44:   c3                      ret
snoop@echidna:~/asm$
```

Синтаксис objdump отличается от nasm!

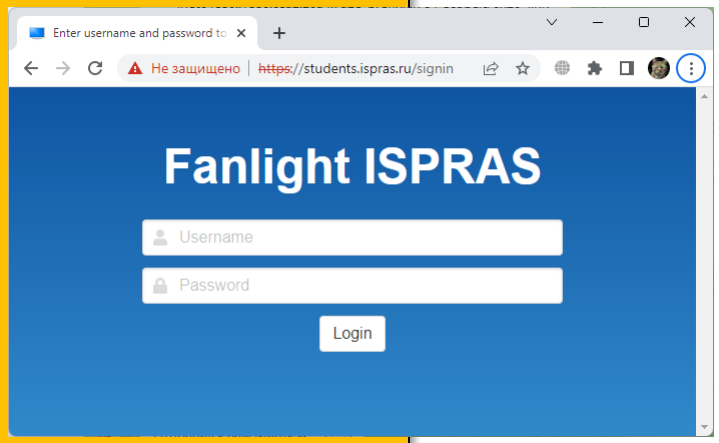
```
extern io_get_dec, io_print_hex, io_newline

section .data
 a    dw 1
 addr dd $
 var  dd 0x1234F00D

section .bss
   cntr resd 1

section .text
global main
main:
   call io_get_dec
   add dword [addr], eax
   mov eax, [addr]
   call io_print_hex
   call io_newline
   mov eax, addr
   call io_print_hex
   call io_newline

   xor eax, eax
   ret
```

Удобная среда разработки для учебных программ: vi, Emacs, Eclipse, …

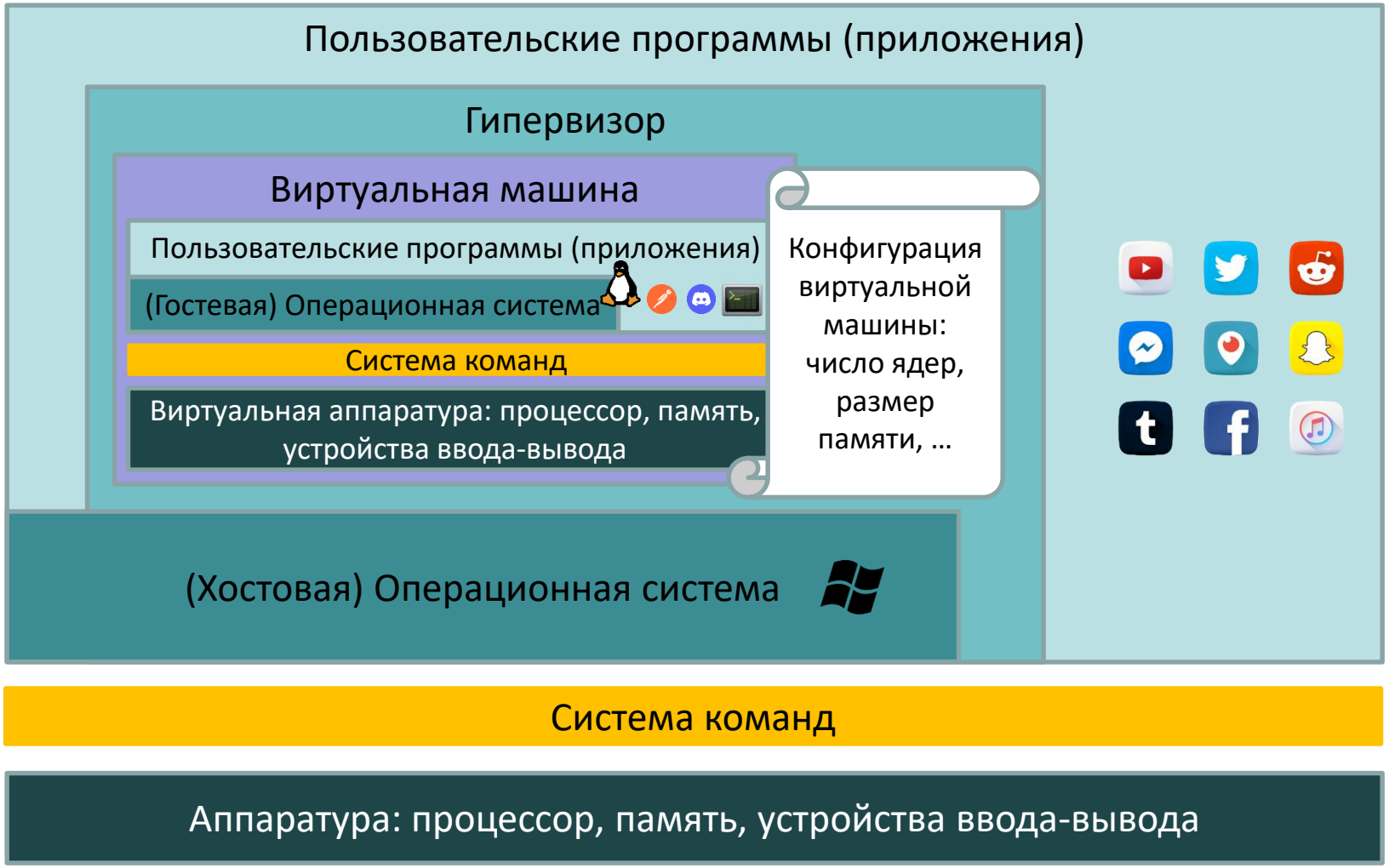
http://dman95.github.io/SASM/
https://github.com/Dman95/SASM

Для тех, у кого нет возможности работать в Linux x86, запущена облачная лаборатория https://students.ispras.ru

- Виртуальная машина с Ubuntu
- Развернут SASM
- Загружены методички
- Доступ через браузер



Виртуальная машина: ключевые понятия



Система команд

Аппаратура: процессор, память, устройства ввода-вывода

```
extern io_get_dec, io_print_hex, io_newline

section .data
 a    dw 1
 addr dd $
 var  dd 0x1234F00D

section .bss
   cntr resd 1

section .text
global main
main:
   call io_get_dec
   add dword [addr], eax
   mov eax, [addr]
   call io_print_hex
   call io_newline
   mov eax, addr
   call io_print_hex
   call io_newline

   xor eax, eax
   ret
```

Секция статических инициализированных данных

Секция статических неинициализированных данных

Секция кода

В тех случаях, когда размер других операндов позволяет определить размер операнда-памяти, допускается не указывать его спецификатор размера

Основные арифметические команды

- MOV
- MOVSX, MOVZX
- ADD, SUB
- INC, DEC
- NEG
  - r/m 8/16/32
- MUL
  - r/m 8/16/32
- IMUL
  - r/m 8/16/32
  - r 16/32, r/m 16/32
  - r 16/32, r/m 16/32, i 16/32
- DIV, IDIV
  - r/m 8/16/32
- CBW, CWD, CDQ