



Искать

Статьи

Форум

Исходники

Каталог

Хостинг

Справочник функций

Искать

Все справочники

Ваш аккаунт

Войти через:

Логин:

Пароль:

☐ Запомнить меня

Войти >>

Забыли пароль?

Регистрация

Информацию о новых материалах можно получать и без регистрации:

RSS   Twitter   ВКонтакте

Последние темы форума

CodeNet / Языки программирования / Ассемблер / Прерывания DOS и BIOS

CodeNet / Платформы / Другие платформы / DOS / Прерывания DOS и BIOS

Функции BIOS - INT 10H: видео сервис.

Здесь детализированы функции INT 10H стандартного видеосервиса ROM-BIOS.

АН сервис

-----

00H уст. видео режим. Очистить экран, установить поля BIOS, ус

вход: AL=режим

AL	Тип	формат	цвета	адаптер	адрес
===	=====	=====	=====	=====	=====
0	текст	40x25	16/8 полутона	CGA, EGA	b800
1	текст	40x25	16/8	CGA, EGA	b800
2	текст	80x25	16/8 полутона	CGA, EGA	b800
3	текст	80x25	16/8	CGA, EGA	b800
4	графика	320x200	4	CGA, EGA	b800
5	графика	320x200	4 полутона	CGA, EGA	b800
6	графика	640x200	2	CGA, EGA	b800
7	текст	80x25	3 (b/w/bold)	MA, EGA	b000
0dH	графика	320x200	16	EGA	A000
0eH	графика	640x200	16	EGA	A000
0fH	графика	640x350	3 (b/w/bold)	EGA	A000
10H	графика	640x350	4 или 16	EGA	A000

8,9,0aH режимы PCjr

0bH,0cH (резервируется для EGA BIOS)

замечание: для EGA и Jr можно добавить 80H к AL, чтс

инициализировать видео режим без очистки экрана

-----

01H уст. размер/форму курсора (текст). курсор, если он видим,

вход: CH = начальная строка (0-1fH; 20H=подавить курсор)

CL = конечная строка (0-1fH)

-----

02H уст. позицию курсора. установка на строку 25 делает курсор

вход: BH = видео страница

DH,DL = строка, колонка (считая от 0)

-----

03H читать позицию и размер курсора

вход: BH = видео страница

выход: DH,DL = текущие строка,колонка курсора

CH,CL = текущие начальная,конечная строки курсора (

-----

04H читать световое перо

Музыка — 2019-04-26 23:59:43 (1)

Общалка :) / Кристина Ивлеева

Как внести числа в сгенерированный массив чисел СЛУЧАЕЖДУ — 2019-04-25 11:15:30 (0)  
Visual Basic / wanx

рабочее зеркало гидры — 2019-04-22 13:27:49 (1)  
Операционные системы / Tony\_Montana

гидра анонимный — 2019-04-21 15:37:08 (0)  
Низкоуровневое программирование / Tony\_Montana

гидра браузер — 2019-04-21 11:49:42 (0)  
Компьютерная безопасность / Tony\_Montana

Закрытие главной формы из другой. — 2019-04-19 09:08:27 (1)  
Borland C++ Builder / Lazy\_Man

гидра оф сайт — 2019-04-18 13:15:50 (0)  
Студентам / Tony\_Montana

Информация про криптовалюту. — 2019-04-18 04:38:31 (6)  
Общалка :) / PotapShavanov

ссылка на гидру зеркало — 2019-04-17 11:53:53 (0)  
Базы данных / Tony\_Montana

гидра сайт доверенных — 2019-04-16 13:58:20 (0)  
Фриланс / Tony\_Montana

Delphi Trichview замена текста — 2019-04-16 13:02:35 (0)  
Delphi & Kylix / ale-sumkin

Реально ли заработать на ставках? — 2019-04-16 07:34:02 (5)  
Общалка :) / Polemik

[Показать новые сообщения »](#)

## Почтовая рассылка

Введите ваш E-Mail

- ☒ Подписаться  
☐ Отписаться

**Вперед!**

Подписчиков: -1

Последний выпуск: **19.06.2015**

вход: нет

выход: AH = триггер (0=нет значений; 1=возвращены значения)  
DH,DL = строка, колонка символа (текст)  
BX = колонка точки (графика)  
CH = строка точки (для графики EGA возвращается в C

05H выбрать активную страницу дисплея

вход: AL = номер страницы (большинство программ используют

06H листать окно вверх (или очистить). Листать на 1 или более  
вход: CH,CL = строка, колонка верхнего левого угла окна (с  
DH,DL = строка, колонка нижнего правого угла окна (с  
AL = число пустых строк,двигаемых снизу (0=очисти  
BH = видео атрибут, используемый для пустых строк

07H листать окно вниз (вдвинуть пустые строки в верхнюю часть  
вход: (аналогично функции 06H)

08H читать символ/атрибут в текущей позиции курсора

вход: BH = номер видео страницы

выход: AL = прочитанный символ

AH = прочитанный видео атрибут (только для текстов

09H писать символ/атрибут в текущей позиции курсора

вход: BH = номер видео страницы

AL = записываемый символ

CX = счетчик (сколько экземпляров символа записать)

BL = видео атрибут (текст) или цвет (графика)

(графические режимы: +80H означает XOR с симво

0aH писать символ в текущей позиции курсора

вход: BH = номер видео страницы

AL = записываемый символ

CX = счетчик (сколько экземпляров символа записать)

0bH выбрать цвет палитры/бордюра (CGA-совместимые режимы)

вход: BH = 0: (текст) выбрать цвет бордюра

BL = цвет бордюра (0-1fH; от 10H до 1fH - и

BH = 1: (графика) выбрать палитру

BL = 0: палитра green/red/brown

BL = 1: палитра cyan/magenta/white

0cH писать графическую точку (слишком медленно для большинства

вход: BH = номер видео страницы

DX,CX = строка, колонка

AL = значение цвета (+80H означает XOR с точкой на

0dH читать графическую точку (слишком медленно для большинства

вход: BH = номер видео страницы

DX,CX = строка, колонка

выход: AL = прочитанное значение цвета

0eH писать символ на активную видео страницу (эмуляция телетай

вход: AL = записываемый символ (использует существующий а

BL = цвет переднего плана (для графических режимов)

0fH читать текущий видео режим

вход: нет

выход: AL = текущий режим (см. функцию 00H)

АН = число текстовых колонок на экране  
 ВН = текущий номер активной страницы дисплея

#### 10H-12H См. EGA BIOS сервис

13H писать строку |AT| EGA Выдает строку в позиции курсора. (CarRet), 0aH (LineFeed), 08H (backspace) и 07H (Beep) тра команды управления и не высвечиваются.

вход: ES:BP => выводимая строка (спецформат для AL=2 и A  
 CX = длина строки (подсчитываются только символы)  
 DH,DL = строка, колонка начала вывода  
 ВН = номер страницы  
 AL = код подфункции:  
 0 = использовать атрибут в BL; не трогать кур  
 1 = использовать атрибут в BL; курсор - в кон  
 2 = формат строки: char,attr, char,attr...; н  
 3 = формат строки: char,attr, char,attr...; п

подфункции 10H-13H: EGA BIOS сервис                      данные BIOS                      функ

#### EGA BIOS сервис

Эти функции устанавливаются в процессе ROM-Scan при наличии пл  
 См. Детали видео сервиса подф. 00H о видео режимах, применимых  
 См. порты в/в EGA о прямом доступе к средствам платы EGA.

АН сервис

#### 10H установить регистры палитры для EGA

Выбирает цвета, используемые данным атрибутом экрана. умал  
 те же, что используются с CGA (см. атрибуты экрана ).

вход: AL = 0: уст. один регистр палитры  
           BL = регистр (идентифицирует 4-битовый цвет  
           ВН = значение регистра (6-битовый rgbRGB вь  
 AL = 1: уст. регистр бордюра. Note: not much of a b  
           ВН = значение регистра  
 AL = 2: уст. все регистры палитры и регистр бордюра  
           ES:BX => 17 байт (регистры палитры и регист  
 AL = 3: выбрать интенсивность или мерцание  
           BL = 0: интенсивность фона (16 возможных це  
           BL = 1: мерцание (8 цветов фона + мерцание  
 значения регистра палитры для подфункций 0,1,and 2:  
 --7-T-6-T-5-T-4-T-3-T-2-T-1-T-0--  
 |не исп.| r    g   b/I   R    G    B | (6 бит = 64 возмс  
 L---+---+---+---+T---+---+---+---

L= действует как интенсивность дл

#### 11H EGA character generator functions. Resets video variables to the current values in the regen buffer (ie, the screen

вход: AL = 0: load user-defined text-mode display font  
           ES:BP => user font table  
           CX = count of character patterns to store  
           DX = character offset (font load starts at  
           BL = font block to load (0 to 4; 4 if 256K  
           ВН = count of bytes per character pattern (  
 AL = 1: load ROM 8x14 character font  
           BL = font block to load (0-4)  
 AL = 2: load ROM 8x8 double-dot character font  
           BL = font block to load (0-4)  
 AL = 3: set block specifier (enables two of the fou

```

The second set of 256 chars is selected by
(in other words, if you select a "foreground
the EGA displays the secondary font).
BL = font block selector (4 bits, 0-0fH)
--7-T-6-T-5-T-4-T-3-T-2-T-1-T-0--
|   not used   |block# |block# |
L---+---+---+---+---+---+---+---
                                L=T=-   L==|= selected
                                L===== used when

```

Sub-functions 10H,11H,and 12H expect page 0 to be a be used directly after a mode reset. These recalcu columns and reprogram several CRTC registers, inclu registers.

AL=10H: load user-defined text-mode display font (s  
AL=11H: load ROM 8x14 character font (see SubFn 01H  
AL=12H: load ROM 8x8 double-dot character font (see

AL=20H: set up user character font for graphics mod  
ES:BP => 2048-byte table of 8x8 character s  
AL=21H: set up user character font for graphics mod  
ES:BP => table of 8x8 character set for INT  
CX = points (bytes per character)  
BL = screen rows code  
0 = user-specified  
DL = character rows on screen  
1 = 0eH (14 character rows on screen)  
2 = 19H (25 character rows on screen)  
3 = 2bH (43 character rows on screen)

AL=22H: set up ROM 8x14 character font for graphics  
BL = screen rows code (see above)  
AL=23H: set up ROM 8x8 double-dot character font fc  
BL = screen rows code (see above)

AL=30H: Get EGA information  
BH = table pointer request code  
0 = return ES:BP => current INT 1fH gr  
1 = return ES:BP => current INT 44H gr  
2 = return ES:BP => ROM 8x14 font tabl  
3 = return ES:BP => ROM 8x8 double dot  
4 = return ES:BP => ROM 8x8 double dot  
5 = return ES:BP => ROM 9x14 alternate  
выход: CX = font points (font-definition bytes per  
DL = screen rows

#### 12H EGA special functions (Alternate Select)

вход: BL=10H: return EGA information  
выход: BH = mode (0=color; 1=monochrome)  
BL = memory size code (0=64K; 1=128K; 2=192  
CH = feature bits (values of those unused R  
CL = switch settings (adaptor/display prima  
вход: BL=20H: select alternate print-screen routine (be s  
every time you change the number of screen

13H write string |AT| EGA Displays a string at a selected cu  
Characters 0dH (CarRet), 0aH (LineFeed), 08H (bksp), and 0  
treated as control commands and are not displayed.  
вход: ES:BP => string to display (special format for AL=  
CX = length of string (character count only)

DH,DL = row,column to start displaying

BH = page number

AL = sub function code:

0 = use attribute in BL; don't update cursor

1 = use attribute in BL; leave cursor at end

2 = string format: char,attr, char,attr...; c

3 = string format: char,attr, char,attr...; c

## Оставить комментарий

Комментарий:

можно использовать BB-коды

Максимальная длина комментария - 4000 символов.

CodeNet

ВКонтакте

Facebook

Twitter

Google

Яндекс

Добавить

## Комментарии

1.

13 февраля 2016 года **97K**

**NUKE** **Adam Smith**  
DEV 0 / / 13.02.2016

+3 / -1



13 февраля 2016, 17:56:33

Неудобно смотреть

2.

20 октября 2010 года **65K**

**S\_Grey**  
0 / / 20.10.2010

+4 / -2



20 октября 2010, 21:47:49

А если мне нужно нажатием клавиши F6 увеличить экран до 43 строк. а при повторном нажатии этой клавиши вернуть обратно 25, это прерывание подойдет?

3.

11 февраля 2008 года **36K**

**stanger**  
2 / / 11.02.2008

+9 / -4



11 февраля 2008, 19:44:36

Побольше бы таких статей

4.

29 ноября 2005 года 14K



Gue

1 / / 29.11.2005

+6 / -2



2 марта 2007, 16:46:39

Замечательно!

Давно искал. А скажите, есть где-нибудь какая-нибудь официальная справка по прерываниям? Пусть на английском.

5.

Аноним



+6 / -1



7 апреля 2006, 11:32:13

Достаточно полное руководство. Не хватает только "новых" функций VBE. И видеорежимов VGA.

6.

Аноним



+7 / -1



20 января 2006, 19:26:13

Хорошо помогло при одном из первых экспериментов с Assembler'ом. Огромное спасибо!!!!!!!!!!!!!!

[Реклама на сайте](#) | [Обмен ссылками](#) | [Ссылки](#) | [Экспорт \(RSS\)](#) | [Контакты](#)  
[Добавить статью](#) | [Добавить исходник](#) | [Добавить хостинг-провайдера](#) | [Добавить сайт в каталог](#)