

WYOS - Выпуск № 16

Дальнейшие эксперименты, FAT12

commrade, Четверг, 01 Июль 2004, 00:04

Напиши свою ОС! #16

Привет вам meine liebe! How do you do? If you well done, then go to the Moon and back!

Перевод: Привет мои дорогие! Как у вас дела? Если все порядке, тогда предлагаю прочитать нашу рассылку и затем вернуться снова в реальность.

Сегодня 21 декабря 2003 года XXI век н.э. Последние события:

- 1) Президент РФ (надеюсь вы еще помните кто это такой) ответил на вопросы россиян.
- 2) Космический аппарат (к сожалению забыл как называется, если кто помнит напишите) европейского космического агентства достиг 4 планеты Солнечной системы - Марса.
- 3) На прошлой недели "" показало фильм "Секретные " - истинна попрежнему где-то рядом.
- 4) После недельного отсутствия с вами вновь ваш commrade (толпа ликует, девушки рвут на себе одежду, скидывают юбки и скандируют "**Commrade foreve! I LOVE YOU COMMRADE! I want 30S or day or night!**") и он продолжит рассказ про технику написания ОС.

Ладно это все была лирика, а теперь серьезно! Здравствуйте мои дорогие!

Вывод сообщений при загрузке

Наконец-то у меня получился код загрузочного сектора для флоппи дискетка. Напомню в прошлых выпусках мы обсуждали вопрос, как загрузив компьютер с дискетки вывести на экран монитора какое-нибудь сообщение. Некоторые могут сказать "Чувак че-то ты гонишь! Че ты прицепился к этому вопросу. Че других тем нет!" Все дело в том, что когда ваша ОС загружается, есть необходимость выводить информацию о том какие модули ОС уже загрузились, какие ошибки случились в процессе загрузки. Для и необходимо понимание того как вывести сообщение на экран используя только BIOS. И так окончательный вариант, этого вопроса представлен ниже.

```
<code style="color:green">
<pre>
org 7C00h
msg      db "Hello, peppers",0
video_mode db 02h
video_page db 02h
```

start:

sel_vidmode: *<i style="color:gray">; выбираем режим </i>*

```
xor    ah,ah
mov     al,[video_mode]
int     10h
```

sel_actvidpage: *<i style="color:gray">; выбираем активную страницу видеопамя</i>*

```
mov     ah,5
mov     al,[video_page]    <i style="color:gray">; выбираем страницу video_page</i>
int     10h
```

```
mov     ah,6
mov     al,0
mov     bh,0ffh
mov     cl,10
mov     ch,5
mov     dl,30
mov     dh,15
int     10h
```

sel_curcorcur: *<i style="color:gray">;устанавливаем координаты </i>*

```
mov     ah,2
mov     bh,[video_page]    <i style="color:gray">; выбираем страницу video_page</i>
mov     dh,12              <i style="color:gray">; строка 12</i>
mov     dl,20              <i style="color:gray">; колонка 20</i>
int     10h
```

<i style="color:gray">;отображаем на активной странице экрана строку msg</i>

```
lea     si,[msg]
call    WriStr
```

```
ret
```

WriStr: *<i style="color:gray">;выводим на экран строку с </i>*

<i style="color:gray">; определяем текущую активную страницу </i>

```
mov     ah,0Fh
int     10h    <i style="color:gray">; номер активной страницы в регистре bh</i>
```

<i style="color:gray">; получаем в dx координаты </i>

```
mov     ah,3
int     10h
```

next_sym: *<i style="color:gray">; в цикле выводим на экран символы строки </i>*

```
mov  ah,9
lodsb
cmp  al,0    <i style="color:gray">; находим конец </i>
je   exit_proc <i style="color:gray">; если конец строки достигнут, то оканчиваем </i>
mov  bl,7    <i style="color:gray">; для отображаемых символов используем </i>
mov  cx,1    <i style="color:gray">; символы строки выводятся без </i>
int  10h
```

<i style="color:gray">; перемещаем курсор в следующую позицию </i>

```
mov  ah,2
inc  dl
int  10h

jmp  next_sym
```

exit_proc:

```
ret
</code>
</pre>
```

Об утилите записи загрузочного кода на флоппи дискету

<p>В процессе подготовки рассылки я получил письмо от Dmitri Alexeeenko, в котором он привел код утилиты записи заг. кода на дискетку. С его разрешения я помещаю этот код в рассылку. От себя могу только добавить, что мы ни в коем случае не навязываем, то или иное решение проблемы, решение этого вопроса зависит только от ваше желания. Так же мы не навязываем каких-бы то нибыло инструментов (Builder, FASM, TASM), что выбрать для решения возникших перед вами проблем зависит только от ваших желаний и возможностей.</p>

```
<code style="color:green">
<pre>
.286
.model tiny
.code
org 100h
entry:
```

```

mov    ah,00
mov    dl,01
int    13h      <i style="color:gray">;reset controller</i>
jc     error    <i style="color:gray">;carry should be clear</i>

mov    si,03h    <i style="color:gray">;number of tries</i>
over:                <i style="color:gray">;'cos the drive should have some time to spin up</i>
mov    ax, 0301h  <i style="color:gray">;write one sector</i>
xor    cx,cx
inc    cl        <i style="color:gray">;on first track </i>
xor    dx,dx      <i style="color:gray">;head 0, drive 0</i>
lea    bx,writeit <i style="color:gray">;code address</i>
int    13h
jnc    ok        <i style="color:gray">;written succesfully?</i>
mov    ah,00      <i style="color:gray">;if not, reset</i>
mov    dl,01
int    13h
dec    si
jnz    over      <i style="color:gray">;and start over again </i>
jmp    error     <i style="color:gray">;failed</i>
ok:
int    20h      <i style="color:gray">;terminate</i>
error:
mov    ah, 09h
lea    dx, mes
int    21h
mov    ax,4c01h
int    21h
writeit:
mov    ax, 0b800h  <i style="color:gray">;this code will be written to destination</i>
mov    es,ax
xor    di,di
mov    ax,5902h    <i style="color:gray">;a face on fancy bkground</i>
stosw      <i style="color:gray">;note it's a word, not a byte</i>
jmp     □
wrsiz=□-writeit
data   db (510-wrsiz) dup (90h)
db     55h, 0aah    ;signature
mes    db 'Error occured.□'
end     entry
</pre>
</code>

```

Еще раз об инструментах

В своих письмах вы обращаете мое внимание на то, что ни у каждого есть возможность писать код прелагаемый мной на Builder. ОК. Я внял вашим письмам. Предлагайте какой компилятор C++ лучше использовать. Требования к компилятору C++: небольшой размер

<http://www.osrc.info/plugins/content/content.php?content.20>
 дистрибутива, возможность закачки по Internet, freeware. Пиши мне на мыло. Мыло в низу, в
 тазике (шутка). Мой e-mail приведен в низу данной страницы, меня если вы не забыли зовут
 comrade.

Что же дальше

Ну хорошо ну вывели мы на экран надпись-приветствие, а что потом? Схема загрузки ОС
 обычно следующая:

```
<span align='Center'><b><font
size="+1">BIOS&mdash;&mdash;&raquo;BootRecord&mdash;&mdash;&raquo;LoaderOS</font></b
></span>
```

Т.е. после приветствия мы должны найти на дискетке файл, который загрузит и соберет ядро
 нашей ОС, пусть будет называться "loader.com". Как же это сделать?

Все просто и не просто одновременно. Работать с дискетой будем через BIOS, а именно через
 прерывание 13H. Для чтения сектора с диска используется функция 02H, где параметры
 функции следующие:

```
<ul>
<li>вход:
<ul>
<li>DL = номер диска (0=диск A...; 80H=тв.диск 0; 81H=тв.диск 1)</li>
<li>DH = номер головки чтения</li>
<li>CH = номер дорожки (цилиндра)(0-n)</li>
<li>CL = номер сектора (1-n)См. замечание ниже.</li>
<li>AL = число секторов (в сумме не больше чем один цилиндр)</li>
<li>ES:BX => адрес буфера вызывающей </li>
<li>0:0078 => таблица параметров дискеты (для гибких </li>
<li>0:0104 => таблица параметров тв.диска (для твердых дисков)</li>
</ul>
</li>
</ul>
<ul>
<li>выход:
<ul>
<li>Carry-флаг=1 при ошибке и код ошибки диска в AH</li>
<li>ES:BX буфер содержит данные, прочитанные с </li>
</ul>
</li>
</ul>
<pre>замечание: на сектор и цилиндр отводится соответственно 6 и 10 бит:
1 1 1 1 1 1
+5-4-3-2-1-0-9-8-7-6-5-4-3-2-1-0+
CX: |c c c c c c c c C c S s s s s s|
+-+--+--+--+--+--+|--+--+--+--+--+--+--+
+=====> исп. как старшие биты номера </pre>
</p>
```

<p>Ну так вот я продолжаю. Используя прерывание 13H и функцию 02H может прочесть

любой сектор как с гибкого

диска так и с "" диска. Для того чтобы прочесть данные с гибкого диска нужно немного знать структуру файловой системы FAT12.</p>

FAT12

FAT12 - файловая система, в настоящее время в основном используется для дискет формата 3,5". Структура FAT12 следующая:

-
- Загрузочная
- FAT (таблица) - в FAT находятся списки кластеров, принадлежащих файлам. Все свободные кластеры отмечены нулями.
- Область корневого каталога (не существует на носителях с FAT32)
- Область данных файлов и
-

Таким образом, если файл занимает несколько кластеров, то эти кластеры связаны в список. При этом элементы таблицы FAT содержат номера следующих используемых данным файлом кластеров. Конец списка отмечен в таблице специальным значением. Номер первого кластера, распределенного файлу, хранится в элементе каталога, описывающего данный файл. Итак, FAT - массив информации об использовании кластеров диска, содержит списки кластеров, принадлежащих файлам. Номера начальных кластеров файлов хранятся в каталогах. По крайней мере в учебнике Фролова так описывается FAT.

Также по Фролову алгоритм работы с дискетой, следующий:

-
- Читаем FAT в память
- Умножить номер начального кластера на 3 (так как FAT12 это 12-разрядная система, т.е. для представления данных тут используется 3 байта)
- Разделить результат на 2 (так как каждый элемент таблицы имеет длину 1,5 байта)
- Прочитать 16-битовое слово из FAT, используя в качестве смещения значение, полученное после деления на 2
- если номер начального кластера четный, на выбранное из FAT слово надо наложить маску 0FFFh, оставив младшие 12 бит, если же номер начального кластера нечетный, выбранное из FAT значение необходимо сдвинуть вправо на 4 бита, оставив старшие 12
- Полученный результат - это номер следующего кластера в цепочке, при этом значение 0FFFh (или другое в диапазоне от 0FF8h до 0FFFh) соответствует концу цепочки
-

Вот тут я сам немножко запутался весь вопрос в том, откуда взять этот начальный кластер, вроде по учебнику получается, что информация о начальном кластере находится в корневом каталоге следом за именем файла. Но я хоть убейте меня так и не смог найти информацию о

том каким по порядку байтом идет этот номер. Вроде, на имя файла отводиться 8 байт, еще 3 байта на расширение файла, итого 11 байт, дальше вроде там идет некая служебная информация, а затем идет вот тот вот самый начальный кластер. Кто-нибудь знает сколько байт занимает вот эта служебная информация? Если знаете - напишите мне, очень поможете!!!!

Код чтения сектора диска

Вот теперь я приведу вам пример кода на ассемблере для чтения сектора диска (в частности для чтения файловой таблицы дискетки FAT):

```
<code style="color:green">
<pre>
FAT rb 3072*2

mov dh,00h  <i style="color:gray">; номер диска - 0</i>
mov dl,00h  <i style="color:gray">; номер головки чтения/записи - 0</i>
mov ch,00h  <i style="color:gray">; номер дорожки (цилиндра)(0-n) - 0 </i>
mov cl,01h  <i style="color:gray">; номер начального сектора - 1 </i>
mov al,512  <i style="color:gray">; число читаемых секторов - 512</i>
les bx,[FAT] <i style="color:gray">; звяжем переменную FAT с es:bx, сюда
;13-ое прерывание запишет нужные нам данные </i>

mov ah, 02h <i style="color:gray">; вызываем функцию чтения сектора диска </i>
int 13h    <i style="color:gray">; вызываем прерывание дискового ввода/вывода </i>
</pre>
</code>
```

Вот такой вот простой пример кода. Мало? А вы что думали commrade будет все за вас писать, а вы только читать и кофе пить. Нетужки. Серый Волк против!!!! Это вам на закуску.

Ну а на последок я задам несколько упражнений для желающих поломать голову.

Упражнение 4

Напишите пример кода с комментариями для решения следующей задачи: нахождение и чтение данных из файла на дискете используя алгоритм Фролова.

Упражнение 5

Объясните что такое дескриптор файла и какая у него структура.

Ну вот и все на сегодня. Ах да не удивляйтесь если ваша дискетка с экспериментальны загрузочным сектором вдруг не читается под Windows. Все дело в том, что данная ОС не понимает дискетки с не-Microsoft структурой загрузочного сектора.

<p>Хочется так же сказать спасибо все тем кто оказал мне помощь при написании статьи и помого разобраться с кодом:

Zensor,
 Sashok Sam,
 Dmitri Alexeenko,
 arkan и другим кого я по своей забывчивости забыл

Всем большое человеческое спасибо.</p>