Kolibri OS INSIDE

#2 MARCH 2020

Журнал выпускается раз в два месяца и может иметь разное количество страниц

Kolibri OS Inside – что это?

Журнал, рассказывающий о новостях мира Kolibri OS. Выпускается в последние числа каждого второго месяца.

Над номером работал – Гордон Фримен.

Пожалуйста, помогите мне в работе! Я не могу одновременно заниматься вёрсткой жунала и написанием статей. Впрочем, помощь разработчикам будет ещё полезнее!

Сайт https://vk.com/kosinside

В 2004 году от MenuetOS отсоединилась команда разработчиков. Они создали форк под названием KolibriOS. Kolibri - очень миниатюрная и невероятно быстрая операционная система. Её основной дистрибутив занимает 1,44 Мб - и это при том, что содержит набор драйверов, браузер, текстовый процессор, графический редактор, просмотрщик, более 30 игр и другие программы.

Такая скорость и компактность достигается благодаря тому, что ядро и большинство программ написаны на ассемблере, тем самым максимально оптимизированы под процессоры x86. Для запуска достаточно всего 8 мегабайт оперативной памяти! Мечтали вы когда-нибудь о том, чтобы приложения запускались мгновенно, сразу после нажатия на иконку? Без крутящихся часиков и кружков. Просто сразу. Попробуйте Kolibri и сравните её с такими тяжеловесами, как Windows и Linux!

Надеемся, вам понравится,

KolibriOS Team

Microsoft решила с мая этого года временно прекратить выпуск обновлений для Windows 10

В связи с пандемией коронавируса корпорация прекратит выпускать апдейты C и D, отвечающие за добавление различных исправлений и улучшений. Выпускаться будут только обновления марки B, связанные с безопасностью.

Intel работает над связкой видеокарт

Если ещё конкретнее, речь о связке из встроенного и дискретного чипов GPU в ноутбуках. Была продемонстрирована симуляция n-тела, состоящего из 4 млн. частиц. При этом активно использовались возможности Direct X 12.

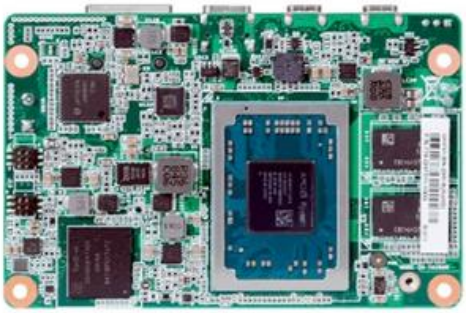
Ранее такую технологию разрабатывала AMD, однако их постигла неудача в плане переключения со встроенной графики на дискретную, и наоборот.

Google вслед на Apple отменяет Google I/O

В связи с пандемией коронавируса корпорация отменяет ежегодную конференцию для разработчиков Google I/O, которая должна была состояться 12 мая. Кроме этого, не будет презентован даже Android 11 и новые смартфоны линейки Pixel, поскольку Google, вопреки ожиданиям не станет проводить конференцию в онлайн-режиме.

Представлен прокаченный Raspberry Pi

Тайваньская компания DFI представила одноплатный ПК размером 84x55 мм. На плате распаяны ЦПУ AMD Ryzen Embedded R1000 с TDP 18 W и Vega 8, 2-8 ГБ ОЗУ DDR4-3200 и eMMC на 16-64 ГБ. Присутствует порт Gigabit Ethernet, два micro-HDMI и один USB Type C. Поддерживает Linux и Windows IoT, запитывается от 12-вольтового аккумулятора. Продажи начнутся в третьем квартале 2020 года, цена пока не известна.



Пишем первую программу для KolibriOS

Автор: Alex2003, 23 марта 2020

Разработка приложений под KolibriOS обычно ведётся в компиляторе FASM. Я не буду учить ассемблеру, просто опишу сам код, и что он делает, не более.

Важно запомнить несколько постулатов программирования в KolibriOS:

1. Номер функции помещается в регистр eax;
2. Вызов системной функции вызывается с помощью прерывания **int 0x40**;
3. Все регистры, кроме явно указанных в возвращаемом значении, включая регистр флагов **eflags**, сохраняются.

Чтобы вы всё поняли, давайте разберём кусок кода (Листинг 1)

Листинг 1

```
    mov eax, 1    ;Вызов функции 1 (поставить точку в окне); список системных функций есть в DOCPACK - sysfuncr.txt
    mov ebx, 10   ; Координата x=10
    mov ecx, 20   ; Координата  y=10
    mov edx, 0xFFFfff ; Цвет точки
    int 0x40      ; Вызов функции (прерывание)
```

Того же самого можно добиться с помощью вызова макроса из macros.inc (Листинг 2)

Листинг 2

```
      mcall 1, 10, 20, 0xFFFfff
```

На разбор я взял explame.asm из дистрибутива: эта программа определяет код клавиши и выводит соответствующий ей звук. Давайте посмотрим первые несколько строчек программы (Листинг 3):

Листинг 3

```
    use32           ; включить 32-битный режим ассемблера
    org   0         ;  адресация производится с 0

    db    'MENUET01'  ; 8-байтный идентификатор, оставшийся со времён MenuetOS
    dd    1           ; Версия заголовка всегда 1
    dd    START       ;  Адрес первой команды
    dd    I_END       ; Размер программы
    dd    MEM         ; Количество памяти, выделенной под приложение
    dd    STACKTOP    ;  Адрес вершины стэка
    dd    0           ;  Адрес буфера для параметров
    dd    0           ; Зарезервировано

    include "macros.inc" ; подключение макросов – они облегчают жизнь ассемблерщику!

    ;---------------------------------------------------------------------
    ;---  Начало программы  ----------------------------------
    ;---------------------------------------------------------------------

    START:

    red:                    ; перерисовать окно

       call draw_window         ; вызов процедуры отрисовки окна
```

Ну, вот и всё начало! Как видим, всё достаточно просто и понятно.

Теперь перейдём к разбору цикла обработки событий (Листинг 4). Для этого мы используем цикл.

Листинг 4

```
;----------------------------------------------------------------
;---  Цикл обработки событий  ----------------------------
;----------------------------------------------------------------

still:
   mcall 10          ; вызов функции 10 – ожидание события

   cmp  eax,1        ; Перерисовать окно?
   je red            ; Если да – перемещение на метку red:
   cmp  eax,2        ; Нажата клавиша?
   je key            ; Если да – перемещение на key
   cmp  eax,3        ; Нажата кнопка ?
   je button         ; Если да – перемещение на button

   jmp  still        ; Если определено другое событие – перемещение в начало цикла
;----------------------------------------------------------------

  button:
   mcall 17          ; функция 17 – получить идентификатор нажатой кнопки

   cmp   ah, 1       ; Если НЕ нажата кнопка с номером 1,
   jne   still       ; вернуться

 .exit:
   mcall -1          ; Иначе – конец программы (вызов функции закрывания окна)
```

Три события, описанные выше, включены по умолчанию, именно поэтому в данном примере только они. В последующем мы научимся включать и выключать требуемые нам события (например, события мыши).

Теперь, когда мы разобрались с обработкой событий, нам

Листинг 5

```
;----------------------------------------------------------------
;---  Определение и отрисовка окна  -------------------
;----------------------------------------------------------------

draw_window:

   mcall 12, 1       ; функция 12: сообщить ОС о начале отрисовки

   mcall 48, 3, sc,sizeof.system_colors

   ; Далее: сначала длинный и закомментированный вариант без макросов
   ; Затем аналогичный, но уже с макросами

;  mov  eax,0                    ; Функция 0 – определить окно
;  mov  ebx,200*65536+300        ; [x старт] *65536 + [x размер]
;  mov  ecx,200*65536+150        ; [y старт] *65536 + [y размер]
;  mov  edx, [sc.work]           ; цвет фона
```

Листинг 5 (продолжение)

```
;   or   edx, 0x33000000            ; тип окна 3 (есть несколько стилей)
;   mov  edi,header                 ; Заголовок окна
;   int  0x40

; теперь вариант с макросами
    mov   edx, [sc.work]      ; цвет фона
    or          edx, 0x33000000     ; тип окна 3 (есть несколько стилей)
    mcall 0, <200,300>, <200,150>, , ,title

    ; вывод текстовой строки
    mov   ecx, [sc.work_text]         ; цвет фона
    or          ecx, 0x90000000     ; тип строки
    mcall 4, <10, 20>, , message


    mcall 12, 2                 ; функция 12.2, закончили рисовать

    ret                         ; выходим из процедуры
```

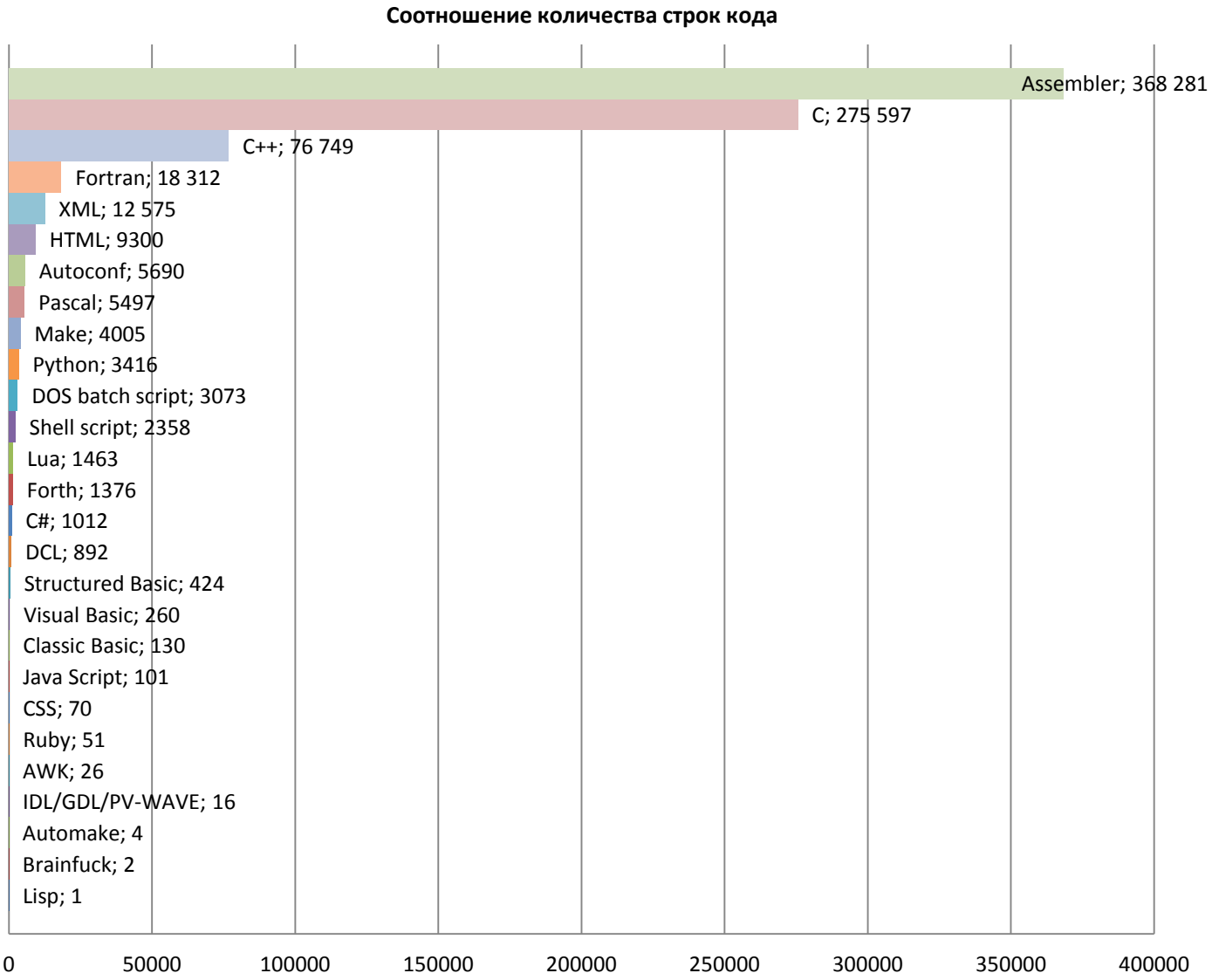
Теперь нам осталось добавить код, который будет озвучивать нажатия клавиш (Листинг 6), и ещё кое-что.

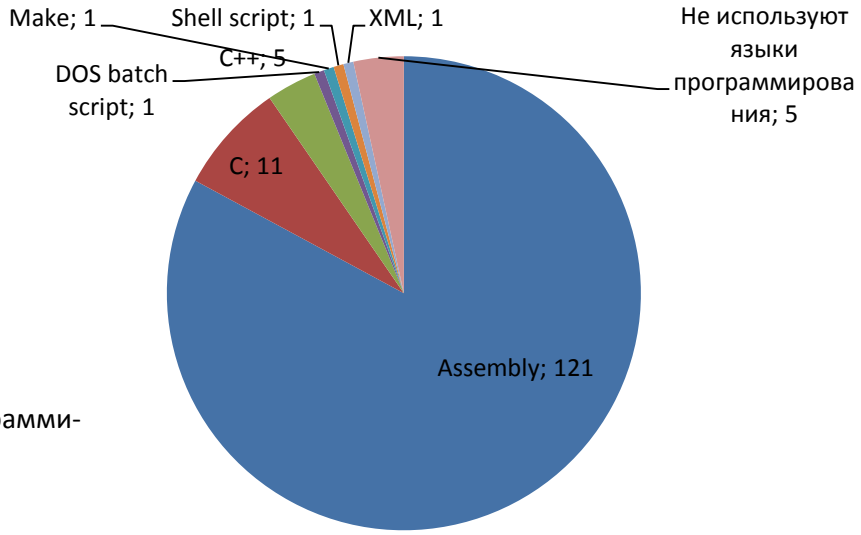
Листинг 6

```
;-----------------------------------------------------------------
;---  Данные программы  ----------------------------------
;-----------------------------------------------------------------

; Вот такая короткая мелодия. Второй байт изменяется нажатием клавиши.

Music:
  db  0x90, 0x30, 0

sc system_colors

message db 'Нажмите любую клавишу...',0  ; Текст внутри окна
title db ‘Пример программы',0                    ; Текст заголовка окна

;-----------------------------------------------------------------

I_END:                          ; конец программы
  rb 4096                 ; память для стэка
align 16
STACKTOP:   ; Метка вершины стэка (при заполнении стэк растёт в сторону уменьшения адресов, потому память для него  в коде программы встречается раньше, чем его вершина)
MEM:            ; Метка, указывающая на конец программы + размер используемой ей ОЗУ
```

Важные указания про конец программы:
Адрес метки **MEM** всегда должен быть больше чем адрес метки **I_END**;
Метка STACKTOP должна располагаться после метки I_END и перед меткой MEM;
Имена меток могут быть с любыми названиями, главное соблюдать правильный порядок их расположения.

Статистика об языках программирования (ЯП)





Наиболее часто используемые языки программирования (из 149 опрошенных)

Данные взяты с OpenHub. 13.02.2020

Послесловие от автора

“Well, mr. Freeman…” – простите, не удержался от того, чтобы не процитировать G-Man’a.

В общем, получите и распишитесь: вы только что прочитали второй выпуск моего журнала! Как вы могли заметить, я внёс некоторые изменения: поменял дизайн, и… всё (а может, и нет). Мне искренне жаль, что я не имею достаточно много свободного времени и таланта статьесложения для того, чтобы целиком и полностью посвятить себя журналу. Впрочем, также виноваты те, кто ежедневно заходят на форум (иногда я наблюдал до 50 (!) человек) и после недолгой прогулки навсегда исчезают, при этом даже не зарегистрировавшись! А ведь если бы в проект влился хотя бы небольшой коллектив (человек пять-десять, ну от силы два), то дела пошли бы в гору.

Но главная беда даже не в этом. Я заметил, что людей в интернете очень сложно организовать: человек захотел – отключился, захотел – послал всех куда подальше (совсем как Mario_r5, flamehowk, diamond …), и тому подобное. Нет, не подумайте о том, что я предлагаю немедленно всем разработчикам переехать в одну страну, область, город, дом, комнату (!) – чтобы деятельность была под чьим-то полным контролем (например, CleverMouse). Просто уже несколько лет наблюдаю, что (как бы это … не звучало) люди, находящиеся в разных частях страны/света делают, и вполне успешно, своё общее дело (Comon Games, к примеру). Значит, команда разработчиков KolibriOS могут организоваться, но ведь что-то мешает… Но что?

…бред

С уважением,

Григорий, человек, который сделал этот номер

KolibriOS Inside, 2020