

ЗАГРУЗКА И СОВМЕСТИМОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Ласый А.М.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Оношко Д.Е. – старший преподаватель

Одним из основных требований к современной операционной системе является возможность запуска и нормальной работы приложения, скомпилированного или для более старой версии данной, или для совершенно другой операционной системы. В данной статье на примере простейшей операционной системы рассмотрены основные моменты, связанные с ее загрузкой и достижением бинарной совместимости.

Проблема бинарной совместимости операционных систем на данный момент является одной из самых актуальных. Каждый день пишется огромное количество новых программ. Многие из этих программ пользователи будут запускать в будущем еще достаточно долго. Однако с течением

времени поддерживать старые программы становится все сложнее, так как операционные системы развиваются достаточно быстро.

В то же время стоит проблема загрузки операционных систем с таких устройств, как флеш-накопитель, в CSM BIOS или в Legacy BIOS, поскольку предоставляемая по этому поводу информация в большинстве своем разрозненна и не согласованна.

Для изучения данных вопросов была разработана с нуля небольшая операционная система для реального режима x86-совместимых процессоров.

Разработка велась на языке ассемблера, что дало возможность максимально контролировать всю происходящую работу с данными. Размер ядра операционной системы составил порядка одного килобайта.

В качестве пользовательского интерфейса с помощью прерывания BIOS 10h была реализована графическая оболочка, позволяющая отображать имеющиеся на диске файлы и запускать их.

Так как самой известной операционной системой, которая работает в реальном режиме, является MS-DOS, целью разработки новой операционной системы с точки зрения бинарной совместимости являлась поддержка COM-файлов. COM-файлы представляют собой один из простых типов исполняемых файлов. Они широко использовались в 70-80-е годы 20 века в операционных системах ввиду своей простоты и удобства, так как код, стек и данные располагались в одном сегменте. [1]

Для достижения поддержки данного типа файлов необходимо было обеспечить корректные запуск и завершение программы. Первое достигается за счёт установки выполнения выбранной программы со смещения 100h в сегменте. Второе же требует реализации прерывания DOS 20h, а также минимального заполнения префикса программного сегмента. В частности, особую важность представляют два байта префикса, поскольку именно там в COM-файлах располагается инструкция INT 20H. Доступ к этой инструкции можно получить посредством предварительной записи операционной системой в стек числа 0 с его последующим считыванием инструкцией ret. [1]

В результате поддержка COM-файлов была успешно осуществлена. Тестирование проводилось как на собственных программах, так и на старых видеоиграх, которые были скомпилированы в 1980-х годах. В качестве примера можно привести видеоигры Pac-Man и Paratrooper. Все тестируемые программы запускались и работали без ошибок.

В качестве файловой системы для данной операционной системы была выбрана FAT12, так как она применяется на дискетах и схожа с файловой системой FAT32, которая до сих пор используется на флеш-накопителях, а также проста и совместима со многими операционными системами. [2]

Для реализации считывания файлов из области данных FAT12 было реализовано два обработчика программных прерывания. Первое осуществляет считывание с накопителя в определенный буфер заданное количество секторов. Второе прерывание, используя первое загружает в буфер файл с указанным именем.

Операционная система была записана на дискету и на флеш-накопитель. Целью опыта являлось успешно запустить ее на разных устройствах. Для записи на накопителе был сформирован образ загрузочной дискеты размером 1.44 Мб. Запись образа на дискету с последующим запуском операционной системы с нее трудностей не вызвал, тогда как запись на флеш-накопитель была проблематична.

На данный момент существует три варианта загрузки операционной системы с флеш-накопителя:

- Как единственный FAT-12 раздел.
- Как Hard Disk Drive с несколькими разделами.
- Как оптический диск.

В данном эксперименте был выбран второй вариант, поскольку он максимально аналогичен запуску с реального HDD. Помимо этого на языке ассемблера проще генерировать FAT-образ, чем образ оптического диска. Однако и этот вариант имеет ряд своих затруднений. Основной проблемой являлось то, что утилиты, осуществляющие разделение диска на тома, при работе с одним и тем же диском получают совершенно разные результаты.

Как следствие, правильно записать значение BPB_HiddSec в загрузочном секторе становится сложно. При этом в случае, если значение окажется неверным, загрузочный сектор будет считывать не с того места, которое нас интересует.

Таким образом, необходимо вручную определять смещение раздела и на основе полученного результата высчитывать количество секторов, на которое был смещен загрузочный сектор. В данном эксперименте смещение составило 1 Мб. Размер сектора насчитывает 512 байт. Имея все необходимые данные, смещение в пересчете на количество секторов можно рассчитать по следующей формуле:

$$s = \frac{d}{512}$$

(1),

где K – смещение загрузочного сектора в пересчете на секторы; O – смещение загрузочного сектора; S – размер сектора в байтах.

Получившаяся в данном эксперименте с помощью утилиты diskpart структура разделов флеш-накопителя представлена на рисунке 1:

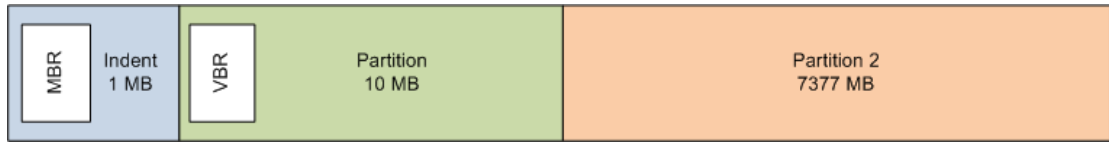


Рисунок 1 – Структура разделов флеш-накопителя

В самом начале диска размещается MBR (master boot record), задача кода которого заключается в нахождении и переходе в нужный раздел диска. А в начале загрузочного раздела с операционной системой расположен VBR (volume boot record), который содержит код по поиску, а также дальнейшей загрузке операционной системы (в данном эксперименте происходила загрузка уже непосредственно ядра, однако во многих случаях загружается так называемый вторичный загрузчик, который и выполняет загрузку ядра). Кроме того, стоит отметить, что не упомянутый ранее раздел 2, представленный на рисунке 1, можно отформатировать и использовать как обычный флеш-накопитель.

По итогу после создания структуры разделов и подсчета смещения O величина K согласно формуле (1) составила 2048 секторов.

Помимо этого, запись образа на выбранный накопитель может быть затруднена из-за немногочисленности программного обеспечения, предоставляющего удобный интерфейс для записи отдельных секторов и корректно работающего в современных версиях операционной системы Windows.

Также стоит отметить, что после осуществления записи необходимо вручную сделать раздел, на который записана операционная система, активным.[3] В противном случае загрузка с данного раздела происходить не будет.

Для тестирования операционной системы было отобрано несколько моделей персональных компьютеров:

- Dell Inspiron 15 7000.
- ASUS Celeron - M540.
- ASUS K53T.
- HP 250 G5

Отбор проводился с целью охвата как можно большего диапазона технических характеристик персональных компьютеров. На всех моделях операционная система загружалась корректно и функционировала в полной мере.

Список использованных источников:

1. Электронный справочник TECH Help! 6 by Flambeaux Software
2. Microsoft Extensible Firmware Initiative FAT32 File System Specification
3. DiskPart Documentation on docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart