FreeLDR bootsector description

- Данный бутсектор предназначен для использования в загрузчике FreeLdr системы osFree (OS/3) для загрузки Micro File System Driver (MicroFSD, aka blackbox). BlackBox – это обычный DOS executable в COM-формате, то есть, его размер теоретически ограничен 64 Кбайтами. Поэтому данный загрузчик также рассчитан на загрузку файлов размером до 64 Кбайт.
- Замечание.
  - Данный загрузчик просто грузит произвольный файл размером до 64 Кб как последовательность секторов и передает управление этому файлу.
  - Файл грузится по определенному сегментному адресу и управление передается по заданному адресу точки входа, находящейся в этом сегменте.
  - Как сегмент, так и точка входа настраиваются – они могут быть любыми.
  - Загружаемый файл может быть любым файлом такого размера, но его первичное предназначение – содержать в себе MicroFSD.
- Идеология следующая:
  - Данный загрузчик не понимает никакой файловой системы.
  - Его предназначение – загрузить Blackbox для конкретной ФС.
  - Смысл метафоры BlackBox'а в том, что есть один единственный исполняемый файл, который понимает формат конкретной ФС.
  - Остальные части загрузчика независимы от ФС, они используют Blackbox как черный ящик, который умеет читать файлы сданной ФС, вся специфика которой спрятана внутри этого черного ящика (англ. black box – черный ящик).
  - BlackBox для этого предоставляет 4 функции:
    1. Открыть файл
    2. Закрыть файл
    3. Прочитать данные из файла
    4. Завершить работу с BlackBox.
  - Данный загрузчик имеет две версии. Обе из них компилируются из одного и того же исходного файла. Для получения первой версии необходимо задать в мейкфайле бутсектора препроцессорный символ OLDCODE (т. е., надо раскомментировать в мейкфайле -dOLDCODE).
  - Первая версия отличается тем, что она
    - совместима с IBM'овским бутсектором для FAT и может грузить IBM FAT MicroFSD
    - умеет грузить только нефрагментированные файлы. (IBM'овский FAT-бутсектор тоже не умеет грузить фрагментированные файлы, но размер IBM FAT MicroFSD – всего 2-3 сектора, но для более сложных ФС это не подходит, т.к. размер MicroFSD значительно больше, и вероятность фрагментации возрастает. (Для примера: размер ext2fs MicroFSD – 30 секторов)). Данная версия подходит, когда загружаемый файл гарантированно не является фрагментированным (или с высокой долей вероятности является таким), например, когда это не файл как таковой, а остаток бутблока ФС после бутсектора, если такая область для данной ФС предусмотрена (например, JFS или HPFS).
  - Вторая версия
    - несовместима с IBM FAT MicroFSD,
    - но может грузить фрагментированный файл MicroFSD.
    - Она более подходит в том случае, когда специальной области под бутблок не предусмотрено, например, FAT12, FAT16 или “чужие” файловые системы,

такие как ext2fs (в ext2fs перед суперблоком для загрузчика предусмотрено лишь два сектора, что очень мало для MicroFSD).

Поля бутсектора (от конца к началу)

1. Сигнатура бутсектора: word, находится по смещению 0x1FE от начала бутсектора и равна 0x55AA.
2. ForceLBA: byte, смещение 0x1FD, если данный байт равен 1, то поддержка доступа через LBA не проверяется, а включается принудительно.

Параметры первой версии (-dOLDCODE)

1. muFSDLen, byte (смещение 0x1FC). Задает длину загружаемого файла в секторах.
2. muFSDAddr, dword (смещение 0x1F8). Задает номер стартового сектора загружаемого файла. Этот номер отсчитывается от начала раздела. (бутсектор имеет номер 0).
3. muFSDEntry, word (смещение 0x1F6). Задает смещение точки входа в загружаемый исполняемый файл от начала сегмента.
4. muFSDLoadSeg, word (смещение 0x1F4). Задает сегментный адрес, по которому производится загрузка.

Параметры второй версии

- В отличие от первой версии, вторая версия не содержит адрес и длину загружаемого файла непосредственно в бутсекторе. Взамен этого, бутсектор содержит только адрес дополнительного сектора с картой размещения загружаемого файла (allocation map sector).
- Более развернуто.
    - Вторая версия бутсектора содержит поддержку загрузки фрагментированного файла. Длина этого файла может быть как максимум, 64 Кбайта. 64 Kb / 512 = 128 секторов. То есть, этот файл может занимать max. 128 секторов. Максимальный вариант фрагментированного файла – это 128 секторов, разбросанные по всему диску. Для хранения адреса каждого сектора потребуется 4 байта. 4 * 128 == 512, то есть, для хранения карты размещения фрагментированного файла длиной 64 Кб потребуется в точности один сектор. Этот сектор представляет собой массив из 128 4-х-байтных указателей на секторы. Каждое 4-байтное двойное слово указывает в точности на один сектор.
    - Сектор с картой размещения загружаемого файла может располагаться где угодно на диске. Он может находиться как непосредственно после бутсектора (как это сделано у нас при загрузке с раздела с файловой системой ext2fs: в ext2 под загрузчик зарезервировано как раз 2 сектора, первый из них – бутсектор, второй – карта распределения.), так и в любом месте области файлов ФС (как это имеет место при загрузке с FAT-раздела).
        1. mapAddr: dword (смещение 0x1F9), задает номер сектора с картой размещения загружаемого файла. Номер отсчитывается от начала раздела (бутсектор имеет номер 0).
        2. muFSDEntry: word (смещение 0x1F7), задает точку входа в загружаемый файл.
        3. mapLoadSeg: word (смещение 0x1F5), задает адрес загрузки allocation map. Загружаемый файл грузится сразу после allocation map (т. е., по сегментному

адресу muFSDLoadSeg == mapLoadSeg + 0x20).

- Загружаемый файл загружается по кускам в один сектор согласно allocation map и грузится непосредственно после allocation map. То есть, если мы хотим, например, загрузить файл по сегментному адресу 0x5660, то нужно задать mapLoadSeg == muFSDLoadSeg - 0x20 == 0x5640.

Замечания

1. Данный бутсектор использует LBA для загрузки секторов с диска, если расширения int 13h поддерживаются BIOS'ом. В противном случае он использует функцию для чтения секторов через CHS.
2. При загрузке с флоппи-диска данный бутсектор использует CHS, кроме того, производится инициализация флоппи-контроллера для работы с дискетами с нестандартным количеством секторов в треке.

© osFree project, 2006, Nov 12.

From:
https://cocorico.osfree.org/doku/ - **osFree wiki**

Permanent link:
https://cocorico.osfree.org/doku/doku.php?id=en:docs:boot:bootsector

Last update: **2014/06/28 07:15**