

Формат Intel-HEX

Введение

Шестнадцатиричный объектный формат файлов Intel-HEX (далее просто HEX-формат) – это способ представить двоичные данные в виде кодов ASCII. Поскольку файл состоит из символов ASCII, а не двоичных кодов, появляется возможность хранить данные на бумаге, перфоленте или перфокартах, выводить их на терминал, принтер и т.д. Восьмибитовый HEX-формат файлов предусматривает размещение данных и кода в 16-разрядном линейном адресном пространстве для 8-разрядных процессоров Intel. 16-разрядный HEX-формат файлов дополнительно позволяет использовать 20-разрядное сегментное пространство адресов 16-разрядных процессоров Intel. И, наконец, 32-разрядный формат позволяет оперировать линейным 32-разрядным адресным пространством 32-разрядных процессоров.

Шестнадцатиричное представление двоичных данных в виде ASCII требует использование двух символов для записи одного байта, при этом первый символ всегда соответствует старшей тетраде битов одного байта. Такой подход увеличивает количество символов в двое по сравнению с количеством двоичных данных.

Формат файла организован в виде набора записей, содержащих сведения о типе, количестве данных, адресе их загрузки в память и дополнительные сведения. В настоящее время определены шесть различных типов записей, однако не все их комбинации определены для разных форматов данных.

Записи могут быть следующих типов:

- Данные (определена для всех форматов данных)
- Маркер конца файла (определена для всех форматов файла)
- Сегментный адрес (определена для 16- и 32-битных форматов)
- Сегментный адрес старта (определена для 16- и 32-битных форматов)
- Линейный адрес (определена только для 32-битного формата)
- Линейный адрес старта (определена только для 32-битного формата)

Общий формат записей

Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPE REC	Данные DATA	Контрольная сумма CHECKSUM
:	1 байт	2 байта	1 байт	RECLEN байт	1 байт

Каждая запись представляет собой ASCII-строку файла. В одной строке – одна запись.

Каждая запись начинается с МАРКЕРА ЗАПИСИ, который обозначается ASCII-символом двоеточие (":").

Каждая запись содержит поле RECLEN, определяющее количество байтов данных или информационных байтов, назначение которых определяется типом записи. Максимальное значение этого поля – 255 (0xFF).

Каждая запись содержит поле OFFSET, определяющее 16-битное смещение в адресном пространстве байтов данных. Это поле используется только в записях данных, а в остальных случаях оно должно быть равно нулю.

Каждая запись содержит поле TYPE REC, определяющее тип текущей записи (из ранее упомянутых шести). Это поле используется для интерпретации всех остальных полей записи. Типы записей кодируются следующими значениями поля TYPE REC (в ASCII):

- "00" – данные
- "01" – маркер конца файла
- "02" – адрес сегмента
- "03" – сегментный адрес старта
- "04" – линейный адрес
- "05" – линейный адрес старта

Каждая запись содержит поле DATA переменной длины, которое содержит ноль или более байтов, закодированных символами ASCII. Назначение этих байтов определяется типом записи.

Наконец, каждая запись завершается полем CHECKSUM, гарантирующим целостность всех данных записи. Значение этого поля равно дополнению по модулю 256 до нуля суммы по модулю 256 всех байтов, начиная с поля RECLEN и заканчивая последним байтом поля DATA. При считывании записи следует суммировать по модулю 256 все байты записи, включая поле CHECKSUM. Если в конце концов сумма равна нулю, это означает, что данные считаны без искажений, в противном случае данные недостоверны.

Запись "Линейный адрес"

Формат записи следующий:



Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPEREC	Данные ULBA	Контрольная сумма CHECKSUM
:	02	0000	04	2 байта	1 байт

Эта запись служит для задания значения битов 16-31 в линейном базовом адресе (LBA, Linear Base Address), причем биты 0-15 LBA равны нулю. Биты 16-31 LBA определяются верхним линейным базовым адресом (ULBA, Upper Linear Base Address). Абсолютное значение адреса байта данных в памяти определяется как сумма значения LBA и значения поля OFFSET в последующих записях данных, плюс индекс байта данных внутри поля DATA. Эта сумма выполняется без учёта переполнения результата (то есть не может превышать 0xFFFFFFFF, 4 Гб).

Фактический линейный адрес байта данных вычисляется в итоге по формуле:

$$\text{ByteAddr} = (\text{LBA} + \text{DRLO} + \text{DRI}) \bmod 4\text{G},$$

где: DRLO - значение поля OFFSET записи данных;
DRI - индекс байта в поле DATA записи данных;
mod 4G - операция "сложение по модулю 2^{32} ".

Когда запись "Линейный адрес" встречается в файле, вычисляется значение LBA, которое действует для всех последующих записей данных, пока не встретится снова запись "Линейный адрес". По умолчанию LBA = 0.

Запись "Адрес сегмента"

Формат записи следующий:

Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPEREC	Данные USBA	Контрольная сумма CHECKSUM
:	02	0000	04	2 байта	1 байт

Эта запись служит для задания значения битов 4-19 сегментного базового адреса (SBA, Segment Base Address), где биты 0-3 SBA равны нулю. Биты 4-19 SBA определяются верхним базовым адресом сегмента (USBA, Upper Segment Base Address). Абсолютный адрес байта в записи данных вычисляется путем прибавления к SBA значения поля OFFSET записи данных и индекса байта относительно начала поля DATA. Прибавление смещения (OFFSET) осуществляется по модулю 65536 (64 К), без учёта переполнения.

Таким образом, адрес конкретного байта вычисляется по формуле:

$$\text{ByteAddr} = \text{SBA} + (\text{DRLO} + \text{DRI}) \bmod 64\text{K},$$

где: DRLO - значение поля OFFSET записи данных;
DRI - индекс байта в поле DATA записи данных;
mod 64K - операция "сложение по модулю 65536".

Когда запись "Адрес сегмента" встречается в файле, вычисляется значение SBA, которое действует для всех последующих записей данных, пока не встретится снова запись "Адрес сегмента". По умолчанию SBA = 0.

Запись данных

Формат записи следующий:

Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPEREC	Данные DATA	Контрольная сумма CHECKSUM
:	1 байт	2 байта	00	RECLEN байтов	1 байт

Эта запись собственно и содержит данные. Метод вычисления фактического (абсолютного) адреса каждого байта данных в памяти определяется по вышеприведённым формулам и зависит от формата данных.

Линейный адрес старта

Формат записи следующий:

Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPEREC	Данные EIP	Контрольная сумма CHECKSUM
:	04	0000	05	4 байта	1 байт

Запись "Линейный адрес старта" используется для указания адреса, с которого начинается исполнение объектного файла. Это значение заносится в регистр EIP процессора. Следует обратить внимание, что эта запись определяет только точку входа сегмента кода для защищённого режима процессоров 80386. В обычном режиме точка старта определяется записью "Сегментный адрес старта", которая определяет значения пары регистров

CS:IP.

Запись "Линейный адрес старта" может находиться в любом месте файла. Если её нет, загрузчик использует адрес старта по умолчанию.

Значение регистра EIP процессора содержится в соответствующем поле записи, для него требуется всегда 4 байта.

Сегментный адрес старта

Формат записи следующий:

Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPE REC	Данные CS:IP	Контрольная сумма CHECKSUM
:	04	0000	03	4 байта	1 байт

Запись "Сегментный адрес старта" используется для указания адреса, с которого начинается исполнение объектного файла. Это значение определяет 20-битный адрес, заносимый в регистры CS:IP процессора. Следует обратить внимание, что эта запись определяет только точку входа в 20-битном адресном пространстве процессоров 8086/80186.

Запись "Сегментный адрес старта" может находиться в любом месте файла. Если её нет, загрузчик использует значение по умолчанию.

Значение регистров CS:IP процессора содержится в соответствующем поле записи, для него требуется всегда 4 байта. Значение хранится в порядке "от старшего к младшему", то есть младший байт значения регистра IP хранится в четвертом байте поля CS:IP, старший - в третьем, затем во втором хранится младший байт значения регистра CS, и в первом - старший байт регистра CS.

Маркер конца файла (терминатор)

Формат записи следующий:

Маркер записи	Кол-во данных RECLEN	Смещение OFFSET	Тип записи TYPE REC	Контрольная сумма CHECKSUM
:	00	0000	01	1 байт

Эта запись не содержит полей с изменяющимися данными, поэтому выглядит всегда совершенно одинаково: ":00000001FF". Запись обозначает конец данных в файле. Все последующие строки, если они есть в файле, игнорируются.

Пример содержимого файла формата Intel-HEX

```
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

- Маркер записи
- Кол-во данных
- Смещение
- Тип записи
- Данные
- Контрольная сумма