WYOS - Выпуск №11
ELF - продолжение
commrade, Среда, 30 Июнь 2004, 19:01

Я, конечно, извиняюся...

За то, что так нагло влезаю в самое начало прекрасного выпуска от commrade, но все же... Это я, Роман. Который к тому же "I Khimov".

Как вы, наверное, уже заметили, выпусков не было фактически две недели. Это было связано с неразберихой среди выпускающих редакторов (кто, когда, о чем) и с неразберихой в высшем командном составе 3OS (кто, когда, зачем). Плюс ко всему, как всегда, загруженность другими делами... Но, мы о вас помним и мы вас ценим и любим, поэтому совесть не позволила нам не выпустить готовый выпуск в эту пятницу. Поэтому он пришел к вам в почтовые ящики сегодня, надеюсь это вас обрадовало.

Я постараюсь сделать на этой неделе очередной выпуск, и, все, больше не могу позволять себе занимать это место, в самом начале выпу...

Напиши свою ОС! #11

Здравствуйте, уважаемые подписчики!

С вами я, ваш commrade и сегодня мы закончим рассматривать под нашим микроскопом формат исполняемых файлов ELF.

Программные секции (продолжение)

Вот возможные значения поля type:

<table border='1' bgcolor='#FFFFB3'>
<tr><td>Имя</td><td></td><td></td></tr>
<tr><td>SHT_NULL </td><td>0 </td><td>Этой величиной маркируют неактивную секцию заголовка.</td></tr>
<tr><td>SHT_PROGBITS</td><td>1</td><td>Эта величина хранит информацию о программе, чей формат и используется для исполнения </td></tr>
<tr><td>SHT_SYMTAB</td><td>2</td><td>Секции SHT_SYMTAB и SHT_DYNSYM хранят таблицу символов. Секция SHT_SYMTAB обеспечивает символы для редактирования связей.</td></tr>
<tr><td>SHT_STRTAB</td><td>3</td><td>Секция содержит строки таблицы символов.</td></tr>
<tr><td>SHT_RELA</td><td>3</td><td>Секция содержит данные для перемещения с явными слагаемыми, как например, тип Elf32_Rela для 32- битового класса объектных файлов.</td></tr>
<tr><td>SHT_HASH</td><td>5</td><td>Секция содержит символьную таблицу хэша. Все объекты, участвующие в динамической связи должны содержать символьную таблицу

</td></tr>
<tr><td>SHT_DYNAMIC</td><td>6</td><td>Секция хранит информацию для динамической свя</td></tr>
<tr><td>SHT_NOTE</td><td>7</td><td>Секция хранит информацию, которая некоторым образом маркирует файл.</td></tr>
<tr><td>SHT_NOBITS</td><td>8</td><td>Секция этого типа не занимает никакого пространства в файле, но в противном случае имеет сходство с SHT_PROGBITS. Хотя эта секция и не содержит никаких байтов, sh_offset элемент содержит концептуальное файловое смещение.</td></tr>
<tr><td>SHT_REL</td><td>9</td><td>Секция содержит данные для перемещения без явных слагаемых, как например, тип Elf32_Rel для 32- битового класса объектных файлов.</td></tr>
<tr><td>SHT_SHLIB</td><td>10</td><td>Этого типа секции зарезервирована но имеет неопределенную семантику.</td></tr>
<tr><td>SHT_DYNSYM</td><td>11</td><td>Эта секция содержит минимальный набор установок для динимической связи </td></tr>
<tr><td>SHT_LOPROC</td><td>0x70000000</td><td>Величины в этом включающем дипазоне зарезервированыдля процессор-специфической семантики.</td></tr>
<tr><td>SHT_HIPROC</td><td>0x7fffffff</td><td>Величины в этом включающем дипазоне зарезервированыдля процессор-специфической семантики.</td></tr>
<tr><td>SHT_LOUSER</td><td>0x80000000</td><td>Эта величина определяет более низкий связанный дипазона индексов зарезервированнх для прикладных программ.</td></tr>
<tr><td>SHT_HIUSER</td><td>0xffffffff</td><td>Эта величина определяет верхний связанный дипазона индексов зарезервированнх для прикладных программ.</td></tr>
</table>

Итак мы рассмотрели возможные значения параметра sh_type, теперь рассмотрим значения sh_flags:

<table border='1' bgcolor='#FFFFB3'>
<tr><td>Имя</td><td></td><td></td></tr>
<tr><td>SHF_WRITE</td><td>0x1</td><td>Секция содержит данные, которые могут быть перезаписанны процессом в течении </td></tr>
<tr><td>SHF_ALLOC</td><td>0x2</td><td>Секция занимат памятьв течении врмени выполнения процесса.</td></tr>
<tr><td>SHF_EXECINSTR</td><td>0x4</td><td>Секция содержит выполнямые машинные инструкции.</td></tr>
<tr><td>SHF_MASKPROC</td><td>0xf0000000</td><td>Все биты включенных в эту маску зарезервированыдля процессор-специфической семантики.</td></tr>
</table>

Два участника в заголовке секции, sh_link и sh_info, хранят специальную информацию, в зависимости от типа секции.

<table align='Center' border='1' bgcolor='#FFFFB3'>
<tr>
<td>sh_type</td><td>sh_link</td><td>sh_info</td>
</tr>
<tr>

<td>SHT_DINAMIC</td><td>Индекс заголовка секции стола строки использовался данными в
</td><td>0</td>
</tr>
<tr>
<td>SHT_HASH</td><td>Индекс заголовка секции символьного стола на который стол мусора относится</td><td>0</td>
</tr>
<tr>
<td>SHT_REL</td><td>Индекс заголовка секции связанного символьного
стола.</td><td>Индекс заголовка секции секции на который the
перемещение относится. </td>
</tr>
<tr>
<td>SHT_RELA</td><td>Индекс заголовка секции связанного символьного стола.

</td><td>Индекс заголовка секции секции на который the
перемещение относится. </td>
</tr>
<tr>
<td>SHT_SYMTAB</td><td>индекс заголовка секции связанного стола строки.</td><td>Один больше, чем символьный табличный индекс последнего локального символа (связь
STB_LOCAL)</td>
</tr>
<tr>
<td>SHT_DYNSYM</td><td>индекс заголовка секции связанного стола
строки.</td><td>Один больше, чем символьный табличный индекс последнего локального символа (связь
STB_LOCAL)</td>
</tr>
<tr>
<td>Other</td><td>SHN_UNDEF</td><td>0</td>
</tr>
</tbody>
</table>

Специальные секции

На этом мы закончим рассматривать программные секции и перейдем к специальным секциям. Специальные секции хранят программную и управляющую информацию. Секции в списке ниже используются системой и иметь указанные типы и атрибуты.

<table border='1' bgcolor='#FFFFB3'>
<tr><td>Имя</td><td></td><td></td><td></td></tr>
<tr><td>.bss</td><td>SHT_NOBITS</td><td>SHF_ALLOC + SHF_WRITE</td><td>Эта секция содержит неинициализировнные данные.</td></tr>
<tr><td>.comment</td><td>SHT_PROGBITS</td><td>none</td><td>Эта секция хранит информацию о версии.</td></tr>
<tr><td>.data</td><td>SHT_PROGBITS</td><td>SHF_ALLOC + SHF_WRITE</td><td>Эта

секция содержит инициализированые данные.</td></tr>
<tr><td>.data1</td><td>SHT_PROGBITS</td><td>SHF_ALLOC + SHF_WRITE</td><td>Эта секция содержит инициализированые данные.</td></tr>
<tr><td>.debug</td><td>SHT_PROGBITS</td><td>none</td><td>Эта секция хранит информацию для символической отладки.</td></tr>
<tr><td>.dynamic</td><td>SHT_DYNAMIC</td><td>see below</td><td>Эта секция хранит информацию о динамической связи. </td></tr>
<tr><td>.dynstr</td><td>SHT_STRTAB</td><td>SHF_ALLOC</td><td>Эта секция содержит строки необходимые для динамической свя</td></tr>
<tr><td>.dynsym</td><td>SHT_DYNSYM</td><td>SHF_ALLOC</td><td>Эта секция содержит динамическую связующую символьную таблицу.</td></tr>
<tr><td>.fini</td><td>SHT_PROGBITS</td><td>SHF_ALLOC + SHF_EXECINSTR</td><td>Эта секция содержит выполняемые инструкции, которые содействуют коду завершения процесса. То есть, когда программа выходит из нормального состояния, система выполняет код в этой секции.</td></tr>
<tr><td>.got</td><td>SHT_PROGBITS</td><td>see below</td><td>Эта секция содержит глобальную таблицу смещения</td></tr>
<tr><td>.hash</td><td>SHT_HASH</td><td>SHF_ALLOC</td><td>Эта секция содержит символьную таблицу хэша.</td></tr>
<tr><td>.init</td><td>SHT_PROGBITS</td><td>SHF_ALLOC + SHF_EXECINSTR</td><td>Эта секция содержит выполняемые инструкции, которые содействуют инициализации процесса.</td></tr>
<tr><td>.interp</td><td>SHT_PROGBITS</td><td>see below</td><td>Эта секция содержит имя пути программного интерпритатора.</td></tr>
<tr><td>.line</td><td>SHT_PROGBITS</td><td>none</td><td>Эта секция хранит информацию о номере строки для символической отладки, которая описывает корреспонденциимежду исходной программой и машинным кодом.</td></tr>
<tr><td>.note</td><td>SHT_NOTE</td><td>none</td><td>Эта секция хранит информацию в </td></tr>
<tr><td>.plt</td><td>SHT_PROGBITS</td><td>see below</td><td>Эта секция содержит процедуры таблицы связи.</td></tr>
<tr><td>.relname</td><td>SHT_REL</td><td>see below</td><td>Эта секция хранит информацию о перемещении загружаемых </td></tr>
<tr><td>.relaname</td><td>SHT_RELA</td><td>see below</td><td>Эта секция хранит информацию о перемещении загружаемых </td></tr>
<tr><td>.rodata</td><td>SHT_PROGBITS</td><td>SHF_ALLOC</td><td>Эта секции содержат данные только для чтения</td></tr>
<tr><td>.rodata1</td><td>SHT_PROGBITS</td><td>SHF_ALLOC</td><td>Эта секции содержат данные только для чтения</td></tr>
<tr><td>.shstrtab</td><td>SHT_STRTAB</td><td>none</td><td>Эта секция содержит имена </td></tr>
<tr><td>.strtab</td><td>SHT_STRTAB</td><td>see below</td><td>Эта секция содержит строки, наиболее общие строки, для связи с символьными табличными данными.</td></tr>
<tr><td>.symtab</td><td>SHT_SYMTAB</td><td>see below</td><td>Эта секция содержит символьную таблицу.</td></tr>
<tr><td>.text</td><td>SHT_PROGBITS</td><td>SHF_ALLOC + SHF_EXECINSTR</td><td>Эта секция содержит код или выполняемые инструкции программы.</td></tr>
</table>

Строковая таблица

Секции строковой таблицы содержат завершенные символьные последовательноти. Объектный файл использует эти строки, чтобы представлять символы и имена секций. Одна строка это как индекс секции в таблице строк. Первый байт, который - нулевой индекс, определен, чтобы содержать нулевой символ. Подобно строковым таблицам последний байт строки содержит нулевой символ, указывающий на завершение строк. Строка, чей индекс является нулем определяется как пустая, в зависимости от контекста. Табличная секция пустой строки разрешена: секция sh_size члена заголовка должна содержать нуль. Не равным нулю индексы недействительныдля пустого стола строки. Секция sh_name члена заголовков содержит индекс в строку заголовка табличной секции секции, как определено e_shstrndx членом заголовка ELF. Следующая таблица показывает строки таблицы с 25 байтами и строки связанными различными индексами.

<table border='1' bgcolor='#FFFFB3'>
<tr><td> </td><td> +0</td><td> +1</td><td> +2</td><td> +3</td><td> +4</td><td>
+5</td><td> +6</td><td> +7</td><td> +8</td><td> +9</td></tr>
<tr><td> 0</td><td> □</td><td> n</td><td> a</td><td> m</td><td> e</td><td>
.</td><td> □</td><td> V</td><td> a</td><td> r</td></tr>
<tr><td> 10</td><td> i</td><td> a</td><td> b</td><td> l</td><td> e</td><td>
</td><td> a</td><td> b</td><td> l</td><td> e</td></tr>
<tr><td> 20</td><td> □</td><td> □</td><td> x</td><td> x</td><td> □</td><td>
</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr>
</table>
или по
<table border='1' bgcolor='#FFFFB3'>
<tr><td> </td><td> </td></tr>
<tr><td> 0</td><td> none</td></tr>
<tr><td> 1</td><td> name.</td></tr>
<tr><td> 7</td><td> Variable</td></tr>
<tr><td> 11</td><td> able</td></tr>
<tr><td> 16</td><td> able</td></tr>
<tr><td> 24</td><td> null string</td></tr>
</table>

Как показно в примере, табличный индекс строки может ссылаться на любой байт в секции. Строка может появиться неоднократно: ссылки на подстроки могут существовать. Нессылочные строки также допускаются.

Таблица символов

Таблица символов объектного файла содержит информацию которую нужно распологать и перемещать программам. Символьный табличный индекс является припиской в этот массив. Индекс 0 нужен для указания на первую запись. Содержание первой записи следующее:
STN_UNDEF=0
Таблица символов имеет следующий формат:

typedef struct

```
{
	Elf32_Word	st_name; /* Этот член содержит индекс на
объектные файлы символьный стол строки, который
содержит символьных имен. */
	Elf32_Addr	st_value;/* Этот член дает величину связанного символа. */
	Elf32_Word	st_size; /* Много размер связанных символов. */
	unsigned char	st_info; /* Этот элемент определяет атрибуты
типа и связи. */
	unsigned char	st_other;/* Этот элемент к настоящему времени
содержит 0 и не имеет определенного значения. */
	Elf32_Half	st_shndx;/* Каждая символная табличная запись -
определенно оносительно нескольких секции: этот элемент
содержит важный заголовок табличного индекса секции. */
}Elf32_Sym;
```

Значения символов

Символьные табличные данные для других объектных файловых типов по разному интерпретируются для члена st_value.

В перемещаемых файлах, st_value содержит ограничения выравнивания для символа чей индекс секции - SHN_COMMON.
В переместимых файлах, st_value содержит компенсацию секции для определенного символа.
В программах и общих объектных файлах, st_value содержит виртуальный адрес.

Хотя символьные табличные величины имеют аналогичные значения для других объектных файлов,
данные допускают эффективный доступ подходящими программами.

Заключение

Уфффффффффффффффффффф!!!!!!!!!! Да это тебе не крозябли бозябли. Ну что хочется сказать в заключение. Тема "Формат исполняемых обектов ELF" очень большая и поэтому очень много осталось за кадром. На самом деле чтобы существенное написать по этому поводу нужно иметь представление о том как работает тот или иной формат данных, к сожалению этим представлением я не обладаю (так как никогда с ним дел не имел). Скорее всего в выпуск закрались ошибки или неточности посему если хотите меня поправить, смело пишите мне на мыло (Или оставляйте комментарий - Roman I Khimov) . И еще выпуск писался с одного иностранного источника поэтому некоторые определения выглядят так если бы китаец попытался спеть песенку "В лесу родилась ". И такие ошибки (корявый перевод и корявое понимание материала) следует поправлять. Смело пишите на "фронт программного обеспечения" бойцу-инженеру commrade с пометкой "Тебе ", с удовольствием выстрелю в спину и отстрелю какую-нибудь очень важную часть тела.

Список используемой литературы:

1. Portable Formats Specification, Version 1.1: Executable and Linkable Format (ELF)

2. http://www.nondot.org/~sabre/os/