РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С НЕТИПИЗИРОВАННЫМИ ФАЙЛАМИ/АРХИВАМИ В СРЕДЕ ПРОГРАММИРОВАНИЯ DELPHI

Николюкин М.С.
Технический колледж ГОУ ВПО «Тамбовский государственный технический университет»
Тамбов, Россия

В последние годы, различные IT-фирмы и крупные компании, стараются с каждым разом всё сильнее и сильнее защитить свой программный продукт от несанкционированного доступа. В тоже время, различные независимые разработчики, иными словами – хакеры, пытаются взламывать разные программные продукты. Процесс взлома очень трудный, и именуется как “хакинг”, и порой длится от нескольких часов до нескольких лет.

В данной статье показан пример разбора одного проприетарного файла, имеющего формат _zar. Также будет реализовано приложение для автоматизации извлечения данных из файла на основе полученных данных на этапе разбора.

Для этой цели я выбрал среду программирования Delphi т.к. Delphi - оптимальный инструмент для создания приложений любой сложности. Оптимальный, т.к. поддерживает технологию визуальной разработки, которая позволяет существенно сократить время разработки (снизить стоимость, соответственно), при сохранении хорошего качества и надежности программного продукта.

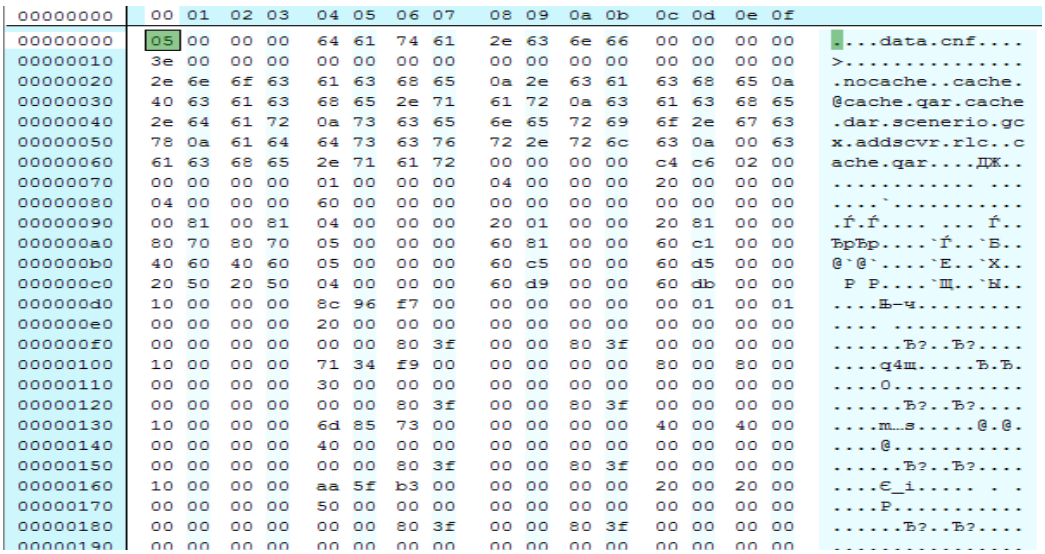
Плюсы языка:

- Низкие требования разработанного приложения к ресурсам компьютера;
- Быстрая скорость разработки;
- Быстродействие;
- наращиваемость за счет встраивания новых компонентов и инструментов в среду Delphi;

Передо мной была поставлена следующая задача:

Разобрать файл _zar, используя HEX-редактор, получить его структуру и на основе этой структуры разработать приложение для извлечения данных.

Откроем файл в HEX-редакторе и взглянем на его структуру (Рис.1):



00000000	00 01 02 03	04 05 06 07	08 09 0a 0b	0c 0d 0e 0f	
00000000	05 00 00 00	64 61 74 61	2e 63 6e 66	00 00 00 00	.....data.cnf....
00000010	3e 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	>..............
00000020	2e 6e 6f 63	61 63 68 65	0a 2e 63 61	63 68 65 0a	.nocache..cache.
00000030	40 63 61 63	68 65 2e 71	61 72 0a 63	61 63 68 65	@cache.qar.cache
00000040	2e 64 61 72	0a 73 63 65	6e 65 72 69	6f 2e 67 63	.dar.scenerio.gc
00000050	78 0a 61 64	64 73 63 76	72 2e 72 6c	63 0a 00 63	x.addscvr.rlc..c
00000060	61 63 68 65	2e 71 61 72	00 00 00 00	c4 c6 02 00	ache.qar....ДЖ..
00000070	00 00 00 00	01 00 00 00	04 00 00 00	20 00 00 00	....
00000080	04 00 00 00	60 00 00 00	00 00 00 00	00 00 00 00	....`...........
00000090	00 81 00 81	04 00 00 00	20 01 00 00	20 81 00 00	.ѓ.ѓ.... ... ѓ..
000000a0	80 70 80 70	05 00 00 00	60 81 00 00	60 c1 00 00	ЂpЂp....`ѓ..`Б..
000000b0	40 60 40 60	05 00 00 00	60 c5 00 00	60 d5 00 00	@`@`....`E..`Х..
000000c0	20 50 20 50	04 00 00 00	60 d9 00 00	60 db 00 00	P P....`Щ..`Ы..
000000d0	10 00 00 00	8c 96 f7 00	00 00 00 00	00 01 00 01	....Њ-ч.........
000000e0	00 00 00 00	20 00 00 00	00 00 00 00	00 00 00 00	....
000000f0	00 00 00 00	00 00 80 3f	00 00 80 3f	00 00 00 00	.......Ђ?..Ђ?....
00000100	10 00 00 00	71 34 f9 00	00 00 00 00	80 00 80 00	....q4щ.....Ђ.Ђ.
00000110	00 00 00 00	30 00 00 00	00 00 00 00	00 00 00 00	....0...........
00000120	00 00 00 00	00 00 80 3f	00 00 80 3f	00 00 00 00	.......Ђ?..Ђ?....
00000130	10 00 00 00	6d 85 73 00	00 00 00 00	40 00 40 00	....m…s.....@.@.
00000140	00 00 00 00	40 00 00 00	00 00 00 00	00 00 00 00	....@...........
00000150	00 00 00 00	00 00 80 3f	00 00 80 3f	00 00 00 00	.......Ђ?..Ђ?....
00000160	10 00 00 00	aa 5f b3 00	00 00 00 00	20 00 20 00	....Є_і..... . .
00000170	00 00 00 00	50 00 00 00	00 00 00 00	00 00 00 00	....P...........
00000180	00 00 00 00	00 00 80 3f	00 00 80 3f	00 00 00 00	.......Ђ?..Ђ?....
00000190	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	....

Рис.1 Структура файла

Первый байт – количество файлов в архиве. Затем сразу бросается в глаза таблица с именами файлов. Сразу после таблицы имён следует таблица размеров для этих имён, а уже потом сами данные. Следовательно, нам необходимо прочесть имя файла, офсет, перейти по нему, считав данные и записать их в новый файл.

Приступим к реализации приложения. Оно будет работать в консольном режиме.

Основные элементы программного кода:

```
AssignFile(zar,zarname);

Reset(zar);

BlockRead(zar,numbers,4);

for i:=1 to numbers do begin

cycle:=0;

REPEAT

BlockRead(zar,Outname[cycle],1);

Inc(cycle);

UNTIL Outname[cycle-1]=#0;

Writeln(Outname, ' - Имя файла');

a:=Filepos(zar);

Writeln(a, ' - позиция после считыввания имени');
```

Для начала необходимо получить доступ к архиву и считать количество файлов внутри в переменную. На основе этих данных, с помощью цикла for, идёт поблочное считывание имени. Так как после каждого имени идёт пустой символ, то с помощью конструкции UNTIL Outname[cycle-1]=#0, определяется, достигла ли программа конца имени или нет. Как только имя считано, программа запоминает позицию и переходит к размеру.

```
BlockRead(zar,Lenght,4);

Writeln(Lenght, ' - Размер');

a:=Filepos(zar);
```

Далее читается офсет и позиция снова записывается в переменную. Затем необходимо считать размер , а опираясь на него и сами данные и записать их в новый файл со считанным именем.

```
BlockRead(zar,buf,lenght);

AssignFile(Outfile,Outname);

Rewrite(Outfile);

BlockWrite(Outfile,buf,lenght);

Close(Outfile);
```

Процесс работы программы показан на Рис.2, а её результат на рис.3.:

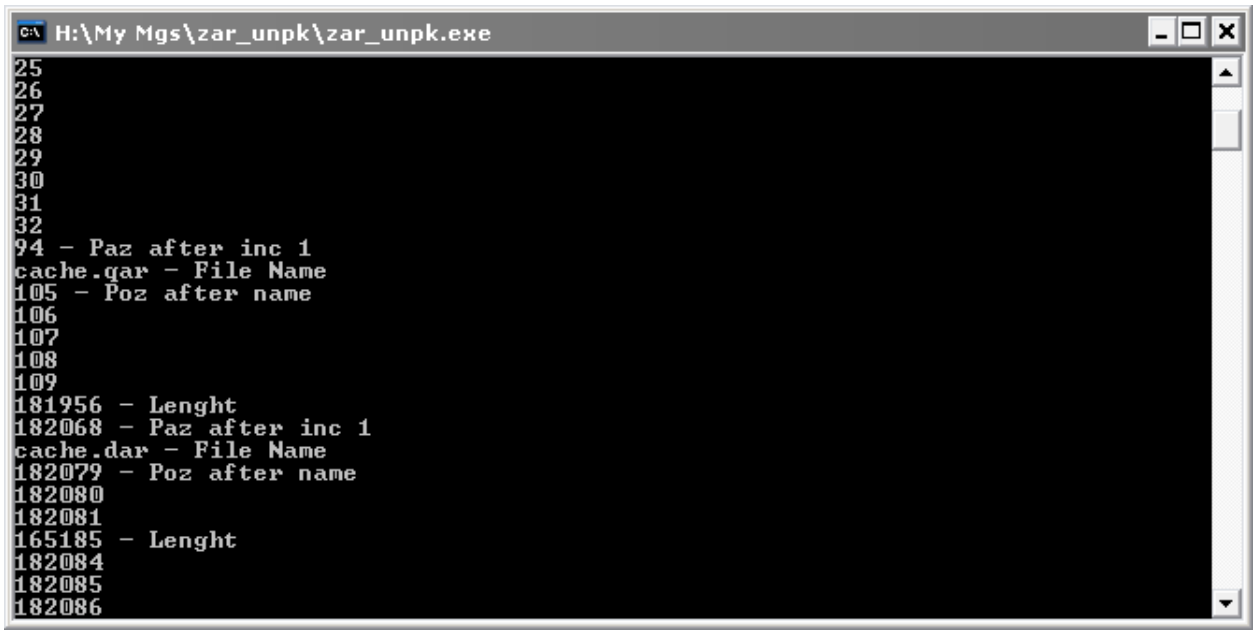


Рис2. Процесс работы программы

Рис3. Результат работы программы

Все 5 файлов были успешно извлечены.