УДК 004.451.52

ВНУТРЕННЯЯ ОРГАНИЗАЦИЯ ФАЙЛОВ РЕЕСТРА WINDOWS В ВЕРСИЯХ СЕРИИ NT

Похилец Н.В., Шевченко О.Г.
Донецкий национальный технический университет

Несмотря на то, что внешний интерфейс для работы с реестром Windows хорошо известен, детали его внутренней организации, и в частности формат двоичных файлов образующих реестр, остаются недокументированными. Знание формата этих файлов необходимо для извлечения данных из реестра после краха система, восстановления поврежденных или удаленных элементов реестра, дефрагментации реестра. В данном докладе рассматривается общее внутреннее строение файлов реестра.

Обзор файлов, составляющих реестр. Реестр Windows представляет собой древовидную структуру, с несколькими корневыми элементами, называемыми ульями (hives). Данные реестра хранятся в файлах, которые называются файлами ульев (hive files), но при этом их состав и количество не соответствуют ульям реестра.

Улей **HKEY_LOCAL_MASHINE** – не имеет файлового отображения и является виртуальным контейнером для подключей **SAM**, **SECURITY**, **SOFTWARE**, **SYSTEM**, которые хранятся отдельно в одноименных файлах (без расширения) в каталоге **%SYSTEMROOT%\system32\config**. Подключ **HARDWARE** является временным и существует только в памяти. Он содержит информацию о присутствующих в системе устройствах, создается при загрузке системы и обновляется при изменении конфигурации оборудования в процессе работы. После выключения системы его содержимое теряется.

Ульи **HKEY_PERFORMANCE_DATA**, **HKEY_PERFORMANCE_NLSTEXT** и **HKEY_PERFORMANCE_TEXT** содержат динамически изменяемые параметры, соответствующие счетчикам производительности системы. Эти ульи недоступны через редактор реестра.

Улей **HKEY_USERS** как и **HKEY_LOCAL_MASHINE**, также является всего лишь контейнером для подключей, в частности подключей настройки пользователей вошедших в систему и присутствующих в ней на данный момент. После выхода пользователя из системы соответствующий подключ выгружается. Каждому пользователю соответствуют два подключа – один с именем равным SID пользователя (загружается из файла **ntuser.dat** в каталоге пользователя), второй – с именем вида **XXXX_Classes,** являющегося псевдонимом для подключа **HKU\XXXX\Software\Classes,** который хранится в отдельном файле – **Local Settings\Application Data\Microsoft\Windows\UsrClass.dat** (в каталоге пользователя).

Отдельно следует отметить подключ **HCU\.DEFAULT** – он хранится в файле **%SYSTEMROOT%\system32\config\default** и является настройками пользователя, которые используются до входа в систему конкретного пользователя.

Улей **HKEY_CURRENT_USER** являются псевдонимом (ссылкой) для подключей улья **HKU**, соответвующего текущему пользователю.

Улей **HKEY_CURRENT_CONFIG** является псевдонимом (ссылкой) для подключа **HKLM\System\CurrentControlSet\Hardware Profiles\Current**.

Улей **HKEY_CLASSES_ROOT** образуется слиянием двух подключей – **HKLM\SOFTWARE\Classes** и **HKCU\Software\Classes**. При слиянии ключи

объединяются, а значения из **HKCU** переопределяют значения из **HKLM** [4]. Согласно [1], а также на основании практических исследований, редактирование элементов в улье **HKCR** происходит по следующим правилам: модификация/удаление значения в **HKCR** влияет на значение в том улье, из которого оно попало в **HKCU**; при удалении ключа удаляются соответствующие ключи из обоих источников; при создании нового ключа или раздела в улье **HKCR** – он создается в **HKCU**, если родительский ключ имеется в **HKCU**, и в **HKLM** в противном случае.

Общая структура файла улья. Структура файла улья имеет три логических уровня – блоки, ячейки и данные.

На уровне блоков происходит обмен данными между оперативной памятью и диском. Поэтому, с целью оптимизации этого обмена, все блоки имеют размер кратный 4Кб. Блоки расположены в файле последовательно без промежутков и перекрытий, поэтому размер всего файла также кратен 4 Кб. Проведенные практические эксперименты позволили получить представление о структуре исследуемых файлов.

Первый блок имеет сигнатуру “regf” и содержит заголовок файла, остальные являются блоками данных. Заголовок занимает только первые 512 байт блока, а остальные заполнены нулями (и согласно [2] и [3] игнорируется). Заголовок хранит временную метку обновления файла, информацию о местонахождении внутри файла корневого ключа, и верификационную информацию – суммарный размер всех блоков данных, контрольную сумму заголовка и два циклических счетчика обновлений – первый инкрементируется перед записью файла на диск, второй – после. Несовпадение этих значений свидетельствует о сбое во время записи файла на диск, и, следовательно, вероятном его повреждении.

Непосредственно за блоком заголовка, т.е. по смещению 4 Кб находится первый блок данных. Блоки данных имеют переменный размер, но всегда кратный 4 Кб, а в большинстве случаев и равный 4 Кб. Блоки большего размера создаются, если блока размером 4 Кб недостаточно для сохранения одной порции данных, что имеет место крайне редко.

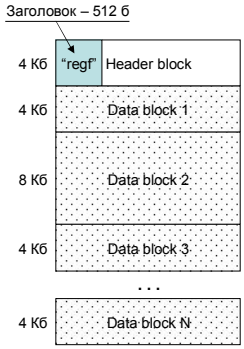

Рис. 1 – Структура файла улья на уровне блоков

Каждый блок данных состоит из 32-байтного заголовка и содержимого. В заголовке хранится сигнатура “hbin”, размер блока (включая заголовок) и смещение данного блока от первого блока.

Каждый блок данных является контейнером для единиц следующего логического уровня – ячеек. Ячейки являются единицей динамического выделения памяти в пределах одного блока. Ячейки могу быть либо пустыми, либо хранить данные. Данные могу занимать ячейку не полностью. Размер ячейки всегда кратен 4 байтам. Ячейка состоит из 4-х байтного поля размера и содержимого. Согласно [3], поле размера является знаковым 32-битным числом, в которое записывается размер

ячейки, включая само поле размера. При этом размер занятой ячейки записывается со знаком «минус», а свободной – со знаком «плюс».

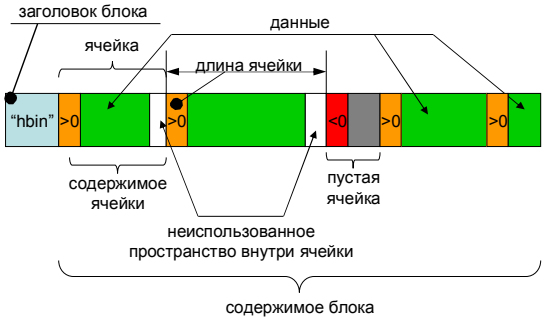

Рис. 2 – Ячейки внутри блока

Блоки и ячейки формируют фундамент для последнего уровня – уровня данных.

Организация данных в файле улья. На уровне данных файл улья состоит из различных структур, размещенных внутри ячеек. В качестве ссылок на эти структуры используются смещения внутри файла относительно начала первого блока данных. По этому смещению находится ячейка, содержимым которой является требуемая структура данных.

Корневой структурой данных является структура с сигнатурой “kn” (*key node*) – описатель ключа. Описатель ключа содержит ссылку на описатель родительского ключа, время последнего обновления ключа, имя ключа, количество значений и дочерних ключей, флаги, а также ссылки на списки значений и дочерних ключей, ссылку на имя класса для данного ключа и ссылку на дескриптор безопасности. Физически структура оканчивается именем, которое имеет переменную длину, поэтому и вся структура имеет переменную длину.

Дескрипторы безопасности хранятся в структурах с сигнатурой “ks” (*key security*). Такие структуры разделяются между несколькими ключами, с помощью счетчика ссылок. Кроме того, все дескрипторы безопасности в пределах одного файла объединяются в циклический двусвязный список.

Список значений в ключе представляет собой простой неструктурированный массив смещений к структурам, описывающим отдельные значения.

Список подключей является хеш-таблицей ссылок на подключи, отсортированной по значению хеша. Значения хеша используются при поиске конкретного подключа и игнорируются при полном переборе всех подключей.

Имеются три типа списков подключей. Список с сигнатурой “lf” – используется в Windows 2000 и в качестве значений хеша использует первые 4 байта имени. В последующих версиях ему на смену пришел список с сигнатурой “lh” – имеющий аналогичную структуру, но другую, более мощную хеш-фукнцию. И, наконец, редко встречающийся “ri”-список – список списков – содержит ссылки на другие списки.

Значения хранятся в структурах с сигнатурой “kv” (*key value*). Аналогично kn-структуре, kv-структура оканчивается именем переменной длины, а потому и длина всей структуры переменна. Кроме того, эта структура хранит в себе тип данных, размер данных, и ссылку на блок данных. При этом если размер данных не превышает 4 байтов, то сами данные записываются в поле ссылки на блок данных внутри kv-структуры, без выделения дополнительного блока.

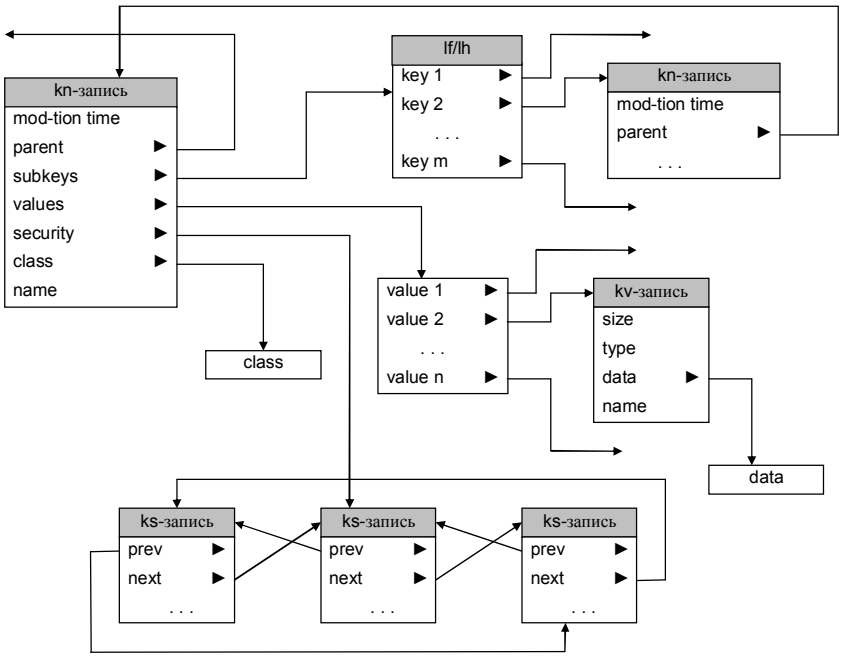

Рис. 3 – Связь между структурами данных в файле улья

Имена ключей и значений являются строками в кодировке Unicode, но при этом, если все символы имени принадлежат ASCII-подмножеству, то имя хранится в кодировке ASCII. Поскольку таких имен большинство, то это существенно уменьшает размер файлов реестра.

Программная реализация анализа файлов реестра позволила: уточнить структуры файлов в целом; понять назначение и использование отдельных полей элементов структур; получить представление об алгоритме выполнения стандартных операций с реестром.

Литература

[1] M.E. Russinovich, D.A. Solomon. *Microsoft® Windows® Internals, Fourth Edition: Microsoft Windows Server™ 2003, Windows XP, and Windows 2000.* Microsoft Press, 2004.

[2] Timothy D. Morgan. *The Windows NT Registry File Format*. August 2008. http://www.sentinelchicken.com/research/registry_format/.

[3] B. D. *WinReg.txt*. http://home.eunet.no/ pnordahl/ntpasswd/WinReg.txt.

[4] MSDN. Merged View of HKEY_CLASSES_ROOT. http://msdn.microsoft.com/en-us/library/ms724498(VS.85).aspx