УДК 004.453.5

Н.Н. Самарин

ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОЙ ДЛИНЫ ПОСЛЕДОВАТЕЛЬНОСТИ КОМАНД ПРОЦЕССОРА, ОПЕРИРУЮЩИХ С АДРЕСАМИ ОПЕРАТИВНОЙ ПАМЯТИ, ДЛЯ ПОЛУЧЕНИЯ ХАРАКТЕРИСТИЧЕСКОЙ СИГНАТУРЫ ВЫПОЛНЯЮЩЕЙСЯ ПРОГРАММЫ

Приведено обоснование оптимальной длинны последовательности команд процессора, работающего с областями оперативной памяти. Предложена кодировка бинарного кода значений обращений процессора к областям памяти в соответствии с таблицей Unicode. Показано что оптимальной величиной свертки строк без потери информативности является величина от 10 до 30 строк в одну.

Ключевые слова: оперативная память, процессор, таблица Unicode, интерпретация, последовательность команд, оптимальная длина.

При аудите программного обеспечения на недекларированные возможности, часто применяются системы виртуализации, которые позволяют на более низком уровне выполнять контроль за обращением программного обеспечения к физическим ресурсам вычислительных систем. Неоднократно говорилось, что использование программного обеспечения не прошедшего проверку на наличие недокументированных возможностей может приводить к техногенным катастрофам [2]. В [1] подробно описан алгоритм получения и интерпретации данных из журнала работы виртуальной машины. Исследования показали, что формализация процедуры обращения процессора к областям оперативной памяти вычислительной системы описывается математической моделью основанной на исчислении высказываний (1).

$$F = RI \,\&\, WIII \,\&\, XI \,\&\, WIV \,\&\, XI \,\&\, RII \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot XII \qquad (1)$$

Таблица 1

Таблица прерываний	**Данные BIOS**	**Память программ**	**Видеопамять**	**BIOS**	**ACPI**	**APIC**
1	0	0	0	0	0	0
0	1	0	0	0	0	0
1	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	0	0	1	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
1	0	0	0	0	0	0

Таблица 2

№ такта	Дополнение	Таблица прерываний	Данные BIOS	Память программ	Видеопамять	BIOS	ACPI	APIC
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0
8	0	0	0	0	1	0	0	0
9	0	0	0	0	0	1	0	0
10	0	1	0	0	0	0	0	0

Информация о работе вычислительной системы основанная на исследовании потактного обращения процессора к областям оперативной памяти представляется в виде матрицы размера 7 на N, где 7 – это количество областей оперативной памяти используемой операционной системой, а N – это количество тактов процессора, фрагмент матрицы показан в табл. 1.

Строка матрицы содержит информацию об изменении состояния ячейки конкретной области оперативной памяти, при этом остальные области не меняют своего состояния. Положив за бит значения строк матрицы получим семибитное представление значения каждой строки. Согласно таблице кодировки KOI-7 значения строк интерпретируется символами. Другими словами, «алфавит» из всех возможных (127) комбинаций формируется слова и предложения, характеризующие работу программы и вычислительной системы в целом. Таблица KOI-7 является кодировкой для русского языка основанной на ASCII. Дополним матрицу слева нулевым столбцом до получения полного байта, табл. 2.

Таблица 3

№ такта	Дополнение	Таблица прерываний	Данные BIOS	Память программ	Видеопамять	BIOS	ACPI	APIC	Значение
1	0	1	0	0	0	0	0	0	@
2	0	0	1	0	0	0	0	0	
3	0	0	0	0	1	0	0	0	BS
4	0	0	0	1	0	0	0	0	DLE
5	0	0	0	0	0	1	0	0	BS
6	0	0	1	0	0	0	0	0	
7	0	0	0	1	0	0	0	0	DLE
8	0	0	0	0	1	0	0	0	BS
9	0	0	0	0	0	1	0	0	EOT
10	0	1	0	0	0	0	0	0	@

Таблица 4

№ такта	Дополнение	Таблица прерываний	Данные BIOS	Память программ	Видеопамять	BIOS	ACPI	APIC
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0
8	0	0	0	0	1	0	0	0
9	0	0	0	0	0	1	0	0
10	0	1	0	0	0	0	0	0
Итог	0	1	1	1	1	1	0	0

Значению строки поставим в соответствие символ из таблицы Unicode (Юникод). Unicode позволяет закодировать символ 16 битами, в данном случае используется только 8 бит, оставшиеся 8 бит полагаются резервными, так как могут быть использованы при дальнейших исследованиях. Имеем следующее представление ячейки используемой области оперативной памяти – {0,X}, где 0 – резервный байт, X – соответствие значения строки матрицы символу по таблице приведено в табл. 3.

Из примера видно, что в большинстве случаев значение строки соответствует специальному символу, который не удобен в автоматической обработке, поэтому если получаемой (базовое) значение меньше пятидесяти, то прибавим к нему величину равную 50, если больше, то оставим без изменения. С учетом этого получаем интервал допустимых значений по таблице Unicode от 50 до 127.

Процессор вычислительной системы за 10 минут работы выполняет десятки миллионов тактов, следовательно, получаемое предложение будет очень громоздким и тяжелым для восприятия и дальнейшего анализа. Для уменьшения объема получаемой информации выполнена свертка строк по алгоритму: бинарной операции сложения, табл. 4.

Для выражения из десяти тактов N_{10} = 01111100, значением в десятичном виде 124 по таблице Юникод является символ вертикальная черта {|}. При выполнении свертки строк возможна потеря информации об обращении процессора к областям оперативной памяти. Для определения оптимального количество строк, подвергаемых свертке, были выполнены многочисленные эксперименты, в ходе которых установлено, что оптимальная свертка без потери информативности, осуществляется от 10 до 30 строк в одну. Значительные потери происходят при свертке более 150 строк в одну. Фрагмент оптимизированного выражения работы программы из журнала виртуальной машины имеет вид:

«F6F6F6F6F6F6F6VFVFVFVF6F6F6F6F6F6FBT6F6F6VFVF6F6F6F6F6F6F6V
FVFVFVF6F6F6F6F6F6FBPBT6F6F6VFVF6F6F6F6F6F6F6VFVFVFVF6F6F6F
6F6F6F6FBPBT6F6F6VFV».

Из результатов исследований следуют выводы:

1. Последовательность обращений процессора к областям памяти в вычислительной системе для получения лексического подчерка работающей программы следует кодировать при помощи таблицы Unicode.

2. Свертка строк без потери информативность осуществляется в интервале от 10 до 30 строк в одну.

Дальнейшие исследования предполагается вести в направлении автоматизации поиска сигнатур разных программ в общем алфавите работы вычислительной системы с целью возможного определения уникальной сигнатуры исследуемой программы без исходных текстов [3].

СПИСОК ЛИТЕРАТУРЫ

1. *Самарин Н.Н.* Поиск скрытых угроз реализуемых программным обеспечением без исходного кода. Материалы международной научно-технической конференции, ч. 5. – М., 2013. – С. 85–90.

2. *Кубрин С.С., Самарин Н.Н.* Результаты комплексного анализа программного обеспечения горнодобывающих компаний на недекларированные возможности. Материалы 11 Международной научной школы молодых ученых и специалистов. – М.: ИПКОН РАН, 2014. – С. 152–154.

3. *Самарин Н.Н.* Программный комплекс определения циклов в областях памяти электронной вычислительной системы с их автоматической регистрацией / Борисов А.В., Кубрин С.С. Заявка на свидетельство о государственной регистрации программы для ЭВМ, № 16123/0203/ПО от 30.12.2014. ГИАБ

КОРОТКО ОБ АВТОРЕ

Самарин Николай Николаевич – начальник научно-исследовательского отделения, e-mail: samarin_nik@mail.ru, Научно-исследовательский институт «Квант».

UDC 004.453.5

ESTIMATION OF OPTIMUM LENGTH FOR CHAIN OF PROCESSOR INSTRUCTIONS FOR ADDRESSES IN MEMORY SPACE TO OBTAIN THE CHARACTERISTIC SIGNATURE OF A RUNNING PROGRAM

Samarin N.N., Head of Department, e-mail: samarin_nik@mail.ru,
Scientific Research Institute «Kvant», Moscow, Russia.

The article describes the rationale for the optimal long sequences of commands CPU working area of RAM. The proposed encoding binary values of appeals of the processor to the memory in accordance with the Unicode table. It is shown that the optimal value of the convolution of the rows without information loss is the value of from 10 to 30 lines in one.

Key words: RAM, CPU, Unicode table, interpretation, sequence of commands, the optimal long.

REFERENCES

1. Samarin N.N. Poisk skrytykh ugroz realizuemykh programmnym obespecheniem bez iskhodnogo koda. *Materialy mezhdunarodnoy nauchno-tekhnicheskoy konferentsii*, ch. 5 (Implicit threat detection using a program release without source code. International Scientific-Technical Conference Proceedings, part 5), Moscow, 2013, pp. 85–90.

2. Kubrin S.S., Samarin N.N. Rezul'taty kompleksnogo analiza programmnogo obespecheniya gornodobyvayushchikh kompaniy na nedeklarirovannye vozmozhnosti. *Materialy 11 Mezhdunarodnoy nauchnoy shkoly molodykh uchenykh i spetsialistov* (Results of the integrated analysis of a program support for mining companies to identify undeclared capacities. Proceedings of 11th International Scientific School for Young Scientists and Specialists), Moscow, IPKON RAN, 2014, pp. 152–154.

3. Samarin N.N. Programmnyy kompleks opredeleniya tsiklov v oblastyakh pamyati elektronnoy vychislitel'noy sistemy s ikh avtomaticheskoy registratsiey (Bundled software for determination and autoregistration of rounds in memory space of computer installation).