Список литературы

1. Блог: программирование микроконтроллеров, настройка UART [Электронный ресурс]. URL: http://radioparty.ru/prog-avr/program-c/307-lesson-usart-avr Режим доступа: свободный (дата обращения: 15.01.2015).
2. Datasheet на микроконтроллер ATmega16.

УДК 004

РАЗРАБОТКА ДРАЙВЕРА ДЛЯ ПРЯМОГО ДОСТУПА К ФИЗИЧЕСКОЙ ПАМЯТИ ОЗУ ДЛЯ ОПЕРАЦИОННЫХ СИСТЕМ WINDOWS XP-8.1

А.Г. Черемнов
Научный руководитель: И.А. Тутов, ассистент кафедры ИКСУ ИК ТПУ
Томский политехнический университет, 634050, Россия, г. Томск, пр. Ленина, 30
E-mail: 8xandr@gmail.ru

Implementation of kernel driver for direct access of physical space of RAM are considered in the paper. The algorithm of realization with a detailed description of each point is presented in this paper.

Key words: Direct access of physical RAM, system space of OS, system programming.
Ключевые слова: Прямой доступ к физической памяти, системное пространство ОС, системное программирование.

Необходимость прямого доступа к физической памяти возникает в таких программах как [1–5] для ускорения работы с памятью и скорости обмена между оперативной памятью и памятью графических ускорителей путём самостоятельного выделения оптимальных с точки зрения быстродействия структур данных и организации сжатия и распаковки данных, а также в задачах анализа вредоносного кода или исполняемых файлов и модулей, имеющих достаточно сложные упаковщики и протекторы кода, из-за которых получение корректного ассемлерного кода не представляется возможным до размещения этого кода непосредственно в оперативную память.

Разработанный драйвер для прямого доступа представляет собой функциональный драйвер, после инициализации которого создаётся виртуального устройство XandrIO для организации взаимодействия между пространствами ядра и пользователя операционной системы Windows.

В процессе отладки использовались следующие инструменты:

- Static Driver Verifier (SDV);
- PREFast for Drivers (PFD);

Для глубокой проверки и отладки исполняемого кода драйвера применялся SDV, позволяющий отслеживать через Windows Driver Model исполнение вызов функций [6]. Для поверхностного анализа операций сегмента кода драйвера использовался PFD, анализировались проблемы с утечкой памяти, например, переполнение буфера.

Отметим, что PFD гораздо быстрее, чем SDV, так как SDV применяется для каждой функции отдельно [6].

В качестве примера приведён фрагмент кода, осуществляющий получение доступа к физической памяти и осуществляющий трансляцию физического адреса в виртуальный для архитектур с 32-разрядной шинной адреса между оперативной памятью и центральным процессором.

```
case IOCTL_GIVEIO_MAPPHYSTOLIN:

        if (dwInputBufferLength)
        {
                memcpy (&PhysStruct, pvIOBuffer, dwInputBufferLength);
                …
                ntStatus = MapPhysicalMemoryToLinearSpace((PVOID)PhysStruct.pvPhysAddress,
                        (SIZE_T)PhysStruct.dwPhysMemSizeInBytes,
                        (PVOID *)&PhysStruct.pvPhysMemLin,
                        (HANDLE *)&PhysStruct.PhysicalMemoryHandle,
                        (PVOID *)&PhysStruct.pvPhysSection);
                if (NT_SUCCESS(ntStatus))
                {
                        memcpy (pvIOBuffer, &PhysStruct, dwInputBufferLength);
                        Irp->IoStatus.Information = dwInputBufferLength;
                }

                Irp->IoStatus.Status = ntStatus;
        }
        else
                Irp->IoStatus.Status = STATUS_INVALID_PARAMETER;
```

Механизм IRP – пакетов использовался для организации обработки прерываний стеком виртуального устройства.

32-разрядная и 64-разрядная версии драйверов отличаются различными системными вызовами для операционной системы Windows, а также узкоспециализированными ассемблерными вставками, основное назначение которых заключается в организации процесса блокировок в случае обращения к функциям драйвера нескольких процессов.

Разработанный драйвер позволяет получить доступ к физической памяти ОЗУ, сделать дамп процесса из оперативной памяти, для его дальнейшего анализа.

Список литературы

1. Аврамчук В.С., Лунева Е.Е., Черемнов А.Г. Оптимизация расчета частотно-временной корреляционной функции на центральном процессоре // Системы управления и информационные технологии. – 2014. – № 2 (56). – С. 58–62.

2. Аврамчук В.С., Лунева Е.Е., Черемнов А.Г. Повышение эффективности использования аппаратных ресурсов ЭВМ при вычислении частотно-временной корреляционной функции [Электронный ресурс] // Интернет журнал Науковедение. – 2013. – № 6 (19). – С. 1–10. – Режим доступа: http://naukovedenie.ru/PDF/26TVN613.pdf.

3. Аврамчук В.С., Лунева Е.Е., Черемнов А.Г. Способы повышения эффективности вычисления быстрого преобразования Фурье [Электронный ресурс] // Науковедение. – 2013. – № 3. – С. 1–6. – Режим доступа: http://naukovedenie.ru/PDF/16tvn313.pdf.

4. Faerman V.A., Cheremnov A.G., Avramchuk V.S., Luneva E.E. Prospects of frequency-time correlation analysis for detecting pipeline leaks by acoustic emission method // IOP Conference Series: Earth and Environmental Science. – 2014. – Vol. 21. – Issue 1. – p. 12041.

5. Avramchuk V.S., Luneva E.E., Cheremnov A.G. Increasing the Efficiency of Using Hardware Resources for Time-Frequency Correlation Function Computation // Advanced Materials Research. – 2014. – Vol. 1040. – P. 969–974.

6. Penny Orwick, Guy Smith. Developing Drivers with the Windows Driver Foundation. – Microsoft, 2008. – P. 880.