



СТРУКТУРА ДРАЙВЕР УСТРОЙСТВА

Даурбекова Ася Мухтаровна

старший преподаватель, Ингушский государственный университет, РФ, г. Магас

Газдиева Мадина Алиевна

студент, Ингушский государственный университет, РФ, г. Магас

Аннотация. В данной статье рассматривается структура драйвер устройства. А также рассматриваются основные процедуры драйвера.

Ключевые слова: драйвер, процедуры драйверов, структура драйверов

Структура драйвер устройства

Драйвер устройства состоит из набора процедур, вызываемых на различных этапах обработки запроса ввода-вывода.

Перечислим основные процедуры драйвера:

- 1) Инициализирующая процедура.** Обязательная процедура драйвера, обычно называемая DriverEntry, которая выполняет инициализацию драйвера. Эта процедура регистрирует остальные процедуры драйвера в диспетчере ввода-вывода, заполняет системные структуры данных информацией о драйвере.
- 2) Процедура добавления устройства.** Реализуется только в устройствах, поддерживающих принцип Plug and Play. С помощью этой процедуры диспетчер PnP отправляет драйверу уведомление об обнаружении устройства, за которое отвечает драйвер. Когда процедура выполняется, драйвер создает объект устройства, описывающий данное аппаратное устройство.
- 3) Процедуры диспетчеризации.** Это основные процедуры, предоставляемые драйвером для чтения или записи на устройство. Диспетчер ввода-вывода вызывает драйвер для выполнения операции ввода-вывода с помощью одной из его процедур диспетчеризации.
- 4) Процедура инициализации ввода-вывода.** Драйвер инициирует передачу данных на устройство или получение данных с устройства. Эта процедура необходима только для драйверов, которым нужно обрабатывать только один запрос за раз, а не несколько запросов ввода-вывода, таких как драйвер клавиатуры. Обратите внимание, что большинство драйверов обрабатывают несколько IRP.
- 5) процедура обслуживания прерывания (ISR).** Когда устройство генерирует прерывание, диспетчер прерываний ядра передает управление этой подпрограмме. ISR работают на уровне IRQL устройства, поэтому выполняют наименьший объем работы, чтобы не блокировать обслуживание прерываний с более низким IRQL. Чтобы выполнить остальную часть не критичной по времени части обработки прерывания, ISR ставит DPC в

очередь. ISR присутствуют только в драйверах устройств, управляемых прерываниями; другие драйверы, такие как драйвер файловой системы, не имеют ISR.

6) DPC-процедура обработки прерываний. DPC-процедура выполняет основную часть обработки прерывания, оставшуюся после выполнения ISR. DPC-процедура отслеживает завершение текущей операции ввода-вывода на устройстве и обеспечивает корректное выполнение следующей операции ввода-вывода на устройстве.

7) Одна или несколько процедур завершения ввода-вывода. Служат для уведомления драйвера о завершении обработки IRP драйвером более низкого уровня. Например, диспетчер ввода-вывода вызывает процедуру завершения ввода-вывода драйвера файловой системы, когда драйвер диска заканчивает запись данных на диск. Эта процедура сообщает драйверу файловой системы об удачном или неудачном завершении операции записи или об ее отмене, а также дает разрешение драйверу на освобождение ресурсов.

8) Процедура отмены ввода-вывода. Если операция ввода-вывода может быть отменена, драйвер определяет одну или более процедур отмены ввода-вывода. Получив для обработки IRP, который может быть отменен, драйвер должен иметь средство, предоставляемое диспетчеру ввода-вывода для отмены IRP. Если поток, выдавший запрос на ввод-вывод, завершается до окончания обработки запроса или отменяет операцию (например, вызывает функцию `CancelIO`), то диспетчер ввода-вывода выполняет процедуру отмены, предоставленную драйвером для данного IRP. Процедура отмены отвечает за выполнение любых действий, необходимых для освобождения ресурсов, выделенных при обработке IRP, и за завершение IRP со статусом отмены.

9) Процедура выгрузки. Освобождает все системные ресурсы, задействованные драйвером, как правило, освобождаются ресурсы, выделенные процедурой инициализации.

10) Процедуры регистрации ошибок. Уведомляют об ошибках диспетчер ввода-вывода, который записывает информацию в журнал ошибок.

Ограничения, налагаемые на драйвер:

1. Драйвер режима ядра не может использовать API пользовательского уровня, только функции ядра.
2. Драйвер не может оперировать числами с плавающей точкой, так как мнемоники регистров FPU и MMX одинаковы, и драйвер не знает, какие именно регистры имеются в виду. При использовании – аварийный останов системы.
3. Нельзя работать с физической памятью напрямую. Можно получить виртуальный адрес и работать с ним.
4. Драйвер не должен работать длительное время при высоком уровне IRQL.