

Міністерство освіти і науки України
Національний авіаційний університет
Навчально-науковий інститут комп'ютерних інформаційних технологій
Кафедра комп'ютеризованих систем управління

Лабораторна робота №1.5
з дисципліни «Системне програмне забезпечення»
на тему «Ознайомлення з середовищами розробки драйверів»

Виконав:
студент ННІКІТ
групи СП-325
Клокун В. Д.
Перевірів:
Гармай В. М.

Київ 2018

1 МЕТА РОБОТИ

Ознайомлення з середовищами розробки драйверів, основними відомостями про програмну структуру драйверів та засобами їх створення.

2 ЗАВДАННЯ РОБОТИ

Створити простий драйвер та використати його у комп'ютерній системі.

3 ХІД РОБОТИ

Запускаємо середовище Microsoft Visual Studio та створюємо у ньому проект драйвера Kernel Mode Driver (KMDF). У створений проект статичної бібліотеки додаємо файли `Device.c` (ліст. 3.1), `Driver.c` (ліст. 3.2) та `Queue.c` (ліст. 3.3).

Лістинг 3.1: Файл `Device.c`

```
1  #include "driver.h"
2  #include "device.tmh"
3
4  #ifdef ALLOC_PRAGMA
5  #pragma alloc_text (PAGE, y03s01syssoftlab05CreateDevice)
6  #endif
7
8  NTSTATUS
9  y03s01syssoftlab05CreateDevice(
10     _Inout_ PWDFDEVICE_INIT DeviceInit
11 )
12
13 {
14     WDF_OBJECT_ATTRIBUTES deviceAttributes;
15     PDEVICE_CONTEXT deviceContext;
16     WDFDEVICE device;
17     NTSTATUS status;
18
19     PAGED_CODE();
20
21     WDF_OBJECT_ATTRIBUTES_INIT_CONTEXT_TYPE(&deviceAttributes,
22     ↵    DEVICE_CONTEXT);
23
24     status = WdfDeviceCreate(&DeviceInit, &deviceAttributes, &device);
25
26     if (NT_SUCCESS(status)) {
27         deviceContext = DeviceGetContext(device);
28     }
```

```

29         deviceContext->PrivateDeviceData = 0;
30
31         status = WdfDeviceCreateDeviceInterface(
32             device,
33             &GUID_DEVINTERFACE_y03s01syssoftlab05,
34             NULL // ReferenceString
35         );
36
37         if (NT_SUCCESS(status)) {
38             status = y03s01syssoftlab05QueueInitialize(device);
39         }
40     }
41
42     return status;
43 }

```

Лістинг 3.2: Файл Driver.c

```

1  #include "driver.h"
2  #include "driver.tmh"
3
4  #ifdef ALLOC_PRAGMA
5  #pragma alloc_text (INIT, DriverEntry)
6  #pragma alloc_text (PAGE, y03s01syssoftlab05EvtDeviceAdd)
7  #pragma alloc_text (PAGE, y03s01syssoftlab05EvtDriverContextCleanup)
8  #endif
9
10 NTSTATUS
11 DriverEntry(
12     _In_ PDRIVER_OBJECT DriverObject,
13     _In_ PUNICODE_STRING RegistryPath
14 )
15 {
16     WDF_DRIVER_CONFIG config;
17     NTSTATUS status;
18     WDF_OBJECT_ATTRIBUTES attributes;
19
20     WPP_INIT_TRACING(DriverObject, RegistryPath);
21
22     TraceEvents(TRACE_LEVEL_INFORMATION, TRACE_DRIVER, "%!FUNC! Entry");
23
24     WDF_OBJECT_ATTRIBUTES_INIT(&attributes);
25     attributes.EvtCleanupCallback =
26         y03s01syssoftlab05EvtDriverContextCleanup;
27
28     WDF_DRIVER_CONFIG_INIT(&config,
29         y03s01syssoftlab05EvtDeviceAdd

```

```

29         );
30
31     status = WdfDriverCreate(DriverObject,
32                             RegistryPath,
33                             &attributes,
34                             &config,
35                             WDF_NO_HANDLE
36                             );
37
38     if (!NT_SUCCESS(status)) {
39         TraceEvents	TRACE_LEVEL_ERROR, TRACE_DRIVER,
40         ↪ "WdfDriverCreate failed %!STATUS!", status);
41         WPP_CLEANUP(DriverObject);
42         return status;
43     }
44
45     TraceEvents	TRACE_LEVEL_INFORMATION, TRACE_DRIVER, "%!FUNC! Exit");
46
47     return status;
48 }
49
50 NTSTATUS
51 y03s01syssoftlab05EvtDeviceAdd(
52     _In_ WDFDRIVER Driver,
53     _Inout_ PWDFDEVICE_INIT DeviceInit
54 )
55 {
56     NTSTATUS status;
57
58     UNREFERENCED_PARAMETER(Driver);
59
60     PAGED_CODE();
61
62     TraceEvents	TRACE_LEVEL_INFORMATION, TRACE_DRIVER, "%!FUNC! Entry");
63
64     status = y03s01syssoftlab05CreateDevice(DeviceInit);
65
66     TraceEvents	TRACE_LEVEL_INFORMATION, TRACE_DRIVER, "%!FUNC! Exit");
67
68     return status;
69 }
70
71 VOID
72 y03s01syssoftlab05EvtDriverContextCleanup(
73     _In_ WDFOBJECT DriverObject
74 )
75 {
76     UNREFERENCED_PARAMETER(DriverObject);

```

```

76
77     PAGED_CODE();
78
79     TraceEvents(TRACE_LEVEL_INFORMATION, TRACE_DRIVER, "%!FUNC! Entry");
80
81     WPP_CLEANUP(WdfDriverWdmGetDriverObject((WDFDRIVER)DriverObject));
82 }

```

Лістинг 3.3: Файл Queue.c

```

1  #include "driver.h"
2  #include "queue.tmh"
3
4  #ifdef ALLOC_PRAGMA
5  #pragma alloc_text (PAGE, y03s01syssoftlab05QueueInitialize)
6  #endif
7
8  NTSTATUS
9  y03s01syssoftlab05QueueInitialize(
10     _In_ WDFDEVICE Device
11 )
12 {
13     WDFQUEUE queue;
14     NTSTATUS status;
15     WDF_IO_QUEUE_CONFIG queueConfig;
16
17     PAGED_CODE();
18
19     WDF_IO_QUEUE_CONFIG_INIT_DEFAULT_QUEUE(
20         &queueConfig,
21         WdfIoQueueDispatchParallel
22     );
23
24     queueConfig.EvtIoDeviceControl = y03s01syssoftlab05EvtIoDeviceControl;
25     queueConfig.EvtIoStop = y03s01syssoftlab05EvtIoStop;
26
27     status = WdfIoQueueCreate(
28         Device,
29         &queueConfig,
30         WDF_NO_OBJECT_ATTRIBUTES,
31         &queue
32     );
33
34     if(!NT_SUCCESS(status)) {
35         TraceEvents(TRACE_LEVEL_ERROR, TRACE_QUEUE,
36             ↪ "WdfIoQueueCreate failed %!STATUS!", status);
37     }
38     return status;
39 }

```

```

37     }
38
39     return status;
40 }
41
42 VOID
43 y03s01syssoftlab05EvtIoDeviceControl(
44     _In_ WDFQUEUE Queue,
45     _In_ WDFREQUEST Request,
46     _In_ size_t OutputBufferLength,
47     _In_ size_t InputBufferLength,
48     _In_ ULONG IoControlCode
49 )
50 {
51     TraceEvents(TRACE_LEVEL_INFORMATION,
52                 TRACE_QUEUE,
53                 "!FUNC! Queue 0x%p, Request 0x%p OutputBufferLength %d\
54                 InputBufferLength %d IoControlCode %d",
55                 Queue, Request, (int) OutputBufferLength, (int)
56                 ↵ InputBufferLength, IoControlCode);
57
58     WdfRequestComplete(Request, STATUS_SUCCESS);
59
60     return;
61 }
62
63 VOID
64 y03s01syssoftlab05EvtIoStop(
65     _In_ WDFQUEUE Queue,
66     _In_ WDFREQUEST Request,
67     _In_ ULONG ActionFlags
68 )
69 {
70     TraceEvents(TRACE_LEVEL_INFORMATION,
71                 TRACE_QUEUE,
72                 "!FUNC! Queue 0x%p, Request 0x%p ActionFlags %d",
73                 Queue, Request, ActionFlags);
74
75     return;
76 }

```

Після зборки проекту починаємо установку драйвера у систему за допомогою утиліти `devcon.exe`, виконавши у командному рядку таку команду:

```

devcon.exe install "y03s01-syssoft-lab-05.inf"
↵ "Root\y03s01-syssoft-lab-05"

```

Після завершення роботи команди перевіряємо результат установки драйвера у Диспетчері пристроїв (рис. 1)

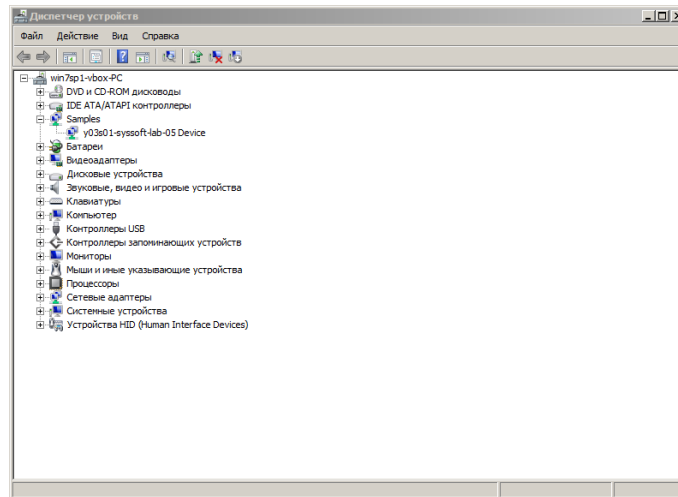


Рис. 1: Перелік пристроїв у Диспетчері пристроїв після встановлення створеного драйвера

4 ВИСНОВОК

Виконуючи дану лабораторну роботу ми ознайомились з середовищами розробки драйверів, основними відомостями про програмну структуру драйверів та способами їх створення.