# 回顾

如果kv不好使，还可以试试 r esp + dds esp

.cxr context 和 dt context 是等价的

.trap trapfram 不一样，具体哪不一样不太清楚。

# 二、课堂笔记

这节课讲了使用常规方式与驱动通信。

驱动通信模型和R3的进程-窗口有点类似：

|  |  |
| --- | --- |
| R3 | R0 |
| 进程 | 驱动 |
| 窗口 | 设备 |
| 给窗口发消息 | 给驱动发IRP |

具体流程是，R3根据名字找到设备，给设备发消息，消息在R0某个函数里打包成IRP，然后调用驱动注册的回调函数。驱动负责实现这些回调函数，并决定要做什么。

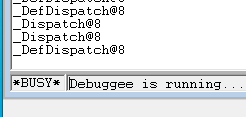
然后直接看代码：

驱动：

|  |
| --- |
| #include <ntifs.h>  #define \_DEVICE\_NAME L"\\device\\hambagadev"  #define \_SYM\_NAME L"\\??\\hambagadev"  NTSTATUS DefDispatch(struct \_DEVICE\_OBJECT\* DeviceObject, struct \_IRP\* Irp)  {  DbgPrintEx(77, 0, "%s\r\n", \_\_FUNCDNAME\_\_);  IoCompleteRequest(Irp, 0);  return STATUS\_SUCCESS;  }  NTSTATUS Dispatch(struct \_DEVICE\_OBJECT\* DeviceObject, struct \_IRP\* Irp)  {  DbgPrintEx(77, 0, "%s\r\n",\_\_FUNCDNAME\_\_);  IoCompleteRequest(Irp, 0);  return STATUS\_SUCCESS;  }  VOID DriverUnload(PDRIVER\_OBJECT pDriver)  {  if (pDriver->DeviceObject)  {  UNICODE\_STRING UnSymName;  RtlInitUnicodeString(&UnSymName, \_SYM\_NAME);  IoDeleteSymbolicLink(&UnSymName);  IoDeleteDevice(pDriver->DeviceObject);  }  }  NTSTATUS DriverEntry(PDRIVER\_OBJECT pDriver, PUNICODE\_STRING pReg)  {  // 初始化设备名  UNICODE\_STRING UnDeviceName;  RtlInitUnicodeString(&UnDeviceName, \_DEVICE\_NAME);  // 设备符号名  UNICODE\_STRING UnSymName;  RtlInitUnicodeString(&UnSymName, \_SYM\_NAME);  // 创建非独占的未知设备  PDEVICE\_OBJECT pDevice = NULL;  NTSTATUS st = IoCreateDevice(pDriver, 0, &UnDeviceName, FILE\_DEVICE\_UNKNOWN, FILE\_DEVICE\_SECURE\_OPEN, FALSE, &pDevice);  if (!NT\_SUCCESS(st))  {  return STATUS\_UNSUCCESSFUL;  }  // 创建符号链接  st = IoCreateSymbolicLink(&UnSymName, &UnDeviceName);  if (!NT\_SUCCESS(st))  {  IoDeleteDevice(pDevice);  return STATUS\_UNSUCCESSFUL;  }  // 设置设备标志，去掉初始化表，使用 buffer IO  pDevice->Flags &= ~DO\_DEVICE\_INITIALIZING;  pDevice->Flags |= DO\_BUFFERED\_IO;  // 设置回调  pDriver->MajorFunction[IRP\_MJ\_CREATE] = DefDispatch;  pDriver->MajorFunction[IRP\_MJ\_CLOSE] = DefDispatch;  pDriver->MajorFunction[IRP\_MJ\_DEVICE\_CONTROL] = Dispatch;  pDriver->DriverUnload = DriverUnload;  return STATUS\_SUCCESS;  } |

R3：

|  |
| --- |
| #include <windows.h>  #include <stdio.h>  #define \_SYM\_NAME "\\\\.\\hambagadev"  int main()  {  HANDLE hDevice = CreateFileA(\_SYM\_NAME, GENERIC\_READ | GENERIC\_WRITE, FILE\_SHARE\_READ | FILE\_SHARE\_WRITE, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);  if (!hDevice)  {  printf("打开设备失败 %d\n", GetLastError());  return 1;  }  system("pause");  ULONG x = 0;  ULONG retLen = 0;  BOOL bSuccess = DeviceIoControl(hDevice, 0, &x, sizeof(x), &x, sizeof(x), &retLen, 0);  printf("success %d %d\n", bSuccess, GetLastError());  system("pause");  bSuccess = DeviceIoControl(hDevice, 0, &x, sizeof(x), &x, sizeof(x), &retLen, 0);  printf("success %d %d\n", bSuccess, GetLastError());  system("pause");  return 0;  } |



# 三、回调函数类型

