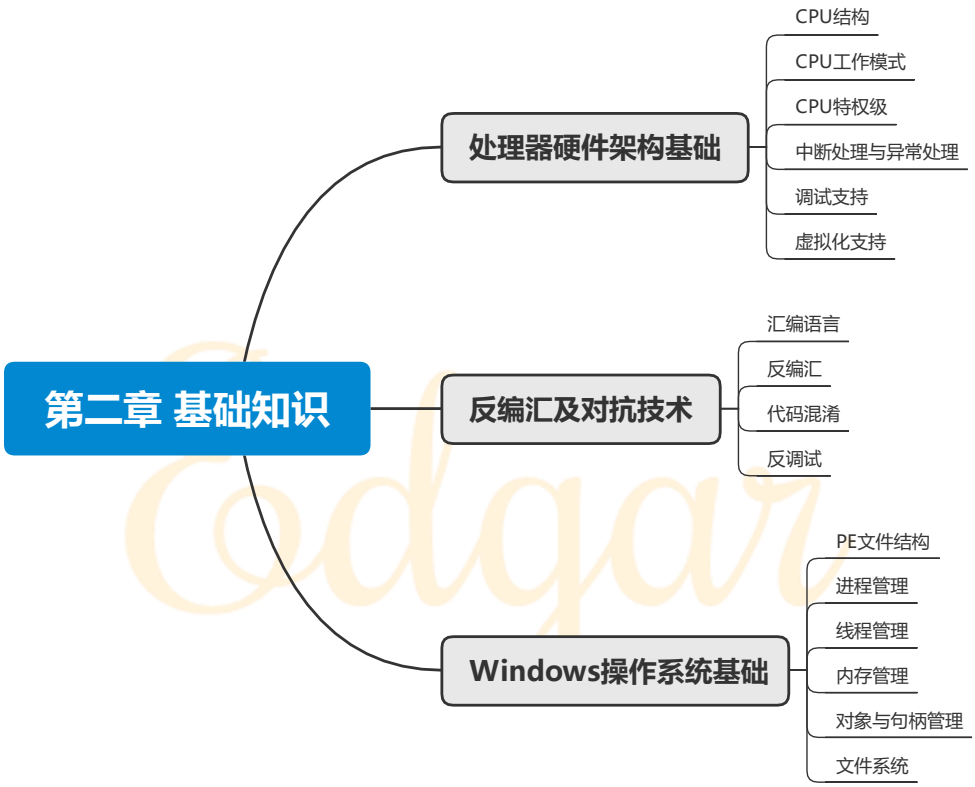
第二章 基础知识

一. 章节主要内容



二. 详细内容

2.1 处理器硬件架构基础

2.1.1 CPU 结构介绍

CPU 是计算机中央处理器的简称，控制着计算机的**操作和执行数据**处理功能，结构包括：

- ✧ 寄存器：提供 CPU 内部储存，用来暂时存放参与运算的数据及运算结果
- ✧ 算数逻辑单位(ALU)：执行计算机的运算功能，包括加减乘除四则运算，左右移位运算，与或非异逻辑运算等
- ✧ 控制器控制计算机各部件工作，包括**取指、译码、执行**
- ✧ 内部总线：将寄存器、ALU 以及控制器进行互连，提供通信机制

IA-32 的 CPU 寄存器包括

- ✧ 指令指针寄存器：EIP 寄存器，存储了当前执行指令的地址，系统根据该寄存器进行寻址，从内存中取出指令，然后在译码、执行
- ✧ 通用数据寄存器：包括 EAX，ECX，EDX，EBX，通常用于储存参与运算的数据及运算的结果
- ✧ 地址指针寄存器：包括 ESP 和 EBP，ESP 记录了当前的栈顶，EBP 通常记录的是当前函数的栈底
- ✧ 变址指针寄存器：包括 ESI 和 EDI，通常 ESI 是操作数源地址，EDI 是操作数目的地址
- ✧ 标志位寄存器：通称为 EFLAGS，具体见 P26
- ✧ 段寄存器；包括代码段寄存器 CS，数据段寄存器 DS，堆栈段寄存器 SS，附加段寄存器 ES，FS，GS
- ✧ 控制寄存器：包括 CR0、CR1、CR2、CR3、CR4，用于记录处理器的运行模式和当前执行任务的属性

✧ ……

2.1.2 保护模式

IA-32 架构的 CPU 有两种工作模式：

✧ 实模式：不支持多线程，不能实现权限分级

✧ 保护模式：实现内存分页和权限分级，支持多线程，多任务

分页功能由 CR3 寄存器支持，分段由内存管理器(GDTR, IDTR, LDTR, TR)支持，具体见 P28-29

2.1.3 特权级

CPU 支持 Ring0、Ring1、Ring2、Ring3 共 4 个权限级别，Ring0 最高，Ring3 最低，Windows 中只使用 Ring0 和 Ring3，为了进行代码段和数据段的特权级检验，需要 3 种类型的特权级支持：

✧ CPL：当前特权级，是当前执行线程的特权级，存储在 CS 段寄存器和 SS 段寄存器的第 0 位和第 1 位，通常情况下 CPL 与当前指令所在代码段的特权级相等

✧ DPL：描述特权级，段或门的特权级，存储在段或门的描述符的 DPL 域中

✧ RPL：请求特权级，赋给段选择子的取代性特权级，储存在段选择子的第 0、1 位

其他特权级指令见 P30

2.1.4 中断处理与异常处理

中断和异常是程序执行过程中的插曲，需要处理器强制暂停当前任务，转移到一个称为中断处理程序或者异常处理程序的特殊任务中。

中断是程序执行期间随机发生的，可以是对硬件信号的响应，也可以是软件中断；**异常**是处理器执行指令过程中发生错误情况时产生的。

IA-32 架构的中断处理机制为收到中断信号或者检测到异常时，处理器挂起当前运行的进程或任务，保存好任务现场后转而去执行中断或者异常处理程序，处理完后恢复现场，继续执行被中断的进程或任务。

引起中断产生的原因或来源称为中断源，包括**硬件中断**和**软件中断**；引起异常产生的原因或来源被称为异常源，包括**处理器检测到程序错误异常**、**软件产生的异常**和**机器检测异常**。

2.1.5 调试支持

IA-32 架构的 CPU 中标志位寄存器 EFLAGS 中的 IF、TF 用于调试模式的开启，将 TF 置于 1 使 CPU 处于单步执行状态，IF 置于 1 使 CPU 开启中断响应。

CPU 中设置了 DR0~DR7 共 8 个调试寄存器用于断点设置功能，DR0~DR3 位断点地址寄存器，用来保存断点地址，DR4，DR5 保留为 DR6 和 DR7 的别名寄存器，DR6 为调试状态寄存器，DR7 位调试控制寄存器。

2.1.6 虚拟化支持

虚拟化技术能够基于系统 CPU、内存、磁盘等资源虚拟出多台主机，提高资源利用率，最大化利用平台的硬件资源。

虚拟化技术典型有 Intel 的 VT 技术(VT-X, VT-D, VT-C)和 AMD 公司的 AMD-V 技术。

2.2 反汇编及对抗技术

2.2.1 汇编语言

- 寻址方式：
  - **寄存器寻址**：最通用的数据寻址方式，使用寄存器的别名作为操作数，支持 8、16、32、64 位的操作数长度
  - **立即寻址**：指令的源操作数为立即数，支持 8、16、32、64 位
  - **直接数据寻址**：将位移量加到默认数据段地址或其他段地址上形成地址
  - **寄存器间接寻址**：通过 BP、BX、DI、SI 等保存偏移地址的一种寻址方式

- ✧ **基址加变址寻址：**需要基址寄存器(EBP、EBX)和变址寄存器(EDI、ESI)叠加使用进行寻址
- ✧ **寄存器相对寻址：**用位移量加基址或变址寄存器的内容寻址数据段中存储的数据
- ✧ **相对基址加变址寻址：**用基址寄存器和变址寄存器加位移量组成存储器地址，通常用来寻址存储器中的二维数据数据
- ✧ **比例变址寻址：**使用两个 32 位寄存器(基址寄存器和变址寄存器)，第二个寄存器与比例因子(1、2、4、8)相乘进行数据寻址

➢ 常用的汇编指令：

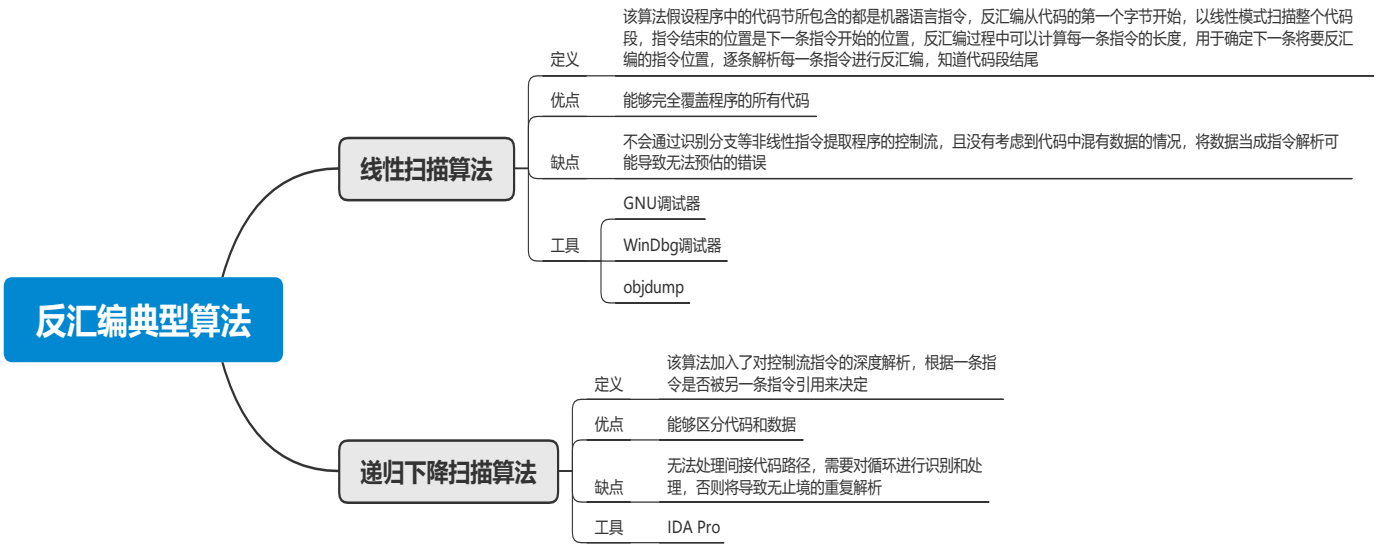
2.2.2 反汇编

反汇编是将机器语言转换成汇编代码的过程，将人类难以理解的机器语言转换具有符号语义的指令语言。

反汇编流程：

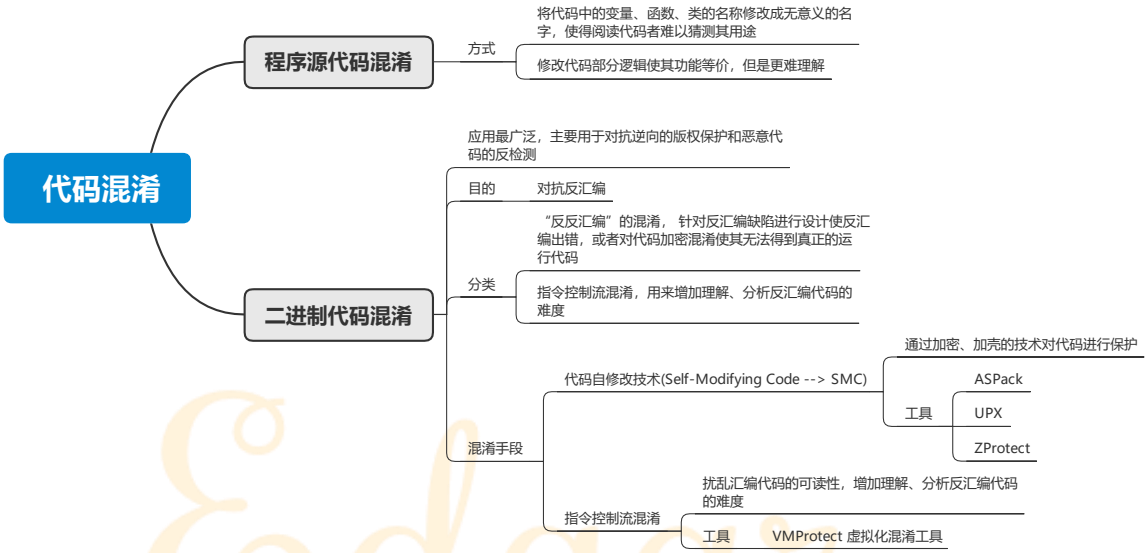
1. 确定反汇编的代码区域，即区分出程序的代码和数据段
2. 确定程序代码入口之后，读取该位置的二进制机器指令，执行表查找，将机器码的值和它对应的汇编语言助记符提取出来，然后根据指令状态机提取操作数
3. 获取指令并解码出所有操作数之后，需要对它的汇编语言等价进行格式化，输出反汇编代码
4. 完成第一条指令的反汇编之后，重复上述过程，继续反汇编下一条指令，直到反汇编完程序文件中的指令代码

典型算法：



2.2.3 代码混淆

代码混淆是一种将计算机程序代码转换成一种功能上等价，但是难以阅读和理解的变形



2.2.3 反调试

- 基于调试特征检测的反调试

  程序处于被调试状态时

  - PEB 中 **bingDebug** 标志会被置于非 0
  - PEB 结构中的 **NtGlobalFlags** 标志在 Windows 2000 及以后平台中会被设置成一个特定的值
  - 进程堆中也有一个标志 **Heap_ForceFlags** 可用于调试检测，通常情况下为 0

- 基于调试特征隐藏代码

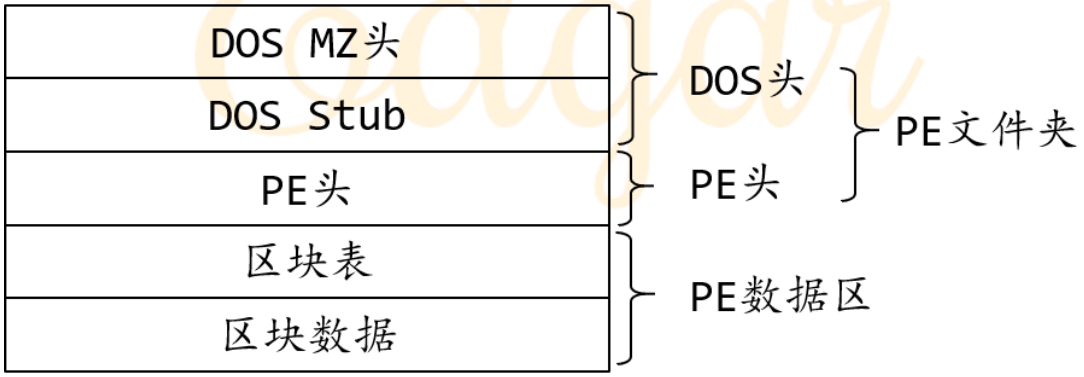
✧ 异常中断指令 int 3 常被用来设置软件断点，在程序代码里植入 int 3 指令是一个经典的反调试手段。 int 3 未调试时是异常，调试时是断点；**int 0x2d** 更有隐蔽性

2.3 Windows 操作系统基础

2.3.1 PE 文件结构

PE(Portable Executable)文件是微软 Windows 操作系统上的可执行文件，包括扩展名 **EXE、DLL、OCX、SYS、COM** 等

PE 文件结构划分为：

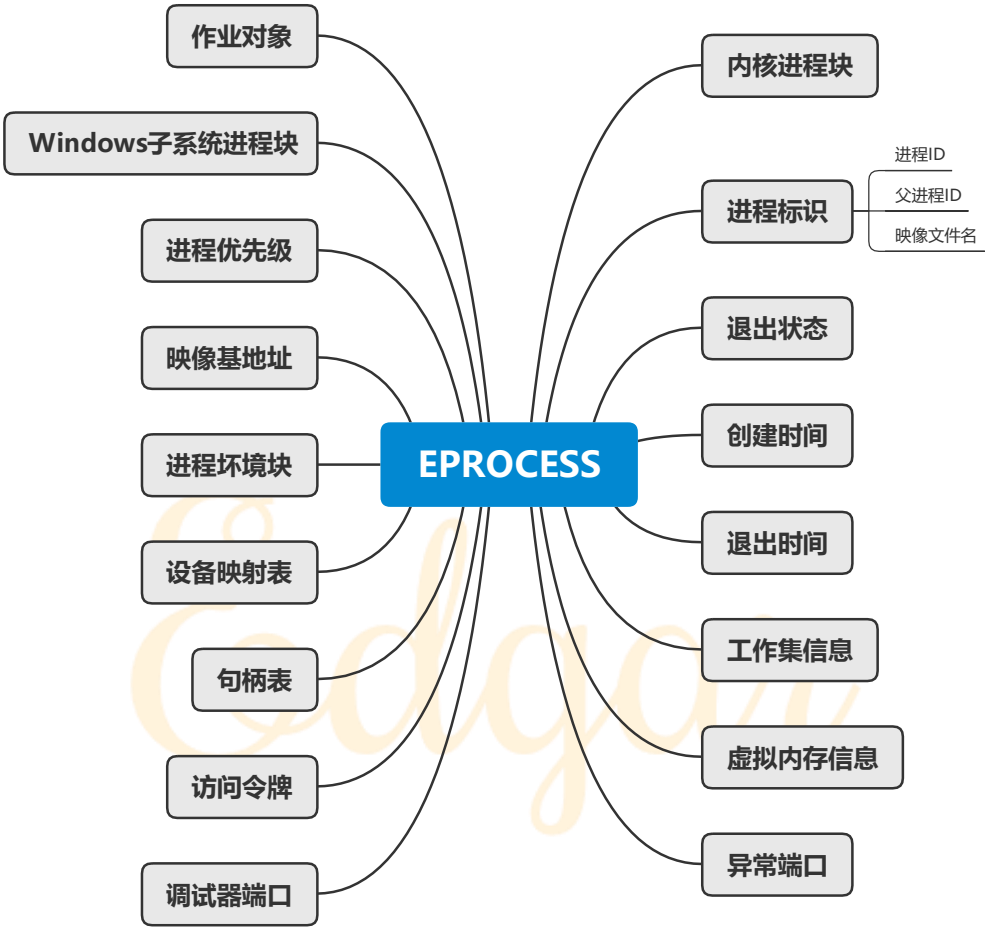


具体内容 P44-50

2.3.2 进程管理

进程是计算机中的程序关于特定数据集合上的一次运行活动，是系统进行**资源调度和分配的基本单位**。

Windows 进程由 EPROCESS 块表示，其中包含进程控制块(PCB)、进程坏境块(PEB)，其主要结构内容：

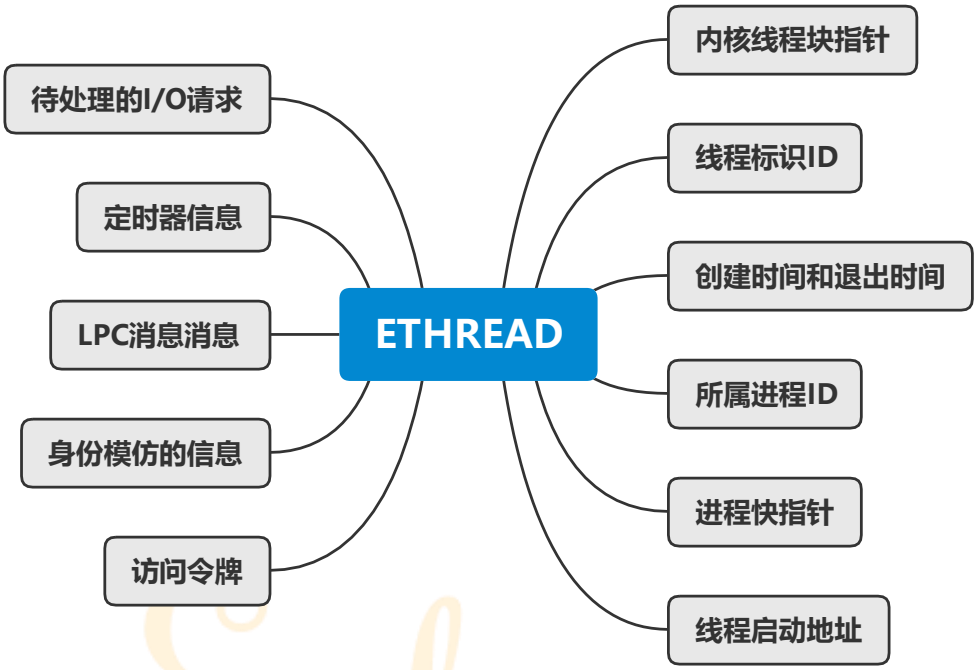


具体说明 P51

2.3.3 线程管理

线程是程序执行流的**最小单元**，是进程中的一个实体，是被操作系统**独立调度和分派的基本单位**。线程本身**不占用系统资源**，与同属于一个进程的其他线程共享进程所拥有的全部资源。

线程块是线程的基本结构，主要内容：

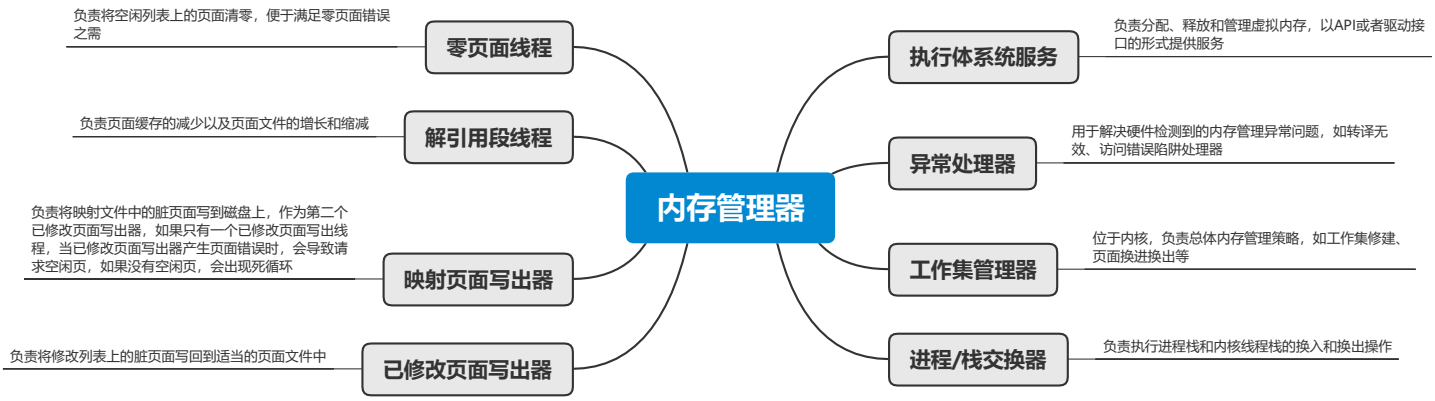


详细内容 P52-54

2.3.4 内存管理

Windows32 操作系统中，0x7fffff 下的地址默认为用户态内存地址，0x8000000 以上的地址默认为系统内核空间地址。

Windows 系统使用内存管理器对内存进行管理，主要负责：①将进程的虚拟地址空间转译到物理内存 ②在内存不足时将数据换页到磁盘



2.3.5 对象和句柄管理

Windows使用对象模型为执行题中实现的各种内部服务提供一致的、安全的访问途径，并设计了对象管理器负责**创建、删除、保护**和**跟踪**对象

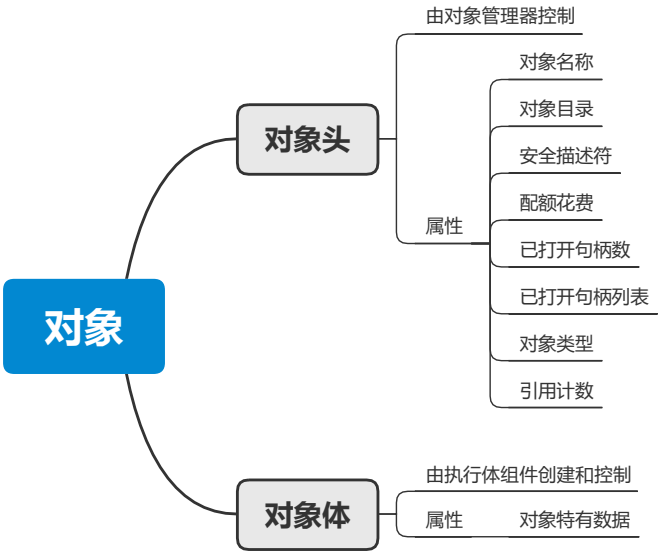
Windows 内部有两种类型对象：

✧ **执行体对象**：由执行组件(进程管理器、内存管理器、I/O 子系统)所实现的对象

✧ **内核对象**：由 Windows 内核实现的一组更为基本的对象，在用户模式下是不可见的，在执行体内部被创建和使用

执行体对象封装了一个或多个内核对象

对象组成：



当一个进程根据名称来创建或者打开对象时，系统返回一个句柄，进程需要根据该句柄对该对象进行访问和管理

Windows32 系统下的句柄表项由**指向对象的指针**和**访问掩码**组成

2.3.6 文件系统

文件系统是操作系统中对文件存储设备的空间进行组织和分配，负责文件存储并对存入的文件进行保护和检索的系统，包括为用户提供**创建文件，读入、修改、存储文件**等具体功能

Windows 操作系统包含对 CDFS、UDF、FAT12、FAT16、FAT32、NTFS 文件系统格式的支持，不同格式使用于不同的特定环境