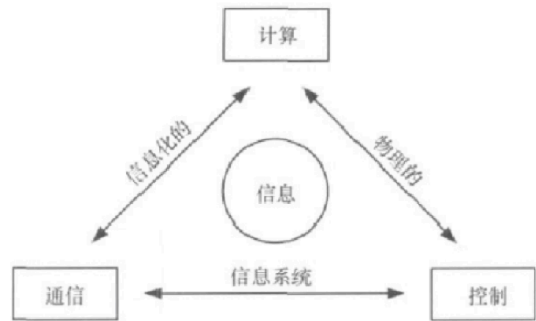


## 2013–2016年嵌入式考试题整理

### 名词解释类

#### 1.CPS: 概念, 典型结构, 典型应用 [15年] [考纲重点]

信息物理系统, 计算进程和物理进程的统一体, 集计算、通信、控制于一体的下一代智能系统



#### 2.传感器: 概念, 请写出常用的传感器 [15年]

通过物理接口将光、声音、温度等变化转化为具体数值的装置

常用的传感器有声音传感器、温度传感器、光传感器、压力传感器

#### 3.看门狗: 概念, 作用, 工作原理 [15年] [16年] [考纲重点]

是一个定时器(1分), 要求程序每隔一段时间向其输出一个信号。如果超过WDT的规定时间, 就给MCU发送一个复位信号, 使其复位(3分), 实现嵌入式系统在无人状态下的连续工作(2分)

#### 4.上下文切换: 概念, 步骤 [15年]

[百度百科] 是指CPU从一个进程或线程切换到另一个进程或线程

- (1)决定是否作上下文切换以及是否允许作上下文切换
- (2)保存当前执行进程的上下文
- (3)使用进程调度算法, 选择一处于就绪状态的进程
- (4)恢复或装配所选进程的上下文, 将CPU控制权交到所选进程手中

#### 5.有限状态机: 定义, 有哪两类状态机, 两者有何区别, 至少三种示例应用 [15年]

[百度百科]

定义: 表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型

分类:

(1)状态模型: 单线程, 只有一个状态可以在任何时候有效

① **Moore state models**: 输出信号仅与当前状态有关, 即可以把Moore型有限状态的输出看成是当前状态的函数

② **Mealy state models**: 输出信号不仅与当前状态有关, 而且还与所有的输入信号有关, 即可以把Mealy型有限状态机的输出看成是当前状态和所有输入信号的函数

(2)分层FSM支持并发和层次结构, 硬件设计上与mealy有相同的问题; 与软件不相关, 导致更紧凑的代码

示例: 电话, 定时炸弹, 自动贩卖机, 电梯等

#### 6.优先级翻转: 定义, 解决 [15年]

[百度百科] 当一个高优先级任务通过信号量机制访问共享资源时, 该信号量已被一低优先级任务占有, 因此造成高优先级任务被许多具有较低优先级任务阻塞, 实时性难以得到保证。

(1)优先级天花板是当任务申请某资源时，把该任务的优先级提升到可访问这个资源的所有任务中的最高优先级，这个优先级称为该资源的优先级天花板。这种方法简单易行，不必进行复杂的判断，不管任务是否阻塞了高优先级任务的运行，只要任务访问共享资源都会提升任务的优先级。

(2)优先级继承是当任务A申请共享资源S时，如果S正在被任务C使用，通过比较任务C与自身的优先级，如发现任务C的优先级小于自身的优先级，则将任务C的优先级提升到自身的优先级，任务C 释放资源S后，再恢复任务C的原优先级。这种方法只在占有资源的低优先级任务阻塞了高优先级任务时才动态的改变任务的优先级，如果过程较复杂，则需要判断。

## 7.仿真开发：定义，使用原因，特点 [15年]

[百度百科] 利用模型复现实际系统中发生的本质过程，并通过对系统模型的实验来研究存在的或设计中的系统

利用计算机实现对于系统的仿真研究不仅方便、灵活，而且也是经济的；

在研制、鉴定和定型全过程都必须全面地应用先进的仿真技术。否则，任何新型的、先进的飞行器和运载工具的研制都将是不可能的

## 8.嵌入式系统：定义，应用（至少五种），特点 [13年] [16年] [考纲重点]

定义：以应用为中心，以计算机技术为基础，软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗等具有严格要求的专用计算机系统

特点：形式多样，面向特定应用

适应于各种微处理器及微处理器架构

及其关注成本

对可靠性和实时性有极高的要求

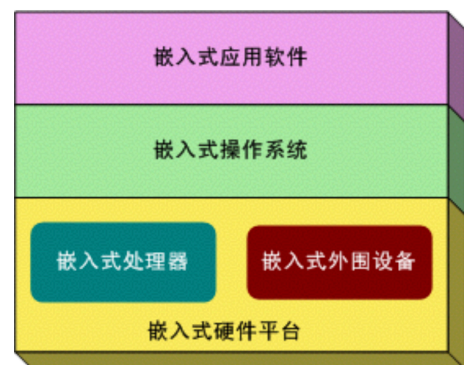
所用的操作系统具有实时性、可靠性、紧凑、可剪裁、

适应于各种微处理器及微处理器架构

的特点

使用特殊工具和特殊方法开发

典型应用：可穿戴设备，通信，工控，航天航空，智能家居，智能驾驶



## 9.物联网：概念，典型结构，典型应用 [16年] [考纲重点]

(1) 物联网就是万物互联的互联网，是互联网与嵌入式系统发展到高级阶段的融合

(2) 套嵌入式的定义，结构，应用

## 10.开源硬件：定义，理解 [16年]

开源硬件是可以通过公开渠道获得的硬件设计，任何人都可以对已有的设计进行学习、修改、发布、制作和销售。硬件设计的源代码的特定的格式可以为其他人获得，以方便对其进行修改。理想情况下，开源硬件使用随处可得的电子元件和材料，标准的过程，开放的基础架构，无限制的内容和开源的设计工具，以最大化个人利用硬件的便利性。开源硬件供人们在控制他们的技术自由的同时共享知识并鼓励硬件设计开放交流贸易

### 11.可重入性：定义，一个程序具备可重入性需要满足哪些特性 [16年]

- (1)一个函数可以同时被多次调用。可重入函数在任何时候都可以被中断，一段时间后继续运行时数据不会丢失
- (2)原子地使用所有共享变量；程序内不调用非可重入的函数；不以非原子方式使用硬件

### 12.建模：定义，举例建模技术 [16年]

建模是为了理解系统，对系统做出抽象、建立模型的过程

- ① 面向状态的模型：FSM
- ② 面向活动的模型：数据流图、过程模型
- ③ 面向结构的模型：系统框图
- ④ 面向数据的模型：ER图

## 简答类

### 1.嵌入式系统设计中，影响软硬件折衷方案的重要因素有哪些？ [15年]

嵌入式系统设计中软硬件划分需要考虑哪些因素？举例说明哪些功能通常由软件实现，哪些通常用硬件实现？ [13年]

今年不是重点

软硬件的双重性，速度，灵活性和开销，其中软硬件的双重性是考虑的前提

- 软件实现的部分：操作系统功能（任务调度、资源管理、设备驱动），协议栈（TCP/IP），应用软件框架
- 硬件实现的部分：基本系统，物理接口，基本逻辑电路
- 双重性部分：算法（加密 / 解密，编码 / 解码，压缩 / 解压），数学运算（浮点运算，FFT）

传统开发过程与软硬件协同设计

传统：需求分析，规格说明，体系结构，构件设计，系统集成

### 2.为什么嵌入式软件要交叉开发？比较宿主机和目标机的差异 [15年]

原因：为了满足代码编写编译调试与运行环境有不同要求的情况。嵌入式系统往往是面向特定应用的一体化软件，不具备软件编写编译功能，可以降低嵌入式系统复杂度，降低成本。

差异：宿主机是通用计算机系统而目标机是多位嵌入式系统。宿主机具有完整的开发环境如IDE、编译器、debugger等，用于软件的编写，生成的二进制可执行代码通过串口传输到专用的面向应用的目标机上，并采取交叉方式进行调试

过程：(1)宿主机上开发，模拟调试 (2)通过串口或网络传到目标机上  
(3)目标机上基于监视器和操作系统的调试 (4)目标机脱离宿主机运行

### 3.如何做到从NAND Flash启动？请描述不同的解决方法 [15年]

bsp, bootloader

#### 嵌入式系统的启动过程\* [考纲重点] [新重点]

- 1.上电复位，板级初始化
- 2.系统引导/升级
- 3.系统系统初始化
- 4.应用程序初始化
- 5.多任务调度运行

## 嵌入式系统的启动过程\* [考纲重点] [新重点]

- ① 上电复位，板级初始化
- ② 系统引导/升级
- ③ 系统初始化
- ④ 应用初始化
- ⑤ 多任务应用运行

### bsp

概念：全称“板级支持包”(Board Support Packages)，一段启动代码

特点：

- 硬件相关性：因为嵌入式实时系统的硬件环境具有相关性，所以，作为高层软件与硬件之间的接口，BSP必须为操作系统提供操作和控制具体硬件的方法。
- 操作系统相关性：不同的操作系统具有各自的软件层次结构，因此，不同的操作系统具有特定的硬件接口形式

### 与bios、EFI区别

- BIOS：BIOS主要是负责在电脑开启时检测、初始化系统设备（设置栈指针，中断分配，内存初始化）、装入操作系统并调度操作系统向硬件发出的指令。
- BSP是和操作系统绑在一起运行在主板上的，尽管BSP的开始部分和BIOS所做的工作类似，可是大部分和BIOS不同，作用也完全不同。程序员还可以编程修改BSP，在BSP中任意添加一些和系统无关的驱动或程序，甚至可以把上层开发的统统放到BSP中。而BIOS程序是用户不能更改，编译编程的，只能对参数进行修改设置。更不会包含一些基本的硬件驱动。
- EFI：由于EFI框架比BIOS要大得多，其启动过程也比BIOS要复杂。于BIOS最大的区别就是EFI首先需要EBC虚拟机，然后再启动设备驱动和EFI应用程序，最后通过EFI boot manager加载操作系统引导程序

### 引导模式（去年有一星，今年不是重点）

- 1.将OS加载到RAM，时间效率低，空间开销大，运行速度快，复杂
- 2.让OS在ROM或FLASH中运行，时间效率高，空间开销小，运行速度慢，简单

### bootloader及其启动过程（去年有一星，今年不是重点）

- 1.硬件初始化
- 2.设定操作系统的硬件环境参数
- 3.跳至操作系统的第一条指令位置
- 4.消亡

## 4.选择嵌入式微处理器需要考虑的因素有哪些？有几种类型？有什么区别？[15年]

答：体积、重量、可靠性、功耗、成本以及工作温度、抗电磁干扰等方面

### 四种类型

- ① MPU微处理单元：体积小，功耗少，成本低，可靠性，可提供工业级应用（32位以上的MCU）
- ② MCU微控制器：体积小，功耗少，成本低，可靠性，工业主流，主要硬件集成到一块芯片中（8位、16位）
- ③ DSP处理器：信号处理，用于数字滤波，频谱分析，FFT
- ④ SOC：体积小，功耗少，成本低，可靠性，产品系统最大包容的集成器件，设计生产效率高

## 5. RTOS的引导模式有哪些？试说明各自的特点 [15年]

- **需要** BootLoader的引导模式：节省空间，牺牲时间，适用于硬件成本低，运行速度快，但启动速度相对慢
- **不需要** BootLoader的引导模式：时间效率高，系统快速启动，直接在NOR flash或ROM系列非易失性存储介质中运行，但不满足运行速度的要求

## 6. 叙述RTOS的主要特点 [15年] [13年]

概念：接受刺激后在预定时间给出预订质量的响应的系统

特点：实时性、**可靠性**、**紧凑**、抢占式内核、**可扩展**、**可剪裁**、**可预测性**

选型原则：实时性、可靠性、紧凑、可扩展、可剪裁、可预测性、成本、功耗、对RAM和ROM的使用开销，工具链，支持模块

## 7. 给出嵌入式软件轮询模式的伪码描述？描述有何变种或者改进？使用时需要注意的事项有什么？具体应用举例说明 [15年]

轮询读：

```
while(true){  
    while(peek(in_status) == 0);  
    *loc = peek(in_char);  
    poke(in_status, 0);  
    loc++;  
}
```

轮询写：

```
while(true){  
    while(peek(out_status) != 0);  
    poke(out_char, *loc);  
    poke(out_status, 1);  
    loc++;  
}
```

变种和改进：采用flagI、flagO；结合终端使用，在服务例程中置位

使用注意事项：及时将寄存器状态位恢复，避免在前端应用程序中造成不可预计的错误

轮询占用大量的CPU时间，使其不能进行其他计算工作

具体应用：轮训模式的忙等IO；轮训模式的嵌入式软件结构

## 8. 描述RTOS ucOSII采取哪些策略与机制来确保任务的实时性 [15年] [14年] [16年] [考纲重点]

- ① 任务调度时总是选择就绪任务中优先级最高的一个运行
- ② 允许任务间的抢占
- ③ 具有优先级的中断机制
- ④ 有确定性的任务调度/切换时间和中断延迟
- ⑤ 精度高的时钟节拍产生时钟中断
- ⑥ 信号量、互斥量、信箱、消息队列等任务间通信机制

实时任务调度\*\*、中断与时钟\*、同步与通信、存储管理（静态、动态）\*\* [考纲重点] [新重点]

同步与互斥：第五章PPT，PIP、PCP算法 [考纲重点] [新重点]

多CPU时，时间调度是脆弱的，见第五章PPT最后部分，增加了处理器，缩短了任务时间，减少了前置条件，反而会导致整体运行时间上升

## 9.冯诺伊曼体系结构和哈佛体系结构的区别 [15年]

- 冯诺伊曼结构：数据和程序放在同一个存储单元，统一编址，指令和数据通过同一个总线访问
- 哈佛结构：程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个相互独立的存储器，每个存储器独立编制、独立访问。与之相对应的是系统中设置的两条总线（程序总线 and 数据总线），从而使数据的吞吐率提高了一倍（更大存储带宽和更大可预测带宽）
- 哈佛结构不能使用自修改代码

## 10.ARM7指令是几级流水线，各阶级执行是什么操作 [15年]

3级流水线：取指——译码——执行

ARM9：取指——译码——执行——数据缓冲——写回

ARM10：取指——发射——译码——执行——存储器——写回

## 11.嵌入式软件系统主要采用的软件体系结构及各自的优缺点（至少5种） [15年] [13年] [考纲重点]

轮询式，轮询中断式，中断式，状态机（switch-case），函数队列，RTOS

	Priority	Response time	Impact of changes	Shared data
<b>Round Robin</b>	None, everything runs in sequence	The sum of all tasks.	Significant. Changing the execution time of tasks or adding tasks impacts all other tasks.	no shared data problems
<b>State Machine</b>	Each state determines the priority of the next state	The sum of all tasks.	Significant. Changing the execution time of tasks or adding tasks impacts all other tasks.	no shared data problems
<b>Round Robin with Interrupts</b>	Interrupts get priority over main loop , Priority of interrupts as well	时间从这里分割 The sum of all tasks or Interrupt execution time	改变带来的影响从这里分割 Less significant for interrupt service routines. Same as Round Robin as main loop	共享数据从这里分割 must deal with data shared with interrupt service routines
<b>Just Interrupts</b>	Interrupts get priority	Interrupt execution time	Little significant for interrupt service routines	Must deal with data shared with interrupt service routines
<b>Function Queue Scheduling</b>	Interrupts have priority.Tasks execute in sequence	Execution time of the longest task	Low. Interrupts manage priority functions. Queue manages lower priority.	must deal with data shared with interrupt service routines



## 12.详细描述嵌入式软件的运行过程 [13年]

## 13.嵌入式微处理器与通用处理器之间的区别 [13年]

## 14.影响流水线性能的因素有哪些？分支指令对流水线性能有何影响？如何解决？ [13年][16年]

影响流水线性能的因素：

结构阻滞：多条指令进入流水线后，硬件资源满足不了指令重叠执行

数据阻滞：指令在流水线中重叠执行时，后继指令需要用到前面指令的执行结果

控制阻滞：流水线遇到分支指令和其他改变 PC 寄存器值的指令

分支指令的影响：遇到分支指令时，由于处理器要执行跳转操作，因此已经在流水线上的顺序指令就是无效的，需要被清空。处理器重新形成流水线需要时间，因此影响效率。

解决方法：应用分支预测技术

## 15.ARM处理器有哪七种运行模式？ [13年]

- ① 用户模式(usr)：ARM处理器正常的程序执行状态
- ② 系统模式(sys)：运行具有特权的操作系统任务
- ③ 快中断模式(fiq)：支持高速数据传输或通道处理
- ④ 管理模式(svc)：操作系统保护模式
- ⑤ 数据访问终止模式(abt)：用于虚拟存储器及存储器保护
- ⑥ 中断模式(irq)：用于通用的中断处理
- ⑦ 未定义指令终止模式(und)：支持硬件协处理器的软件仿真

## 16.比较忙等I/O与中断驱动I/O [13年]

忙等IO：用指令来检查设备是否就绪

- 最简单的设备驱动方式
- CPU不能在检查设备的过程中执行其他指令
- 很难处理同时发生的IO操作

中断IO：基于子程序调用，使下一条指令为一个子程序调用的预定位置，返回位置被保存确保执行前台程序

- CPU和设备之间通过总线连接
- CPU和设备之间进行握手
- 设备发出中断请求
- 当CPU能处理中断时发出中断确认请求

两种机制确保中断更加有效：优先级决定哪个中断先获得CPU；中断向量决定每个中断对应的执行代码

### 17. NorFlash和NANDFlash之间的区别（至少五种） [13年]

#### 从NorFlash启动和NANDFlash启动有什么不同 [16年]

- ① 性能不同(2分): NOR 的读速度比 NAND 稍快, NAND 的写速度比 NOR 快很多
- ② 接口不同(2分): NOR 的地址线与数据线分开, 可以直接读写; NAND 的地址线与数据线在一起, 读写时需要程序控制
- ③ 造成的启动方式不同(2分): NOR 可以 XIP, NAND 需要 bootloader 将程序复制或映射到内存上再运行

其他:

NOR, NAND

存储代码, 存储文件

随机存取, 顺序存取

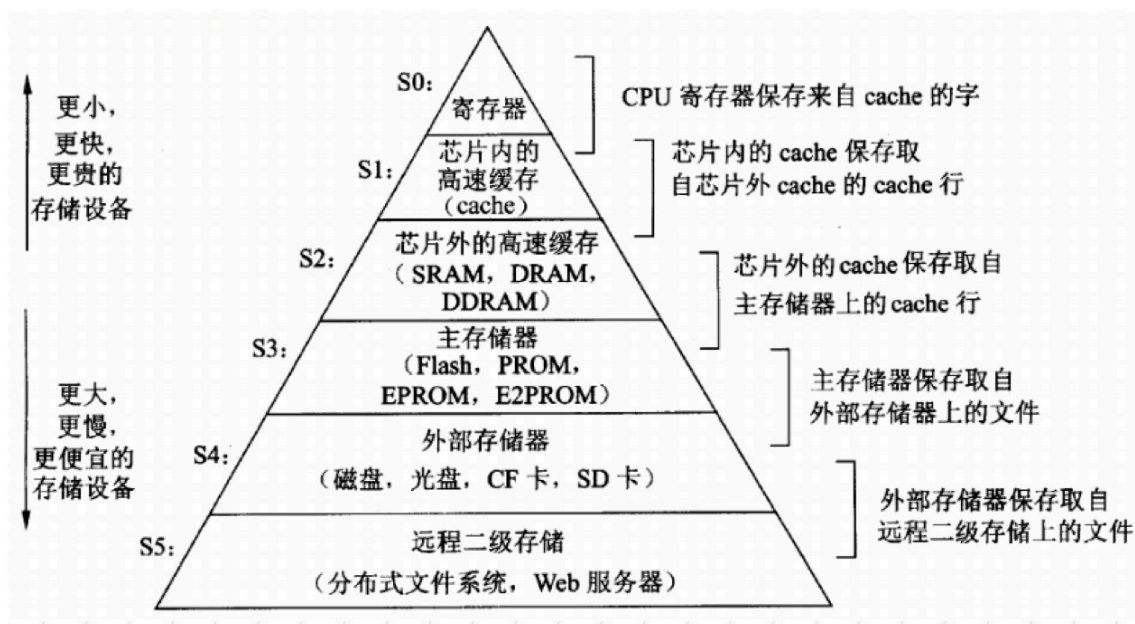
不容易坏, 容易坏

单位价格高, 单位价格低

待命时省电, 运行时省电

### 18. 实时操作系统ucOS-II如何从64个任务扩展到支持256个任务 [13年]

### 19. 画出嵌入式系统存储结构的金字塔图 [14年]



寄存器——芯片内高速缓存 (cache)——芯片外高速缓存 (SRAM, DRAM)——主存储器 (ROM, FLASH)——外部存储器 (磁盘、磁带)——远程二级存储器 (云盘, 服务器)

ROM的种类与选型: PROM, EPROM, ROM, EEPROM

RAM的种类与选型: SRAM, DRAM, SDRAM

Cache: 存储数据的, 存储程序的, 统一的, 分级cache



## 嵌入式系统静态和动态内存管理\*\* (2017年新) [考纲重点][新重点]

存储管理：固定分区的存储管理方法

OSMemCreate, OSMemGet, OSMemPut

put时必须放入get时去除的内存块

Heap1: 不支持free, 简单的数组分配, 适用于确定性任务

Heap2: 简单的free, 会导致碎片, 适用于每次申请和释放内存大小相同的人物

Heap3: 调用了malloc和free, 通过挂起调度器达到了线程安全性

## 20.画出嵌入式系统双总线结构 [14年] [考纲重点][新重点]

- ① 单总线结构：使用一条单一的系统总线来链接CPU、主存和I/O设备。总线只能分时工作, 使信息传送的吞吐量受到限制。四次握手 (1enq, 2ack, 2low\_ack, 1low\_enq)
- ② 双总线结构：在CPU和主存之间专门设置了一组高速的存储总线, 使CPU可通过专用总线与存储器交换信息, 并减轻了系统总线的负担。中间用桥连接。主存仍可通过系统总线与外设之间实现DMA操作, 而不必经过CPU。
- ③ 多总线结构：在双总线系统的基础上增加I/O总线, 其中系统总线是CPU、主存和通道 (IOP)之间进行数据传送的公共通路, 而I/O是多个外部设备与通道之间进行数据传送的公共通路。通道实际上是一台具有特殊功能的处理器, 它分担了一部分CPU的功能, 以实现对外设的统一管理及外设与主存之间的数据传送。

## 21.嵌入式系统设计面临的挑战 [16年]

多少硬件? 如何保证系统可靠工作? 实时性如何达到? 如何降低成本? 如何降低功耗? 如何保证系统易升级? 今年不是重点

## 22.中断设计方法 [16年] [考纲重点]

- ① 先设计中断再编写ISR
- ② 保持ISR简短
- ③ ISR执行后立即开中断
- ④ 用空指针填充未被使用的中断向量

## 23.中断处理过程, 中断机制相关的执行时间开销 [16年] [考纲重点]

中断处理过程:

- ① 中断请求
- ② 中断排队或优先级仲裁
- ③ 中断处理
- ④ 中断返回

中断机制的执行时间开销:

- ① 中断延迟
- ② 寄存器的现场保护
- ③ 进入终端服务
- ④ 中断恢复

时钟：一种特殊的中断，频率在10–100HZ之间，需要专门的定时器或50/60Hz的交流电源信号，调用中断程序OSTimeTick，将每个delay的进程时间–1，如果为0，则进入就绪队列

24.轮询的不足？前后台体系结构在轮询的基础上的改变？运行方式？适用于哪些嵌入式应用？ [16年]

不足：效率低，大量 CPU 时间消耗在等待设备响应上

前后台：引入了中断机制

运行方式：后台的一组程序通过轮询方式访问CPU，前台在触发中断后，将请求插入到后台轮询环中的某个位置完成处理

适用场合：单用户交互与实时 I/O 设备控制

RMS和EDF策略调度 [考纲重点]

有限状态机模型设计题 [考纲重点]

1.分别给出基于FSM和层次型有限状态机的设计

自动售卖机（多种支付方式：硬币、纸币、支付宝、微信）

体现软硬件协同设计；功能在软硬件上的合理划分

2.上述需求中没有周全的考虑某些细节，请补充并给出修改后的设计

3.实现有几种方式，选择哪种实现？说明原因

考虑嵌入式系统中的功能、可靠性、成本、体积、功耗等多方面因素

实现方式：(1)switch-case (2)状态表 (3)状态模式

选择状态模式，因为更易于维护和扩展，降低了实际状态之间的耦合，通过抽象接口来实现，当需要修改或添加状态时，只需修改内部实现，不用改变接口。隐藏内部实现一定程度上分解了复杂度，使代码易于理解。