

高等教育自学考试

名词解释题总结

课程：**13015 计算机系统原理**

讲 师 小飞学长 Pro

微 信 公 众 号 小飞学长 Pro

全网统一账号 小飞学长 Pro

(B 站/抖音/快手/小红书)

主 讲 专 业 计算机科学与技术

完成时间 2025 年 3 月 23 日

第一章 计算机系统概述

中央处理器 (CPU)：中央处理器是计算机系统的核心部件，负责执行程序中的指令、处理数据，并控制计算机的各个部件协调工作。CPU 包含算术逻辑单元(ALU)和控制单元，执行算术运算、逻辑运算、数据传送和输入/输出操作的控制。**[2024年4月]**

算术逻辑部件 (ALU)：算术逻辑部件是 CPU 中的一个关键组件，负责执行算术运算（如加、减、乘、除）和逻辑运算（如与、或、非、异或）。ALU 接收来自控制单元的操作码，并根据这些操作码执行相应的运算。

通用寄存器：通用寄存器是 CPU 内部的高速存储单元，用于暂存指令、数据和地址。它们可以被 CPU 快速访问，从而加速数据处理速度。

程序计数器 (PC)：程序计数器是一个特殊的寄存器，用于存储下一条将要执行的指令的地址。当 CPU 执行一条指令时，程序计数器自动加 1（或按照跳转指令更改），以指向下一条要执行的指令。

指令寄存器 (IR)：指令寄存器用于暂存当前正在执行的指令。当 CPU 从主存储器中读取一条指令后，会将其放入指令寄存器中，解码并执行。

控制器：控制器是 CPU 中的另一个关键组件，负责从主存储器中取出指令，对指令进行译码，并根据指令的要求，有序地、协调地指挥和控制计算机各部件自动工作。

主存储器：主存储器是计算机中的主要存储部件，用于存储程序和数据。CPU 可以直接访问主存储器中的数据，而不需要通过其他外部设备。

总线：总线是计算机各部件之间传输信息的公共通道，由导线组成，用于在 CPU、主存储器、I/O 设备之间传输数据和控制信号。

主存地址寄存器 (MAR)：主存地址寄存器用于存放欲访问的主存单元的地址，其地址由 CPU 中的程序计数器给出。MAR 与主存储器地址译码电路相连，为 CPU 访问主存提供所要访问的主存单元的地址。

主存数据寄存器 (MDR)：主存数据寄存器用于暂存从主存储器读出的一条指令或一个数据字，或准备写入主存储器的字。MDR 与主存储器数据总线相连，与 MAR 配合使用，实现 CPU 与主存储器之间的数据交换。

机器指令：机器指令是 CPU 能直接识别并执行的指令，由操作码和操作数地址码组成。每一条指令完成一个基本的微操作，如数据传送、算术运算等。

指令操作码：指令操作码是机器指令中的一部分，用于指定该指令所要执行的操作类型，如加法、减法、存储等。

高级程序设计语言：高级程序设计语言是一种接近于人类自然语言和数学语言的程序设计语言，易于编写和理解，经过编译或解释后能转换成机器语言执行。

汇编语言：汇编语言是一种用助记符表示的指令系统中的指令，是一种低级程序设计语言。汇编语言需要汇编程序翻译成机器语言才能执行。

机器语言：机器语言是直接由二进制代码指令表达的计算机语言，是计算机唯一能直接识别和执行的语言。

机器级语言：机器级语言通常指的是与特定机器或处理器直接相关的低级编程语言，包括机器语言和汇编语言。

源程序：源程序是用高级程序设计语言编写的、尚未编译或解释的程序。

目标程序：目标程序是源程序经编译或解释后得到的机器语言程序，可以直接被计算机执行。

编译程序：编译程序是一种将高级程序设计语言编写的源程序翻译成目标程序的软件工具。编译过程通常包括词法分析、语法分析、语义分析、优化和代码生成等阶段。

解释程序：解释程序是一种直接执行高级程序设计语言编写的源程序的软件工具。解释程序对源程序进行逐句扫描和解释，并立即执行相应的操作，不生成独立的目标程序。

汇编程序：汇编程序是一种将汇编语言源程序翻译成机器语言目标程序的软件工具。汇编过程包括将汇编语言指令转换为机器语言指令，并分配内存地址等。

语言处理系统：语言处理系统是指用于处理各种程序设计语言的软件工具集合，包括编译器、解释器、汇编器等。

设备控制器：设备控制器是计算机中的一个硬件部件，负责控制和管理外部设备（如打印机、磁盘驱动器等）与 CPU 之间的数据传输。

最终用户：最终用户是指直接使用计算机系统和应用程序的个人或组织，他们通过用户界面与计算机进行交互，完成各种任务。

系统管理员：系统管理员是负责管理和维护计算机系统的专业人员，他们负责安装、配置、监控和维护计算机硬件、软件和网络等。

应用程序员：应用程序员是负责开发、编写和维护各种应用程序的专业人员，他们使用高级程序设计语言进行程序设计，以满足用户的特定需求。

系统程序员：系统程序员是负责设计、开发和维护计算机系统软件的专业人员，他们负责编写操作系统、编译器、数据库管理系统等核心软件。

指令集体系结构（ISA）：指令集体系结构定义了计算机指令的格式、数量和功能，以及 CPU 如何执行这些指令

微体系结构：微体系结构是计算机处理器设计的一个关键部分，它定义了处理器内部各个组件的结构、功能以及它们之间的交互方式。微体系结构决定了处理器如何执行指令、管理数据、处理中断等，从而影响了处理器的性能、功耗和效率。

透明：在计算机科学中，透明通常指的是某个方面对使用者是不可见的或者是不需要考虑的。例如，内存管理的某些细节对程序员可能是透明的，意味着程序员不需要直接处理这些细节，而由操作系统或硬件自动管理。

响应时间：响应时间是指系统或组件从接收到输入或请求到产生相应输出或响应

所需的时间。在计算机系统中，响应时间通常用于衡量用户界面、网络请求或数据库查询等的速度。

吞吐率：吞吐率是指系统或组件在单位时间内处理输入或输出数据的数量或速度。它衡量了系统处理数据的整体效率，通常以每秒处理的事务数、每秒传输的字节数等来度量。

用户 CPU 时间：用户 CPU 时间是指 CPU 在执行用户程序时所花费的时间。这是区别于系统 CPU 时间的一个概念，系统 CPU 时间指的是 CPU 在执行操作系统内核任务时所花费的时间。

时钟周期：时钟周期是计算机处理器中的一个基本时间单位，它表示处理器执行一个基本操作所需的时间。每个时钟周期，处理器都会执行一系列预定义的指令或操作。

主频：主频是指处理器每秒钟的时钟周期数，通常以赫兹（Hz）或兆赫兹（MHz）、吉赫兹（GHz）为单位表示。主频越高，处理器执行指令的速度通常越快。[2024 年 10 月]

CPI：CPI 是指处理器执行一条指令所需的平均时钟周期数。它衡量了处理器的指令执行效率，CPI 越低，说明处理器执行指令的速度越快。

基准程序：基准程序是一组用于测试和评估计算机系统性能的预定义程序和测试套件。通过运行这些基准程序，可以比较不同系统的性能，并找出性能瓶颈。

MIPS：MIPS 是一个衡量处理器性能的指标，表示处理器每秒钟能执行的百万条指令数。MIPS 越高，说明处理器的性能越好。

峰值 MIPS：峰值 MIPS 是指处理器在理想条件下能够达到的最高 MIPS 值。这通常是在处理器没有遇到任何瓶颈（如内存延迟、I/O 限制等）时测得的性能上限。

相对 MIPS：相对 MIPS 是用于比较不同处理器或系统性能的 MIPS 指标。它通常是将一个系统的 MIPS 值除以另一个基准系统的 MIPS 值，从而得到一个相对性能的比值。

MFLOPS：每秒百万次浮点运算。

GFLOPS：每秒十亿次浮点运算。

TFLOPS：每秒万亿次浮点运算。

PFLOPS：每秒千万亿次浮点运算。

EFLOPS：每秒百亿亿次浮点运算。

第二章 数据的表示和运算

真值：真值是数学上的值，即我们实际想要表示的数，不考虑计算机内部如何表示。

机器数：机器数是在计算机内部用来表示数值的二进制编码。它可能由于编码方式（如原码、补码等）和机器的特性而有所不同。

数值数据：数值数据是指那些可以表示数量或大小的数据，如整数、浮点数等。

非数值数据：非数值数据是指不能表示为数量或大小的数据，如字符、文本、图片等。

无符号整数：无符号整数是一个非负的二进制数，它只表示大小，不表示正负。

带符号整数：带符号整数可以表示正数或负数。在计算机中，通常使用最高位作为符号位，其余位表示数值大小。

定点数：定点数是一种数值表示方法，其中小数点位置是固定的。定点数主要用于表示整数或小数。

浮点数：浮点数是一种数值表示方法，其中小数点位置可以浮动。它通常由尾数（mantissa）和阶码（exponent）两部分组成，用于表示非常大或非常小的数。

尾数：尾数是浮点数中用于表示数值大小的部分。

阶和阶码：阶是浮点数中小数点实际位置与默认位置之间的偏移量。阶码则是这个偏移量的编码表示。

溢出：溢出是指计算机在运算过程中，结果超出了它能表示的范围。这可能导致不正确的结果或错误。

规格化数：规格化数是指满足特定条件的浮点数，例如尾数的最高位是 1（对于二进制数）。这种表示方式可以提高浮点数的精度和范围。

左规：左规是在浮点数运算中，当尾数向左移动时，阶码相应减小的过程。这通常发生在尾数溢出时。

右规：右规是在浮点数运算中，当尾数向右移动时，阶码相应增加的过程。这通常发生在尾数下溢时。

ASCII 码：ASCII（American Standard Code for Information Interchange）码是一种用于表示英文字符和其他控制字符的编码系统。

汉字输入码：汉字输入码是用户通过键盘等设备输入汉字时所使用的编码，如拼音、五笔等。

汉字内码：汉字内码是计算机内部用于表示汉字的编码，如 GB2312、GBK、UTF-8 等。

字长：字长是指计算机内部处理数据的位数，例如 32 位或 64 位处理器。它决定了计算机一次能处理的数据量。

大端方式：大端方式（Big Endian）是一种数据存储方式，其中高位字节存储在内存的低地址端，低位字节存储在内存的高地址端。

小端方式: 小端方式 (Little Endian) 是另一种数据存储方式, 其中低位字节存储在内存的低地址端, 高位字节存储在内存的高地址端。

ALU: ALU 是计算机中的一个重要组件, 负责执行算术运算 (如加、减、乘、除) 和逻辑运算 (如与、或、非)。

第三章 程序的转换及机器级表示

机器语言程序：一种由二进制代码组成的程序，直接由计算机硬件执行的指令集合。

汇编指令：一种由助记符表示的机器指令，与机器语言一一对应，可被汇编器转换成对应的机器语言指令。

汇编语言程序：使用汇编语言编写的程序，以助记符形式表示的指令序列，需要经过汇编器转换为机器语言程序。

汇编助记符：在汇编语言中使用的符号，代表对应的机器指令。

汇编程序：将汇编语言程序转换成机器语言程序的程序，也称为汇编器。

反汇编程序：将机器语言程序转换回汇编语言程序的程序。

机器级代码：直接在计算机硬件上执行的指令，通常是由汇编器将汇编语言转换而来的。

CISC：复杂指令集计算机，指一种计算机架构，其指令集包含了多种复杂的指令。

RISC：精简指令集计算机，指一种计算机架构，其指令集相对简单，每条指令的执行时间相对较短。

通用寄存器：用于存储临时数据和地址的寄存器，通常用于存储计算过程中的中间结果。

变址寄存器：用于存储地址计算中的偏移量的寄存器，通常用于实现数组、结构体等数据结构的访问。

基址寄存器：用于存储内存地址基址的寄存器，通常与变址寄存器一起使用，用于计算实际内存地址。

栈指针寄存器：用于存储栈顶地址的寄存器，用于实现函数调用和局部变量存储等操作。

指令指针寄存器：用于存储当前正在执行指令的地址的寄存器，通常指向下一条要执行的指令。

标志寄存器：用于存储程序运行状态标志的寄存器，例如零标志、进位标志等。

条件标志(条件码)：标志寄存器中的一部分，用于表示上一条指令的执行结果，通常用于控制条件跳转指令的执行。

寻址方式：指定操作数的寻址方式，例如立即寻址、寄存器寻址、相对寻址等。

立即寻址：直接将操作数的值嵌入到指令中。

寄存器寻址：使用寄存器存储操作数的地址。

相对寻址：使用相对于某个基址的偏移量来寻址。

存储器操作数：指令中的操作数直接引用存储器中的数据。

实地址模式：CPU 直接使用物理地址访问内存的模式。

保护模式：一种 CPU 工作模式，操作系统可在其中对内存和其他资源进行保护和管理。

有效地址：执行指令时计算出的操作数在内存中的真实地址。

比例变址：根据一个基址寄存器和一个比例系数来计算操作数的地址。

非比例变址：仅根据一个基址寄存器来计算操作数的地址。

比例系数(比例因子)：用于比例变址计算的一个数值。

MMX 指令：Intel 处理器中用于多媒体处理的指令集。

SSE 指令集：Intel 处理器中的一种扩展指令集，用于实现单指令多数据(SIMD)操作。

SIMD：单指令多数据，一种并行计算的技术，能够同时处理多个数据。

多媒体扩展通用寄存器：用于 SIMD 指令集操作的寄存器。

栈(Stack)：一种后进先出(LIFO)的数据结构，用于临时存储函数调用的上下文和局部变量。

调用者保存寄存器：在函数调用过程中，由调用者负责保存和恢复的寄存器。

被调用者保存寄存器：在函数调用过程中，由被调用者负责保存和恢复的寄存器。

帧指针寄存器：用于指向当前函数调用的栈帧的寄存器。

当前栈帧：当前函数调用的栈帧，存储了函数的参数、局部变量和返回地址等信息。

按值传递参数：将参数的值传递给函数。

按地址传递参数：将参数的地址传递给函数，函数可以直接访问参数的值。

嵌套调用：在一个函数内部调用另一个函数的过程。

递归调用：函数直接或间接地调用自身的过程。

第四章 可执行文件的生成和加载执行

链接: 在计算机编程中, 链接是指将多个源文件或目标文件中的代码和数据组合成一个可执行文件或共享库文件的过程。

可重定位文件: 包含了程序的代码和数据, 但是还没有完成地址的最终分配, 可以在链接时进行地址的重定位, 得到可执行文件。

可执行文件: 经过链接之后, 完成了地址的最终分配, 可以直接在操作系统上执行的文件。

符号解析: 在链接过程中, 将符号 (如函数名、变量名) 与其在程序中的地址或者其他符号相关联的过程。

重定位: 将程序中的符号引用与其真正的地址或者其他符号关联起来的过程。

[2024 年 10 月]

ELF 头: Executable and Linkable Format 头, 是一种用于描述可执行文件和共享库文件结构的数据结构。

节头表: 包含了可执行文件或共享库文件中各个节 (section) 的信息, 如起始地址、大小等。

程序头表: 包含了可执行文件或共享库文件中各个程序段 (segment) 的信息, 如加载地址、大小等。

只读代码段: 存放程序的代码, 只允许读取操作, 不允许写入操作。

可读/写数据段: 存放程序的数据, 允许读取和写入操作。

全局符号: 在整个程序中都可见的符号, 可以被其他文件引用。

外部符号: 在当前文件中未定义, 但在其他文件中定义的符号。

本地符号: 仅在当前文件中可见的符号, 对其他文件不可见。

强符号: 在链接过程中, 优先选择的符号。

COMMON 符号: 未被初始化的全局变量或未定义的外部变量的符号。

静态链接: 在编译时将所有的库文件链接到可执行文件中的链接方式。

共享库文件: 包含了可执行文件所需的代码和数据, 可以在多个进程间共享使用的文件。

动态链接: 在程序运行时将共享库文件链接到内存中的链接方式。

动态链接器: 负责在程序运行时将共享库文件链接到内存中的系统程序。

动态链接库: 包含了可执行文件所需的代码和数据, 可以在程序运行时被动态链接到内存中使用的库文件。

进程: 计算机中正在运行的程序的实例, 是操作系统进行资源分配和调度的基本单位。 **[2024 年 4 月]**

进程描述符: 操作系统中用于描述进程的数据结构。

命令行解释程序: 接收用户输入的命令行, 并将其解释为操作系统能够执行的命令的程序。

指令周期： CPU 执行一条指令所经历的时间周期。

指令译码器： 负责将指令解码成对应的操作控制信号的硬件模块。

内部异常： 由程序执行过程中的错误或不正常情况引起的异常。

外部中断： 由外部设备（如硬件中断、时钟中断）引起的中断。

数据通路： CPU 中负责数据传输和处理的部件的集合。

执行部件： CPU 中负责执行指令的部件，包括算术逻辑单元 (ALU) 等。

功能部件： CPU 中用于实现特定功能的部件，如乘法器、除法等。

控制器： CPU 负责控制整个运算过程的部件。

时钟信号： CPU 内部的时钟信号，用于同步各个部件的工作。

指令流水线： CPU 中用于提高指令执行速度的一种技术，将指令执行过程划分为多个阶段并行处理。

第五章 程序的存储访问

随机存取存储器 (RAM): RAM (Random Access Memory) 是一种临时存储设备, 用于存储计算机当前正在处理的数据和程序。它可以快速地读取和写入数据, 但一旦电源关闭, 其中的信息就会丢失。

只读存储器 (ROM): ROM (Read-Only Memory) 是一种数据存储设备, 一旦数据被写入, 就不能被修改或删除。它通常用于存储计算机的基本启动指令或固件。

易失性存储器: 易失性存储器是指当电源关闭后, 数据会丢失的存储设备。RAM 就是一种易失性存储器。

记忆单元 (Cell): 记忆单元是存储设备中最基本的存储单位, 用于存储一个二进制位 (bit) 的信息。

存储阵列 (Bank): 在内存设计中, 存储阵列 (Bank) 指的是一组物理上相互独立的内存模块, 每个模块都有自己的地址和数据总线。这允许处理器并行访问多个 Bank, 从而提高性能。

编址单位: 编址单位指的是在内存或其他存储设备中用于标识和访问单个数据元素的最小单位。

编址方式: 编址方式是指如何给存储设备中的每个单元分配地址的方法。不同的编址方式会影响存储器的访问速度和效率。

存储周期: 存储周期是指存储器进行连续两次独立的存储器操作 (如读或写) 之间所需的最小时间间隔。

静态 RAM (SRAM): SRAM (Static Random Access Memory) 是一种 RAM 类型, 它的存储单元使用双稳态电路来存储数据, 不需要刷新, 因此速度较快但功耗较高。

动态 RAM (DRAM): DRAM (Dynamic Random Access Memory) 是另一种 RAM 类型, 它的存储单元使用电容来存储数据, 由于电容会漏电, 所以需要定期刷新来保持数据。

闪存 (Flash 存储器): 闪存 (Flash Memory) 是一种非易失性存储设备, 它使用浮栅晶体管来存储数据。闪存常用于 USB 驱动器、SSD 和某些类型的移动设备中。

SDRAM: SDRAM (Synchronous Dynamic Random Access Memory) 是一种同步动态 RAM, 其内部命令的发送与接收采用时钟同步方式。它广泛应用于现代计算机系统中。

行地址选通信号 (RAS): 在 DRAM 中, 行地址选通信号 (Row Address Strobe) 用于指定要访问的存储单元的行地址。

列地址选通信号 (CAS): 列地址选通信号 (Column Address Strobe) 在 DRAM 中用于指定要访问的存储单元的列地址。

磁盘驱动器: 磁盘驱动器是用于读取和写入磁盘 (如硬盘) 数据的设备。

磁盘控制器: 磁盘控制器是管理和控制磁盘驱动器操作的硬件或软件组件。

寻道时间：寻道时间是指磁盘驱动器将读写头移动到指定磁道所需的时间。

旋转(等待)时间：旋转时间(或等待时间)是指从磁盘驱动器开始寻找数据到数据所在扇区旋转到读写头下方所需的时间。

数据传输率：数据传输率是指存储设备在单位时间内可以传输的数据量。

平均存取时间：平均存取时间是指从发出数据访问请求到数据实际可用所需的平均时间。

固态硬盘(SSD)：固态硬盘是一种使用闪存技术存储数据的存储设备，它比传统的机械硬盘更快、更耐用。

时间局部性：时间局部性是指如果一个信息项被访问，那么在未来不久它很可能再次被访问。

空间局部性：空间局部性是指如果程序访问了某个存储单元，那么在不久的将来，其附近的存储单元也很可能被访问。

主存块：在缓存和主存之间，主存块是主存中可以被作为一个单元整体地传输到缓存中的一组数据。

cache 行(槽)：在计算机缓存中，cache 行(或槽)是缓存中用于存储数据的基本单元。

命中率：命中率是指缓存中访问成功的次数与总访问次数的比例。

命中时间：命中时间是指缓存命中时，从发出数据请求到数据实际可用所需的时间。

缺失损失：缺失损失是指当缓存未命中时，从慢速存储器(如主存)获取数据所需的额外时间和开销。

直接映射：在计算机存储体系中，直接映射是一种缓存映射方式。

在这种方式下，每个主存页面直接映射到缓存中的特定页面。这种方法简单但可能导致缓存冲突，降低缓存效率。

全相联映射：全相联映射是一种缓存映射方式，在这种方式下，主存中的任一页面可以映射到缓存中的任一页面位置，没有固定的对应关系。

组相联映射：组相联映射是介于直接映射和全相联映射之间的一种缓存映射方式。它将主存页面分成若干组，每组内的页面映射到缓存中的固定位置，但组内各页面可以映射到缓存中的不同位置。

替换策略：替换策略决定了当缓存满时，如何选择数据项进行替换。不同的替换策略会影响缓存的效率和性能。

FIFO 算法：FIFO 算法是一种缓存置换算法，也被称为先进先出算法。当缓存满时，最早进入缓存的数据项会被首先移出，以腾出空间给新进入的数据项。

LRU 算法：LRU 算法是一种常用的缓存置换算法。

该算法认为最近最少使用的数据项在未来被访问的可能性最小，因此当缓存满时，会选择最久未使用的数据项进行替换。

LRU 位：LRU 位通常用于跟踪缓存中数据项的使用情况，以便实现 LRU 算法。

在某些实现中，每次数据项被访问时，其 LRU 位都会被更新，以便在需要替换时能够找到最久未使用的数据项。

cache 写策略：Cache 写策略决定了 CPU 对 Cache 中的数据进行修改时，如何更新主存中的数据。常见的写策略包括写回法和写通法。

回写法：在缓存系统中，回写法是一种数据更新策略。当 CPU 修改缓存中的数据时，修改并不会立即写回到主存中，而是等到缓存中的数据块被替换或显式地执行写回操作时，才将修改后的数据写回主存。

通写法：通写法是另一种缓存写策略。与回写法不同，当 CPU 修改缓存中的数据时，修改会立即写回到主存中，确保主存中的数据始终是最新的。

虚拟地址（逻辑地址）：虚拟地址是用户程序中的地址表示方式，它并不直接对应物理内存中的实际地址。在程序执行过程中，虚拟地址需要通过地址映射或地址重定位转换为物理地址。

虚拟页号：虚拟页号是虚拟地址中的一部分，用于标识虚拟内存中的页面。

物理地址：物理地址是内存中各存储单元的编号，即存储单元的真实地址，它是可识别、可寻址并实际存在的。它是 CPU 访问内存时使用的地址。

页框（页帧）：页框是物理内存中的一个固定大小的块，用于存储从虚拟内存中加载的数据。

物理页号：物理页号是物理地址中的一部分，用于标识物理内存中的页面。

未分配页：未分配页是尚未被进程申请使用的虚拟内存页面，它不占用物理内存或磁盘空间。

未缓存页：未缓存页是指在虚拟内存中，但尚未被加载到物理内存中的页面。这些页面在需要时会被加载到物理内存中。

请求分页：请求分页是一种内存管理技术，它允许程序在需要从磁盘或其他辅助存储中加载页面到物理内存中。

页故障：页故障是指在程序执行过程中，当试图访问一个尚未加载到物理内存中的虚拟页面时发生的情况。此时，系统需要执行页面替换和加载操作来解决故障。

页表：页表是操作系统用于记录虚拟内存页面与物理内存页面之间映射关系的数据结构。通过页表，CPU 可以将虚拟地址转换为物理地址。

页表基址寄存器：页表基址寄存器用于存放当前运行程序的页表的起始地址。当 CPU 需要访问某个虚拟地址时，会与页表基址寄存器中的值拼接，从而找到对应的页表项。

有效位（装入位）：有效位通常用于标识缓存中的页面是否有效或已被加载。当页面在缓存中时，其有效位会被设置为 1；当页面不在缓存中或已被替换时，有效位会被设置为 0。

修改位（脏位）：修改位用于标识缓存中的页面是否被修改过。如果页面被修改过，其修改位会被设置为 1；否则，修改位为 0。这个信息在决定是否需要将页面写回到主存时非常重要。

访问权限：访问权限定义了程序或用户对特定内存区域的访问级别，如可读、可写或可执行。

快表（TLB）：快表是一种缓存机制，用于存储虚拟地址到物理地址的转换表的一部分。当 CPU 需要解析虚拟地址时，首先会查找快表，如果找到对应的转换条目，则可以直接进行地址转换，从而加速地址解析过程。

特权指令：特权指令是只能由操作系统或具有特权级别的程序执行的指令。这些指令通常用于执行系统级操作，如访问硬件、修改系统状态等。

特权模式：特权模式是操作系统的一种运行模式，在这种模式下，程序可以执行特权指令并直接访问系统资源。

内核态：内核态是操作系统内核的运行模式，这种模式具有最高的权限级别，可以执行所有指令并访问所有内存区域。

用户态：用户态是操作系统的一种运行模式，在这种模式下，程序运行在非特权级别，不能执行特权指令，也不能直接访问系统资源。

存储保护：存储保护是操作系统提供的一种机制，用于防止程序访问或修改其不应该访问的内存区域，从而确保系统的稳定性和安全性。

地址越界：地址越界是指程序尝试访问其权限范围之外的内存地址。这通常会导致程序崩溃或系统错误。

访问越权：访问越权是指程序或用户尝试执行超出其权限范围的操作，如访问受保护的内存区域或执行特权指令。

第六章 程序中 I/O 操作的实现

I/O 硬件：指的是输入/输出设备，如键盘、鼠标、显示器、磁盘等，这些硬件用于与计算机进行数据传输。

I/O 软件：是一组程序和工具，用于管理和控制 I/O 硬件的操作，包括设备驱动程序、中断处理程序、缓冲管理以及相关的系统调用等。

用户空间 I/O 软件：用户空间 I/O 软件是运行在操作系统用户空间中的程序，用于处理输入/输出 (I/O) 操作。这些软件通常通过系统调用来与内核空间 I/O 软件进行交互，从而实现对硬件设备的访问和控制。

内核空间 I/O 软件：内核空间 I/O 软件是操作系统内核中负责处理 I/O 操作的部分。它管理硬件设备的驱动程序，处理中断，以及提供系统调用接口供用户空间程序使用。

系统调用处理程序：是操作系统内核中的一部分，负责处理来自用户空间的系统调用请求，实现用户程序与内核之间的交互。

系统调用服务例程：是系统调用处理程序调用的内核函数，用于实现具体的系统调用功能。

设备驱动程序：设备驱动程序是一种软件，用于控制计算机上的硬件设备（如打印机、键盘、鼠标、音频/视频设备等），以便它们能够与计算机系统协调工作。设备驱动程序通常由设备制造商提供，并与操作系统紧密集成，以便系统可以识别和使用设备。

中断服务程序：中断服务程序是操作系统内核中用于处理特定中断事件的程序。

系统级 I/O 函数：是操作系统提供的用于执行 I/O 操作的 API(应用程序接口)，如 C 语言中的 `stdio` 函数库。

虚拟文件系统：是一种用于网络环境的分布式文件系统接口，它允许不同的文件系统与操作系统进行交互，而无需考虑具体的文件系统实现细节。

文件描述符：文件描述符是由无符号整数表示的句柄，进程使用它来标识打开的文件。文件描述符与包括相关信息（如文件的打开模式、文件的位置类型、文件的初始类型等）的文件对象相关联。

文件元数据：文件元数据是关于文件自身的信息，如文件的创建时间、修改时间、大小、权限等。这些信息通常存储在文件系统的元数据结构中，用于管理和访问文件。

流缓冲区：用于提高 I/O 操作的效率和性能，通常用于在数据读取或写入时进行暂存，以减少系统调用的次数和内存访问的延迟。

索引节点：通常用于文件系统中，包含文件的元数据（如文件大小、权限、创建时间等），是文件系统用来定位和访问文件的关键数据结构。

目录文件：目录文件是文件系统中的特殊文件，用于存储多个文件目录的信息。它包含了多个文件或目录的元数据，使得文件系统能够方便地管理和查找这

些文件或目录。

目录项: 目录项是文件系统中用于记录文件或目录信息的条目。它包含了文件或目录的名称、属性以及其他相关元数据,使得用户可以方便地查找和访问这些文件或目录。

系统打开文件表: 是操作系统内核维护的一个数据结构,用于记录当前打开的文件及其相关信息,如文件描述符、文件状态等。

磁盘高速缓存: 是操作系统用于提高磁盘 I/O 性能的一种技术,通过缓存最近访问过的磁盘数据块,减少对磁盘的物理访问次数。

高速缓存 RAM: 高速缓存 RAM (Random Access Memory) 是一种用于暂时存储数据和程序的内存设备。它具有快速的读写能力,并通常用于存储 CPU 经常访问的数据,以提高数据访问速度。

I/O 控制方式: I/O 控制方式是指计算机系统中用于管理 I/O 操作的方法和技术。常见的 I/O 控制方式包括程序直接控制、中断驱动、DMA 等。

程序直接控制 I/O: 指用户程序直接通过特定的 I/O 指令或系统调用进行 I/O 操作,而不依赖于操作系统提供的 I/O 软件。

就绪状态: 在多任务操作系统中,通常用于描述一个进程或线程已经准备好执行,等待被调度器分配到 CPU 上运行的状态。

中断控制 I/O: 中断控制 I/O 是一种 I/O 控制方式,其中当外部设备准备好进行数据传输时,它会向 CPU 发送一个中断信号。CPU 在接收到中断信号后,会暂停当前的任务,转而执行中断处理程序来处理 I/O 操作。

中断屏蔽字: 中断屏蔽字是一种用于控制中断响应的寄存器或变量。通过设置中断屏蔽字,可以屏蔽或允许特定的中断请求,以实现中断的优先级管理。

多重中断: 指的是在一个中断处理程序执行过程中,发生了另一个中断,需要暂时保存当前中断的上下文并处理新中断的情况。

中断嵌套: 与多重中断类似,但更强调中断之间的层次关系,即一个中断可以在另一个中断的处理过程中被触发和处理。

DMA 方式: DMA (Direct Memory Access) 方式是一种允许某些硬件子系统(如磁盘控制器)直接访问主存,而无需通过 CPU 进行数据传输的技术。它提高了数据传输的效率,减少了 CPU 的介入。

DMA 控制器: DMA 控制器是一种硬件设备,用于实现 DMA 方式的数据传输。它负责在内存和设备之间直接传输数据,而无需 CPU 的干预。

设备控制器: 是硬件设备与计算机之间的接口电路,负责控制和管理硬件设备的操作,如磁盘控制器、显卡控制器等。

I/O 端口: 是 CPU 与外部设备通信的接口,用于发送和接收数据以及控制信号。

控制端口: 控制端口是计算机系统中用于发送和接收控制信号的端口。它通常用于与硬件设备通信,以控制设备的操作和行为。

数据端口: 数据端口是计算机系统中用于传输数据的端口。它通常用于与硬件设

备通信，以发送和接收数据。

状态端口：是 I/O 端口的一种，用于读取或设置硬件设备的状态信息。

I/O 地址空间：是计算机内存中用于映射 I/O 端口和设备控制器的地址范围，使得 CPU 可以通过内存访问的方式与 I/O 设备进行通信。

独立编址方式：独立编址方式是指 I/O 端口地址和内存地址相互独立，使用不同的地址空间进行编址。这种编址方式使得 I/O 端口和内存可以互不干扰地进行访问。当中断发生时，CPU 会跳转到相应的中断服务程序执行，以处理中断事件。

统一编址方式：统一编址方式是指将 I/O 端口地址和内存地址统一进行编址，使用相同的地址空间。这种编址方式简化了 I/O 操作的访问方式，但也可能导致内存空间的浪费。

存储器映射 I/O：是一种 I/O 技术，通过将 I/O 端口或设备控制器映射到内存地址空间，使得 CPU 可以像访问内存一样访问这些 I/O 设备。

I/O 指令：是 CPU 执行的指令，用于启动 I/O 操作，如读取或写入数据。

可编程中断控制器：可编程中断控制器是一种用于管理中断请求的硬件设备。它允许操作系统对中断进行优先级排序、屏蔽和响应控制，以实现中断的高效处理。

中断请求寄存器：中断请求寄存器是用于记录当前待处理的中断请求的寄存器。当中断发生时，相应的中断请求会被记录到中断请求寄存器中，以供 CPU 或中断控制器处理。

中断响应优先级：是操作系统用来决定哪个中断应该首先被响应的优先级规则。

中断处理优先级：是操作系统用来决定哪个中断处理程序应该首先被执行的优先级规则。