
🌟 操作系统复习总结 (PPT 2-13)

2-intro-hardware (系统与硬件基础)

◇ 重点知识

- **硬件组成**: CPU、内存、I/O设备、系统总线结构。
- **CPU 模式切换**: 用户态 (User Mode) 与内核态 (Kernel Mode) 。
- **中断与异常 (Interrupt & Exception)**:
 - 硬件中断: I/O 事件。
 - 软件异常: 系统调用 (trap) 。
- **操作系统职责**: 进程管理、内存管理、文件系统、设备驱动。

🧩 可能题型

- 选择题:
 - “内核态与用户态的主要区别是什么?”
 - “哪个属于软件中断?”
- 大题:
 - “简述中断处理的基本过程。”

3-intro-syscall (系统调用与程序执行流程)

◇ 重点知识

- **系统调用 (System Call)**: 应用程序与内核的接口。
 - 常考: `fork()`, `execve()`, `wait()`, `open()`, `read()`, `write()`, `exit()`
- **系统调用机制**: 通过 **trap 指令** 进入内核态。
- **程序到进程的生命周期**:
源代码 → 编译 → 链接 → 加载 (`ld`, `exec`)
- **ELF 文件结构**: `.text`, `.data`, `.bss`, `.rodata`, `.symtab`。
- **内存保护机制**: 基址寄存器(Base) 与限长寄存器(Limit)。

🧩 可能题型

- 选择题:
 - “哪个函数会触发系统调用?”
 - “ELF 文件的 `.bss` 段保存什么内容?”
 - 大题:
 - “描述从源码到可执行文件的全过程。”
 - “系统调用如何实现用户态到内核态的切换?”
-

4-process-intro (进程概念)

◇ 重点知识

- **进程定义**：程序的运行实例。
- **PCB (Process Control Block)**：存储进程状态、寄存器、资源等。
- **进程状态转换图**：新建→就绪→运行→阻塞→终止。
- **上下文切换 (Context Switch)**：保存与恢复 CPU 状态。
- **多道程序设计 (Multiprogramming)**：提高 CPU 利用率。

🧩 可能题型

- 选择题：
 - “进程与程序的区别？”
 - “上下文切换会保存哪些信息？”
- 大题：
 - “画出并解释进程状态转换图。”

5-process-control (进程控制与管理)

◇ 重点知识

- **进程创建**：`fork()`，父子进程空间独立。
- **加载新程序**：`execve()`。
- **进程终止**：`exit()`、`_exit()`、`abort()`。
- **等待子进程**：`wait()`、`waitpid()`。
- **僵尸 (Zombie) 与孤儿 (Orphan) 进程**。
- **优先级调整**：`nice()`，`renice()`。

🧩 可能题型

- 选择题：
 - “fork() 的返回值在父进程和子进程中分别是什么？”
- 大题：
 - “写出 fork-exec-wait 的执行流程图并说明各阶段作用。”
 - “解释孤儿与僵尸进程形成原因及其解决办法。”

6-process-signal (信号机制)

◇ 重点知识

- **信号 (Signal)**：软件中断，用于异步事件通知。
- **常见信号**：
 - `SIGKILL`，`SIGSTOP`（不可捕获/忽略）
 - `SIGINT` (Ctrl+C), `SIGTSTP` (Ctrl+Z), `SIGCHLD`
- **信号处理函数**：`sigaction()` 优于旧的 `signal()`。
- **阻塞与挂起**：`sigprocmask()`，`sigpending()`。
- **信号生命周期**：生成→挂起→传递→处理。

🔗 可能题型

- 选择题：
 - “SIGKILL 是否能被捕获？”
 - “哪个系统调用用于安装信号处理函数？”
- 大题：
 - “详细说明信号的产生、阻塞与传递过程。”

7-process-msg-shm (进程间通信 IPC)

◇ 重点知识

- 两大模型：
 - 共享内存 (Shared Memory) : `mmap()` 建立共享区。
 - 消息传递 (Message Passing) : 通过消息队列通信。
- POSIX 共享内存调用：
`shm_open()`, `ftruncate()`, `mmap()`, `munmap()`, `shm_unlink()`。
- POSIX 消息队列调用：
`ftok()`, `msgget()`, `msgsnd()`, `msgrcv()`, `msgctl()`。
- 生产者-消费者问题：典型同步模型。
- `mmap()` 参数含义 与内存映射图 (p13-14) 非常重要。

🔗 可能题型

- 选择题：
 - “共享内存和消息队列的区别？”
 - “哪个系统调用用于创建共享内存对象？”
- 大题：
 - “用 `shm_open/mmap` 实现进程通信的基本流程。”
 - “解释生产者-消费者问题中的同步机制。”

8-process-pipe (管道与无名通信)

◇ 重点知识

- 管道类型：
 - 匿名管道 (`pipe()`): 仅限父子进程。
 - 命名管道 (`mkfifo()`): 跨进程通信。
- 特点：
 - 半双工 (单向通信)。
 - 先进先出队列结构。
- 典型用法: `ls | grep | wc`。

🔗 可能题型

- 选择题：
 - “pipe() 能否在无关进程间通信？”
- 大题：
 - “比较管道、消息队列与共享内存三种IPC机制。”

9-process-distributedsys (分布式系统基础)

◇ 重点知识

- **分布式系统通信**：基于消息传递。
- **透明性 (Transparency)**：
 - 访问透明、位置透明、并发透明、故障透明。
- **消息模型**：
 - 方向（单向/半双工/全双工）
 - 连接模型（面向连接/无连接）
 - 可靠性（消息丢失、损坏、乱序）

🔗 可能题型

- 选择题：
 - “哪个属性体现了用户不感知节点位置？”
- 大题：
 - “解释分布式系统中消息传递的可靠性问题。”

10-process-socket (套接字通信)

◇ 重点知识

- **Socket 编程接口**：
`socket()`, `bind()`, `listen()`, `accept()`, `connect()`, `send()`, `recv()`。
- **客户端-服务器模型**。
- **TCP vs UDP 对比**。
- **端口号与 IP 地址绑定机制**。

🔗 可能题型

- 选择题：
 - “socket() 返回的是什么类型的对象？”
 - “TCP 与 UDP 的主要区别？”
- 大题：
 - “描述基于 socket 的通信流程。”

11-process-rpc-scheduling (RPC与CPU调度概念)

◇ 重点知识

- **RPC (远程过程调用) :**
 - Stub、Marshalling、Unmarshalling。
 - 调用过程像本地函数但实质跨网络。
- **CPU 调度基础指标:**
 - 周转时间 (Turnaround)、响应时间 (Response)、等待时间 (Waiting)、吞吐量、CPU 利用率。

🔗 可能题型

- 选择题:
 - “RPC 的 marshalling 作用是什么?”
- 大题:
 - “解释 CPU 调度性能指标及其相互关系。”

12-process-scheduling2 (CPU调度算法 I)

◇ 重点知识

- **FCFS:** 先来先服务, 非抢占。
 - 优点: 简单。
 - 缺点: 短任务被长任务阻塞。
- **RR (时间片轮转) :**
 - 每个任务分配时间片 Q。
 - 过短导致高切换开销, 过长退化为 FCFS。
- **SJF (最短作业优先) :**
 - 理论上最优, 但需预测 CPU burst。
 - 抢占式版本: SRTF。
- **饥饿与预测机制:** 指数移动平均估计下一个 burst 长度。

🔗 可能题型

- 选择题:
 - “RR 时间片过短会导致什么问题?”
 - “SJF 为什么可能导致长任务饥饿?”
- 大题:
 - “比较 FCFS、RR、SJF 三种算法的平均等待时间。”

13-process-scheduling3 (CPU调度算法 II)

◇ 重点知识

- **Priority Scheduling:** SJF 是其特例; 低优先级可能饥饿。
 - 解决方案: Aging (优先级随等待时间提升)。
- **Multilevel Queue (MLQ):** 不同类型进程放入不同队列。
- **Multilevel Feedback Queue (MLFQ):**

- 各级队列使用 RR，不同队列时间片不同。
 - 任务可根据表现升/降优先级。
- **Lottery Scheduling**: 通过彩票票数分配 CPU 比例。
- **SMP 调度**:
 - 对称（每核自调度）vs 非对称（单核负责）。
- **实时调度 (Real-Time Scheduling)**:
 - Soft vs Hard 实时。
 - **Rate-Monotonic (RM)**: 周期越短优先级越高。
 - **Earliest Deadline First (EDF)**: 最早截止期优先。
 - **Least Laxity First (LLF)**: 剩余空闲时间最少优先。

可能题型

- 选择题:
 - “MLFQ 中任务何时提升优先级？”
 - “EDF 与 RM 调度的区别？”
- 大题:
 - “比较 Priority, MLFQ, Lottery 调度的公平性与开销。”
 - “解释 EDF 与 LLF 的区别及适用场景。”