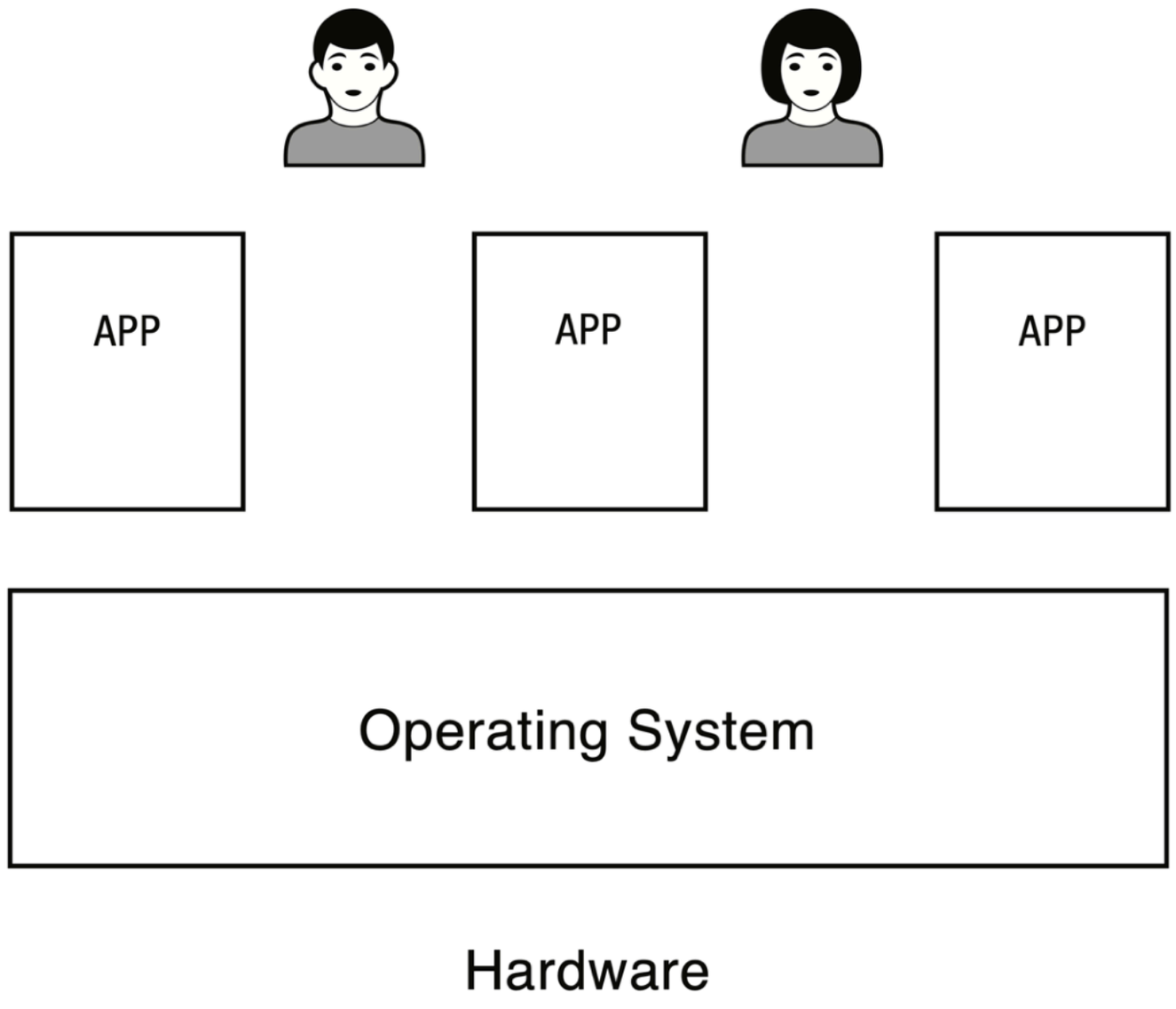
课程总结

2026

操作系统

操作系统是位于计算机用户和硬件资源中间的一层软件系统，其目的在于对硬件进行管理和抽象，并为应用提供服务



应用程序因而能在更受限(防止损害)、更强大(克服硬件限制)和更有用(提供通用服务)的环境中运行 *(easy to use)*

为应用提供与设备无关的服务 *(device-independent)*

管理各种硬件设备 *(device-specific)*

操作系统

- 虚拟化 (virtualisation)
  - 将 CPU 和物理内存转换为一个更加通用、具有更强能力、且更加易用的虚拟资源形式
- 并发 (concurrency)
  - 以正确的方式同时处理多件事情
  - 在没有原子性和顺序性保证时如何写一个正确的多线程程序
- 持久化 (persistency)
  - 以方便、高效和可靠的方式存储和访问信息
  - 对各种 I/O 设备进行管理

抽象

创建各种对象，并为其提供简洁易用的接口 (system calls)

- 进程 (process) 和线程 (thread)
  - 进程/线程控制块
  - 进程的创建和结束: `fork()`, `execve()`, `wait()` 和 `exit()` 等
  - 重定向 (redirection) 和管道 (pipe) 的实现
- 地址空间 (address space)
  - 若干连续内存段的集合
  - 空间的分配和回收: `malloc()` 和 `mmap()` 等

抽象

创建各种对象，并为其提供简洁易用的接口 (system calls)

- 文件 (file) 和目录 (directory)
  - FAT 和 Unix 文件系统的文件和目录结构
  - 文件的打开和读写: `open()`, `read()`, `write()` 和 `link()` 等
  - 文件描述符 (file descriptor) 和打开文件表 (open file table)

抽象

使用户程序运行在抽象的资源之上

- CPU 是虚拟的
  - 形成每个进程 “独占” 一个 CPU 的幻象
  - 其实是多个进程分时共享有限个 CPU
- 内存也是虚拟的
  - 形成每个进程 “独占” 整个物理内存的幻象
  - 其实每个进程仅占据物理内存的一部分 (甚至没有全部载入)
- 对 “任意字符序列” (文件或设备) 的访问
  - 按文件名/文件描述符进行读写

保护

在管理资源时还需提供保护以确保多个进程/线程之间没有冲突

- 对 CPU 的管理
  - 特权指令 (privilege instructions)
  - 上下文切换 (context switch)
- 对物理内存的管理 (对文件的管理也类似)
  - 地址转换 (address translation)
  - 读/写/执行等权限 (access rights)

保护

多线程并发编程的一个关键部分也是实现保护

- 互斥 (mutual exclusion): 避免多个线程对共享变量的无序任意访问
  - 锁 (locks): `lock()` 和 `unlock()`
- 同步 (synchronization): 确保多个线程协同推进
  - 条件变量 (condition variables): `wait()` 和 `signal()`
  - 信号量 (semaphores): `P()` 和 `V()`
- 正确实现并发是困难的
  - 借助硬件指令的帮助
  - 违反原子性和顺序性、以及死锁 (deadlock)

性能

一个奇慢无比的操作系统是没人使用的

- 进程的受限直接执行 (limited direct execution)
- 对局部性的利用: 各类缓存层 (cache and buffer, TLB)
- 对物理介质特性的利用: 快速文件系统 (fast file system)
- 尽可能推迟资源的分配 (lazy)
  - 写时复制 (copy-on-write)
  - 请求调页 (demand paging)

性能

一个奇慢无比的操作系统是没人使用的

- 不同策略的设计
  - 进程调度 (scheduling): 优化周转时间、响应时间、公平性等
  - 页面替换 (page replacement): 优化缺页错误率
- 考虑真实负载下的 Fast Path 和 Slow Path 权衡
  - 两阶段锁: spin 还是 block
  - inode 的索引结构: array 还是 tree

期末考试

📅 考试时间：2026-06-22 8:00-10:00 (星期一)

🏫 考试地点：仙 II-122

- 考试题目包括 6 个简答题和 4 个应用题
  - 题目类型和期中考试、以及课件 Quiz 类似
  - 后半学期内容大概占 60%
- 考察知识点
  - 基本概念和基本方法
  - 是什么 / 为什么 / 怎么做

总评成绩

📄 **实验 30% + 期中 20% + 期末 50%**

- 实验成绩：两次 Lab 的平均分
  - 每次 Lab 得分为 OJ 返回分数 (95%) + 人工评分 (5%)
  - 所有 Labs 的最终提交截止时间为 **2026-07-10 23:59**
  - 提交部分完成的代码也有一定的分数
- 平时分：
  - 随堂 QQ 群作业签到 (仅限在限定时间内提交)
  - 计入实验和期中成绩部分，上限 100 分

考试范围

Section I 操作系统概述

- 操作系统的定义和视角
- 操作系统的核心概念
  - 虚拟化、并发和持久化
  - 为应用程序提供服务 (系统调用)
  - 一个中断驱动的 C 程序

考试范围

Section 2 进程和线程

- 进程
  - 相关操作: `fork()`,`wait()`,`execve()`,`exit()`
  - 重定向 (redirection) 和管道 (pipe) 的实现
  - 进程的状态和生命周期
  - 进程上下文切换 (context switch) 的基本原理
- 线程
  - 引入线程的动机、线程的基本模型
  - 线程 user-level 和 kernel 实现的区别

考试范围

Section 3 内存管理

- 地址空间 (address space)、虚拟地址和物理地址
- 内部和外部碎片
- 分页 (paging)
    - 共享和写时复制 (Copy-on-Write)
    - 多级页表 (multi-level page table)
- 虚拟内存和请求调页 (demand paging)

考试范围

Section 4 互斥和同步

- 多线程并发程序中的违反原子性和顺序性问题
- 锁：`lock()`和 `unlock()`
  - Spin Lock 可能存在的问题
  - 两阶段锁 (Two-Phase Locking) 的基本思想
- 条件变量：`wait()`和 `signal()`
- 信号量：`P()`和 `V()`
- 使用 "锁 + 条件变量" 或 "信号量" 解决典型的同步问题
  - 生产者/消费者问题 (Producer/Consumer Problem)

考试范围

Section 4 互斥和同步

- 死锁
  - 哲学家就餐问题 (Dining Philosophers Problem)
  - 死锁的必要条件

考试范围

Section 5 文件系统

- 文件 (File) 和目录 (Directory)
- 文件系统的布局、以及文件和目录的实现
  - FAT 文件系统：File Allocation Table + 文件属性存储在目录项
  - Unix 文件系统：inode Structure + 文件属性存储在 inode
- 文件大小和文件占用的磁盘空间

考试范围

Section 5 文件系统

- Unix 文件系统
  - 文件描述符 (File Descriptor) 和打开文件表 (Open File Table)
  - 硬链接 (Hard Link) 和符号链接 (Symbolic Link)
  - 文件不同操作的实现 (涉及的磁盘访问情况)
    - `open()` 操作涉及的 resolve file name