1. 名词解释

1. 地址映射

将逻辑地址（如虚拟地址）转换为物理地址的过程，通常由MMU（内存管理单元）实现。

2. 动态重定位

程序运行时通过基址寄存器调整内存地址的技术，支持程序在物理内存中灵活加载。

3. 虚拟存储器

通过分页/分段技术，将磁盘扩展为“虚拟内存”，使程序可使用比物理内存更大的地址空间。

4. 静态链接

编译时将全部库代码合并到可执行文件中，生成独立程序（如.a静态库）。

5. 对换（Swapping）

将整个进程从内存移到磁盘交换区，以腾出内存空间的技术。

6. 设备驱动程序

操作系统内核中控制特定硬件的模块（如显卡驱动），提供标准接口抽象硬件细节。

7. SPOOLing

通过磁盘缓冲实现I/O设备的假脱机操作（如打印队列），提高CPU与设备并行性。

8. I/O通道

专用处理器（如IBM大型机的通道子系统），独立管理复杂I/O操作，减轻CPU负担。

9. 文件系统

操作系统用于管理磁盘文件的软件机制，包括存储、检索、权限控制等（如NTFS、ext4）。

10. 目标文件

编译器生成的中间文件（如.o），包含机器码和符号表，需链接生成可执行文件。

11. 文件的逻辑结构

用户视角的文件组织形式（如文本文件的字符流、数据库表的记录结构）。

12. 有结构文件

由固定格式记录组成的文件（如数据库文件），区别于无结构的字节流文件。

13. 位示图

用二进制位表示磁盘块或内存页使用状态的数据结构（1=已占用，0=空闲）。

14. 程序接口

供应用程序调用的编程接口（API），如库函数（fopen()）或系统调用（read()）。

15. 系统调用

操作系统内核提供的服务接口（如fork()），用户程序通过软中断触发特权操作。

16. I/O中断

设备完成I/O操作后向CPU发送的信号，触发中断处理程序（如键盘输入）。

17. 文件管理系统

操作系统中负责文件创建、读写、删除等功能的子系统（如Linux的VFS）。

18. 文件

存储在外部设备上的命名数据集合（如文档、程序），是操作系统管理的基本单元。

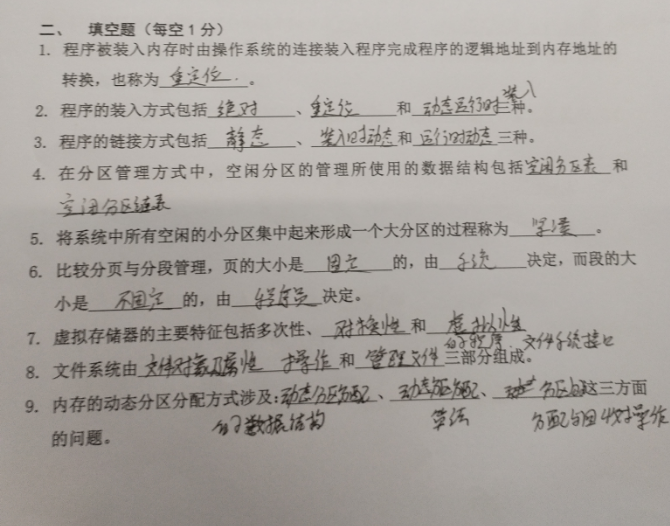
19. 文件的逻辑结构

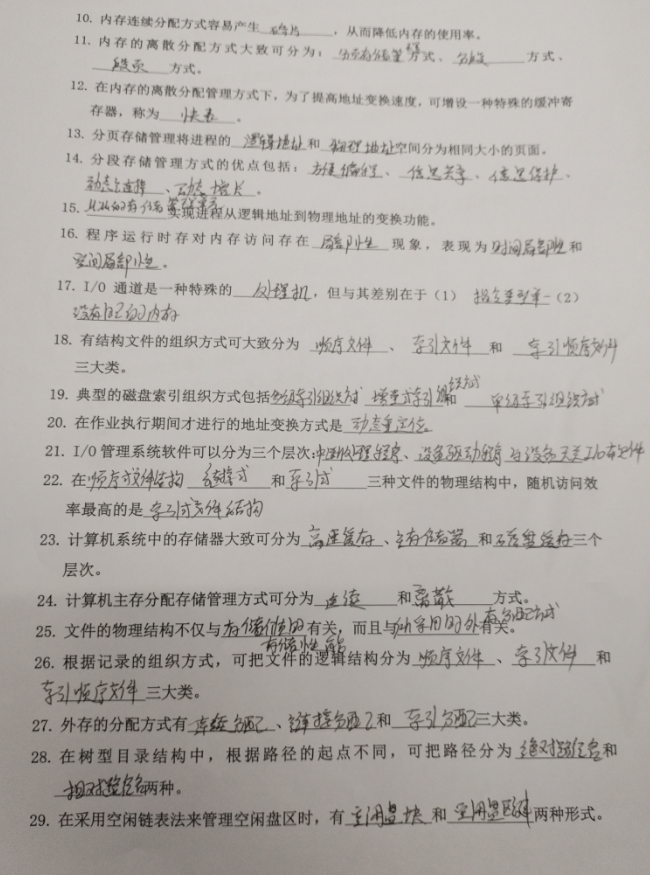
用户可见的文件组织方式，如流式文件或记录式文件。

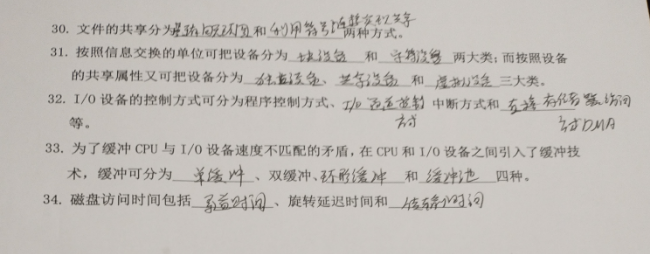
20. 文件的物理结构

文件在磁盘上的实际存储方式（如连续分配、链式分配、索引分配）。

1. 填空题







1. 简答题

1. 计算机存储器系统层次及性能分析

层次结构（自顶向下）：

寄存器：CPU内部，速度最快，容量最小（KB级）。

高速缓存（Cache）：分L1/L2/L3，速度ns级，MB级容量。

主存（RAM）：速度10-100ns，GB级容量。

外存（磁盘/SSD）：速度ms级，TB级容量。

备份存储（磁带/云存储）：速度最慢，容量无限扩展。

性能分析：

速度：寄存器 > Cache > 主存 > 外存 > 备份存储。

容量/成本：与速度成反比，层次越低容量越大、单位成本越低。

局部性原理：通过缓存技术（Cache、虚拟内存）弥补速度差距。

2. 程序从编写到运行的主要过程

编辑：编写源代码（如.c文件）。

编译：编译器生成目标文件（.o，含机器码和符号表）。

链接：静态/动态链接库，生成可执行文件（如.exe）。

加载：操作系统将程序读入内存，建立进程映像。

执行：CPU逐条执行指令，可能涉及动态链接、系统调用等。

终止：释放内存，返回退出状态码。

3. 程序链接的主要方式及异同

链接方式 时机 优点 缺点

静态链接 编译时 独立性强，运行快 文件大，更新需重新编译

动态链接 运行时 节省内存，支持热更新 依赖环境，启动稍慢

相同点：均需解析符号地址，生成可执行代码。

4. 内存动态分区分配算法

搜索方式分类：

首次适应（FF）：从低地址开始找第一个足够大的空闲区。

最佳适应（BF）：选择最小的可用空闲区，减少碎片。

最坏适应（WF）：选择最大的空闲区，避免小碎片。

循环首次适应（NF）：从上次分配位置继续搜索，均衡分配。

5. 内存分区算法 vs CPU调度算法

内存分区算法：FF、BF、WF、NF（关注空间分配）。

CPU调度算法：FCFS、SJF、RR、优先级调度（关注时间分配）。

相同点：均需解决资源竞争，优化效率。

不同点：内存算法侧重减少碎片，CPU算法侧重响应时间/吞吐量。

6. 页面置换算法 vs CPU调度算法

页面置换算法：FIFO、LRU、Clock（淘汰内存页）。

CPU调度算法：RR、SJF、多级队列（分配CPU时间片）。

相同点：均基于局部性原理或公平性。

不同点：页面置换关注缺页率，CPU调度关注任务响应速度。

7. 页面置换算法 vs 磁盘调度算法

页面置换算法：LRU、FIFO（管理内存页）。

磁盘调度算法：SCAN、C-SCAN、LOOK（优化磁头移动）。

相同点：均需减少访问延迟。

不同点：页面置换无物理移动开销，磁盘调度需考虑寻道时间。

8. 分页/分段/段页式地址变换

分页：虚拟地址→页号+页内偏移→查页表→物理地址。

分段：虚拟地址→段号+段内偏移→查段表→物理地址。

段页式：先分段再分页，需查段表和页表两级结构。

9. 中断处理步骤

保护现场：保存寄存器状态到内核栈。

识别中断源：读取中断向量表。

执行处理程序：如键盘输入处理。

恢复现场：恢复寄存器状态。

返回断点：继续原程序执行。

10. 文件类型分类

用途：系统文件、用户文件、库文件。

数据类型：文本文件、二进制文件。

组织方式：顺序文件、索引文件、哈希文件。

管理方式：普通文件、目录文件、设备文件。

11. 文件目录分类及查询

单级目录：全盘线性搜索，效率低。

二级目录：用户目录+文件目录，缓解命名冲突。

树形目录：支持子目录（如Linux /home/user），路径查询。

哈希目录：通过文件名哈希快速定位。

12. 文件组织方式对比

方式 优点 缺点

顺序结构 连续读写快 插入/删除效率低

链式结构 动态扩展方便 随机访问慢

索引结构 支持快速随机访问 索引表占用额外空间

13. 内存 vs 外存分配方式

内存分配：连续分配（固定/动态分区）、分页、分段。

外存分配：连续分配、链式分配、索引分配。

相同点：均需解决碎片问题。

不同点：内存分配关注速度，外存关注存储效率。

14. 提高磁盘I/O速度的途径

磁盘调度算法：如SCAN减少寻道时间。

缓存技术：使用磁盘缓存（如Buffer Cache）。

RAID技术：并行读写多磁盘。

预读（Read-ahead）：提前加载相邻数据。

文件系统优化：如ext4的延迟分配。