

期末知识点总结梳理 (chapter5)

- 不同的存储器之间的区别
 - 易失性存储
 - Cache (SRAM)
 - Memory (DRAM)
 - 非易失性存储
 - 闪存：闪存存储器(Flash ROM)
 - SLC：单阶存储单元闪存每个存储单元内存储1个信息比特
 - MLC：多阶存储单元闪存每个存储单元内存储2个以上的信息比特
 - TLC：三阶存储单元闪存每个存储单元内存储3个信息比特
 - 磁盘
 - 记录密度：单位长度内所存储的二进制信息量
 - 磁盘位密度 $D_b = \frac{f_t}{\pi d_{min}}$
 - f_t 每道总位数； d_{min} 同心圆最小直径
 - 存储容量
 - 磁盘存储器容量 $C = n \times k \times s$
 - n 代表存放信息的盘面数
 - k 为每个盘面的磁道数
 - s 为每条磁道记录的二进制代码数
 - 磁带
 - 磁盘保固期最多3-5年，磁带最长可达50年。
 - 光盘
 - 光盘是以光信息做为存储的载体并用来存储数据的一种物品。
 - 如何组合多个存储器提供稳定的存储服务
 - 磁盘阵列(RAID)
 - 如何防止存储错误
 - 纠错/检错 (汉明编码)
 - Fault: failure of a component
 - May or may not lead to system failure
 - Reliability: mean time to failure (MTTF)
 - Service interruption: mean time to repair (MTTR)
 - Mean time between failures

- $MTBF = MTTF + MTTR$
- $Availability = MTTF / (MTTF + MTTR)$
- 存储层次结构的设计依据
 - 时间局部性
 - 空间局部性
- 存储层次结构的工作原理
 - 处理器
 - 缓存 (Cache) - SRAM memory 连接到 CPU 的高速缓存存储器
 - 主存储器 - DRAM memory
 - 磁盘存储
- 存储层次结构中的Cache的原理
 - Cache读访问原理
 - Hit (命中)
 - 条件: 要访问的数据在上层存储中存在
 - 结果: 直接从上层 (更快的) 存储中获取数据
 - 性能指标: $Hit\ ratio = hits / accesses$
 - Miss (缺失)
 - 条件: 要访问的数据不在上层存储中
 - 处理: 从下层存储复制整个块到上层
 - 代价: Miss penalty (缺失惩罚) - 花费的额外时间
 - 然后: 从上层存储提供所访问的数据
 - Cache写访问原理
 - 写直达
 - 写命中时: 同时更新cache和内存
 - 问题: cache和内存保持一致, 但写操作变慢
 - 写回法
 - 写命中时: 只更新cache中的块, 不立即写内存
 - 跟踪机制: 记录每个块是否为"dirty" (脏块)
 - 写回时机: 当脏块被替换时才写回内存
 - 每个cache块需要额外的控制位:

Cache Block Structure:

```

+-----+-----+-----+-----+
| Valid | Tag  | Data | Dirty  |
+-----+-----+-----+-----+
  
```

● 写回法具体

2. 读操作流程

Cache Hit

1. 检查Valid位和Tag匹配
2. 直接返回cache中的数据
3. 不影响Dirty位

Cache Miss

1. 检查要替换的块是否为dirty
2. 如果dirty, 先将该块写回内存
3. 从内存读取新块到cache
4. 设置Valid=1, Dirty=0
5. 返回请求的数据

3. 写操作流程

Cache Hit (关键操作)

1. 检查Valid位和Tag匹配
2. 更新cache中的数据
3. 设置Dirty=1 (标记为脏块)
4. 不写内存!

Cache Miss

有两种处理方式:

写分配 (Write Allocate) :

1. 先处理miss (如读操作miss)
2. 将块载入cache
3. 然后执行写操作
4. 设置Dirty=1

1. 先处理miss (如读操作miss)
2. 将块载入cache
3. 然后执行写操作
4. 设置Dirty=1

写不分配 (No Write Allocate) :

1. 直接写内存
2. 不将块载入cache

● 特性	Write-through	Write-back
● 一致性	始终一致	延迟一致
● 性能	较慢	较快
● 复杂度	简单	复杂
● 内存流量	高	低

● 地址映射方式

-