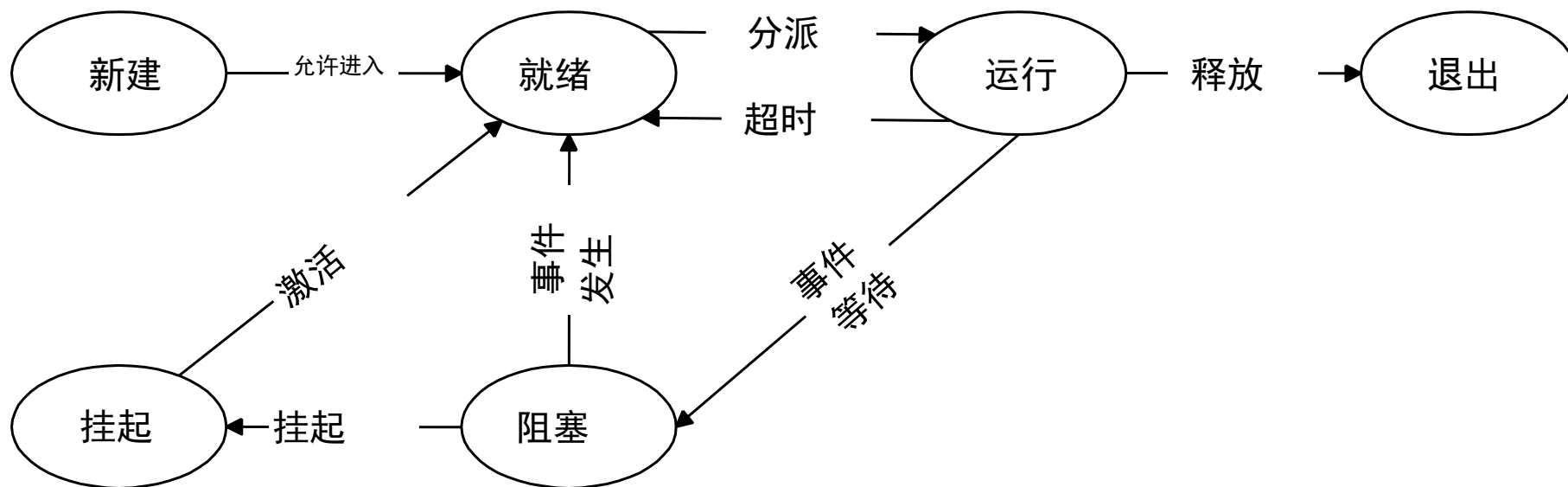


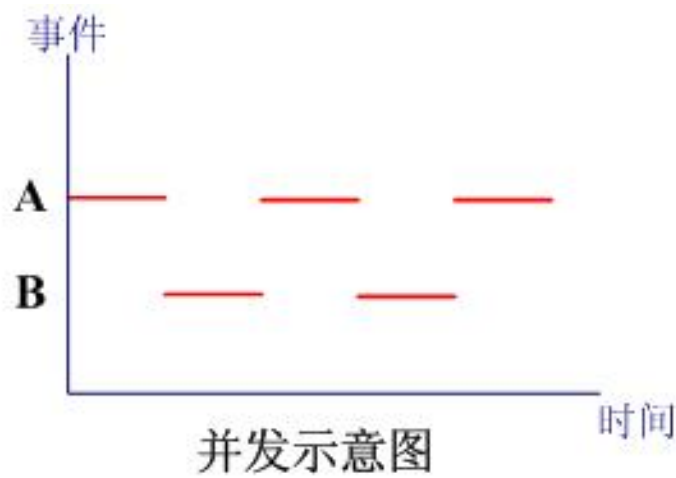
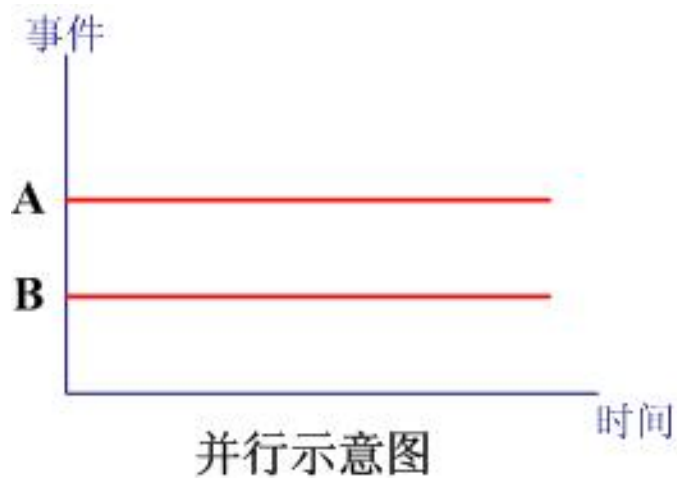
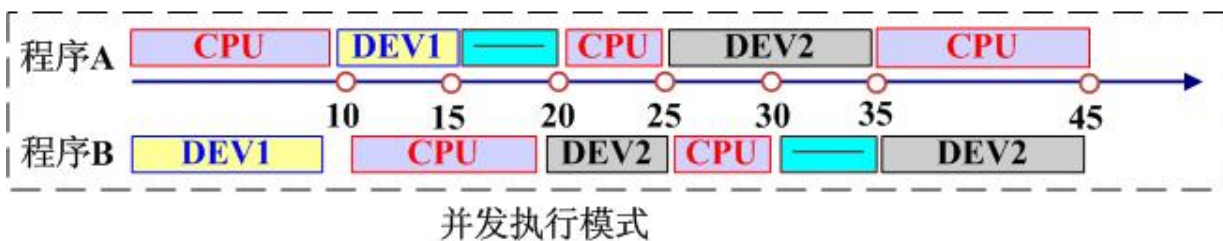
# 基础实验总结

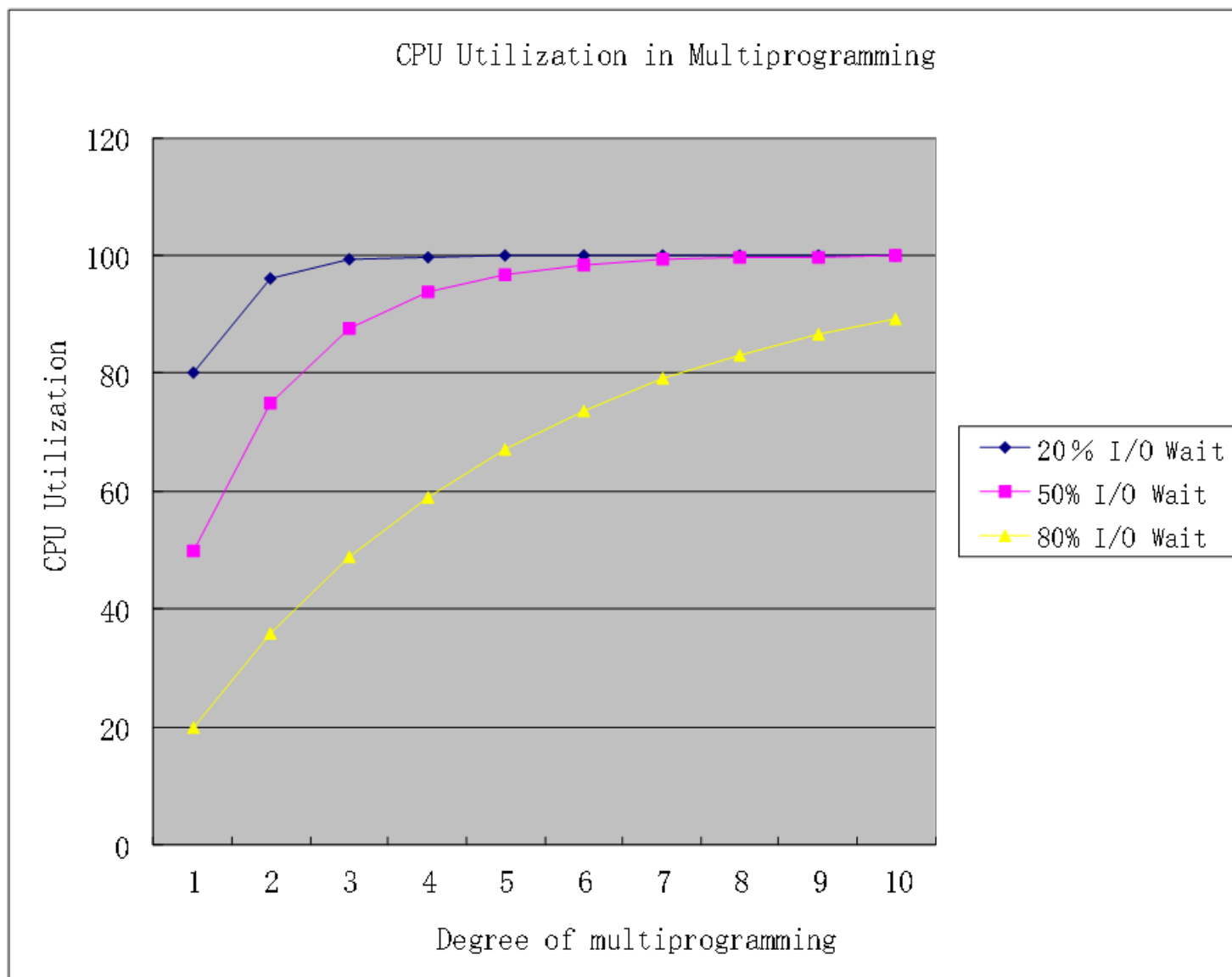
---

- 编程知识
  - 提供给管理员的多种命令
    - 文件读写、重命名、正则表达式、管理进程和线程、管理共享资源
  - 提供给程序员的多种API
    - 文件读写、虚拟地址空间映射、创建进程线程、创建IPC、网络通信、处理异常
  - 提供给程序员的多种工具
    - Makefile, gdb, ipcs, netstat, top, eBPF等

- 操作系统知识
  - 进程状态的管理
  - 进程地址空间的管理
  - 多核心、多进程、多线程的管理
  - 文件系统的管理
  - 设备的设备

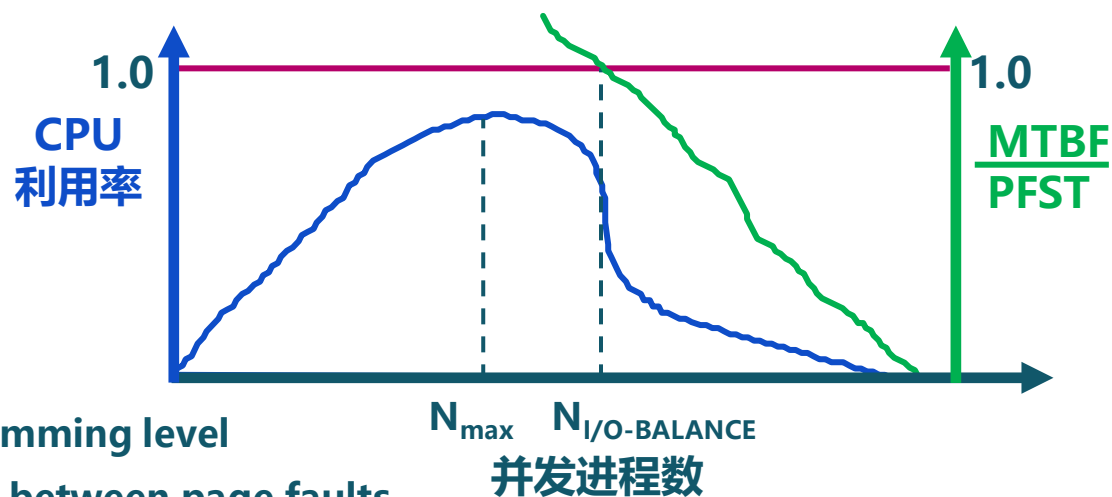






## 负载控制

- 通过调节并发进程数 (MPL) 来进行系统负载控制
  - ▣  $\sum WSi$  = 内存的大小
  - ▣ 平均缺页间隔时间(MTBF) = 缺页异常处理时间(PFST)



MPL-multiprogramming level

MTBF-mean time between page faults

PFST-page fault service time

## 抖动问题(thrashing)

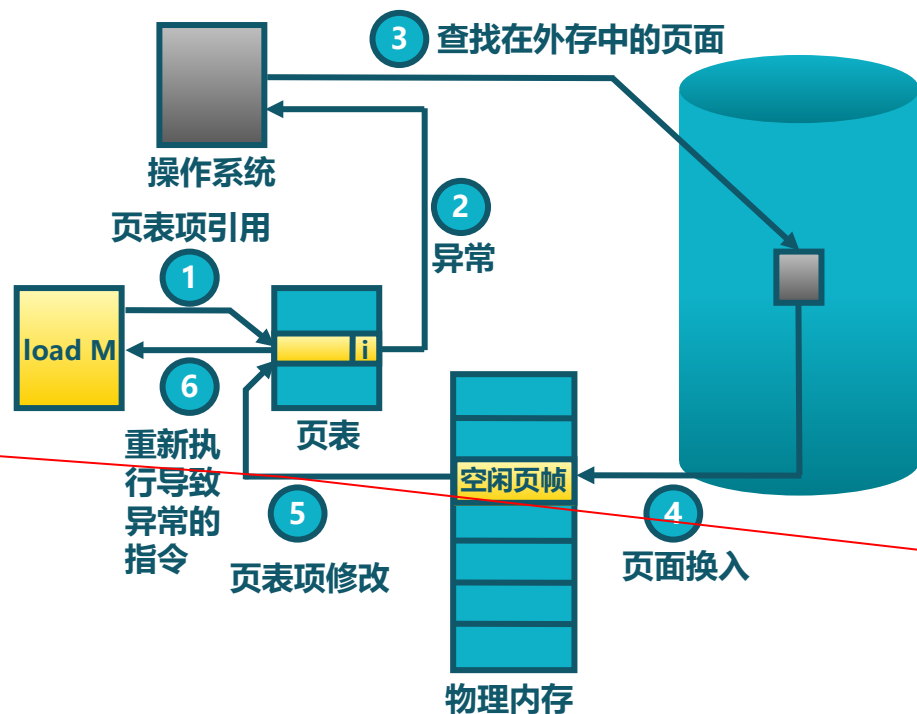
- 抖动
  - ▣ 进程物理页面太少，不能包含工作集
  - ▣ 造成大量缺页，频繁置换
  - ▣ 进程运行速度变慢
- 产生抖动的原因
  - ▣ 随着驻留内存的进程数目增加，分配给每个进程的物理页面数不断减小，缺页率不断上升
- 操作系统需在并发水平和缺页率之间达到一个平衡
  - ▣ 选择一个适当的进程数目和进程需要的物理页面数



例：执行 `int variable = *(p+102400);`  
或者执行 `*p(204800) = 4096;`

## 缺页异常（缺页中断）的处理流程

在现代的操作系统中，通常这个换出的过程不会阻塞在换入的过程中发生，而是将这一过程拆成两段。有一个线程在系统不忙且空闲页框不足时，完成这些写回和释放的过程，在缺页时确保有可用的页框

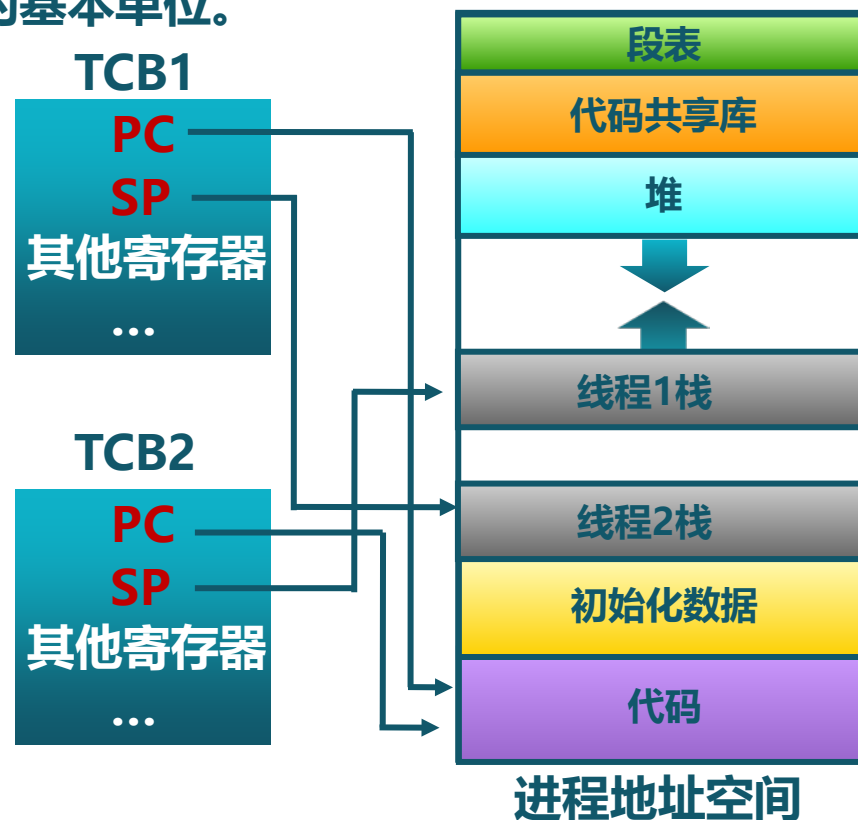


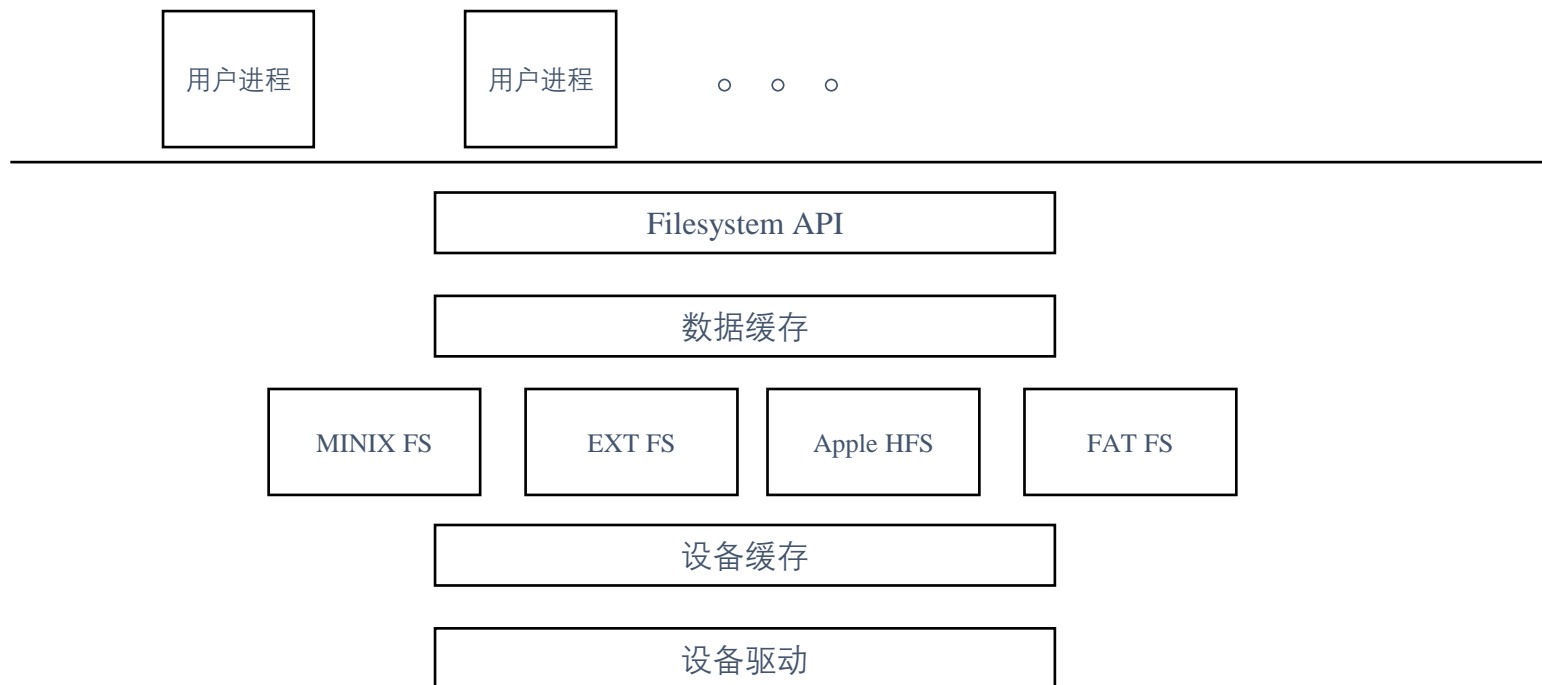
- 在内存中有空闲物理页面时，分配一物理页帧 $f$ ，转第E步；
- 依据页面置换算法选择将被替换的物理页帧 $f$ ，对应逻辑页 $q$
- 如 $q$ 被修改过，则把它写回外存；
- 修改 $q$ 的页表项中驻留位置为0；
- 将需要访问的页 $p$ 装入到物理页面 $f$
- 修改 $p$ 的页表项驻留位为1，物理页帧号为 $f$ ；
- 重新执行产生缺页的指令

查找vma对应的描述（系统中的maps），找到该页对应的文件和偏移量  
借助文件系统，将文件名和偏移量，转换成磁道和扇区  
驱动磁盘的磁臂和磁头，取到数据

线程是进程的一部分，描述指令流执行状态。它是进程中的**指令执行流**的最小单元，是CPU**调度**的基本单位。

- ▶ 进程的资源分配角色：  
进程由一组相关资源构成，包括地址空间（代码段、数据段）、打开的文件等各种资源
- ▶ 线程的处理机调度角色：  
线程描述在进程资源环境中的指令流执行状态





在没有中断、异常发生，没有人发起系统调用的时候  
操作系统在干什么？

应用程序应该如何处理与操作系统的关系？



感谢阅读

---