- Linux
  - 平均负载
    - 概念 —— 平均活跃进程数 = 可运行进程数 / 不可中断进程数 R/D
    - 上升原因
      - 处理器密集型
      - IO密集型 —— 但CPU使用率不一定高
      - 大量cpu进程调度 -> 导致CPU使用率上升
        - 上下文
          - CPU寄存器
          - 程序计数器 —— 下一条要执行指令
        - 上下文切换
          - 进程上下文切换
            - 内核态
            - 用户态 —— 系统调用
              - 一次系统调用 需要 两次CPU上下文切换
              - 注意：不会切换进程
              - 指：在同一个进程中切换
            - 指：一个进程切换到另一个进程中
              - 保存用户态的 虚拟内存、栈、全局变量等
              - 保存内核态的程序计数器和CPU寄存器
              - 过程：保存当前进程”状态“ -> 加载其他进程”状态“ —— 正是减少进程运行的”元凶“
            - 触发场景
              - 公平竞争 —— 划分平等的时间片
              - 系统资源不足时
              - 自己挂起 —— sleep函数
              - 更高优先级的进程运行
              - 硬件中断 —— 会挂起进程，跳转内核中执行中断进程
          - 线程上下文切换
            - 线程是**调度**的基本单位
            - 进程时**资源拥有**的基本单位
            - 当进程只有一个线程时可理解线程就是进程
            - 当有多个线程在同一进程时，多线程共享进程提供的资源：共享虚拟内存和全局变量
            - 线程拥有自己的栈、寄存器、程序计数器等，上下文切换时需要保存
            - 切换方式
              - 两个线程在不同的进程中 —— 与进程切换上下文一致
              - 两个线程在同一进程中
                - 仅保存线程中的栈、程序计数器等 —— 消耗资源要比进程切换要少虚拟内存，全局变量等
                - 也是多线程代替多进程的主要原因。
          - 中断上下文切换
            - 优先级更高
            - 短小精悍
            - 由于中断是在内核态切换 —— 所以仅保存栈，寄存器，硬件参数等
        - TODO 上下文切换的分析
  - Command
    - CPU核数 —— grep 'model name' /proc/cpuinfo | wc -l
    - mpstat —— mpstat -P ALL 5
    - pidstat —— pidstat -u 5 1
    - iostat —— 磁盘io