

# 操作系统

---

REFERENCE: <http://jyywiki.cn/>

## 目录

---

1. (why)为什么要学操作系统?
2. (what)什么是操作系统?
3. (how)如何学操作系统?

# 为什么要学操作系统？

---

EAXMALP:

为什么要学微积分？（下面是微积分的三个重要的议题）

1. 启蒙、应用与挑战（Newton 时代）
  - 机械论世界观（模型驱动的系统分析）
  - 数学是理解世界的“基本工具”：导数、微积分基本定理、.....
2. 严格化与公理化（Cauchy 时代）
  - 各种卡出的 bug（Weierstrass 函数、Peano 曲线.....）
3. 大规模问题的数值计算（von Neumann 时代）
  - 优化、有限元、PID.....

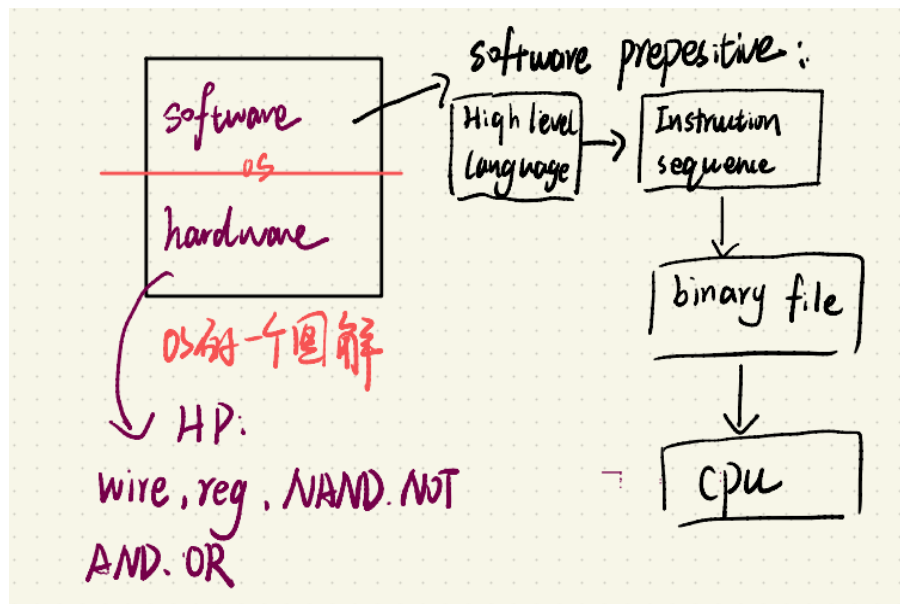
重走从无到有的**发现历程**

- **基本思想、基本方法、里程碑、走过的弯路**
- 最终目的：**应用/创新**（做题得分不是目的而是手段）

---

作业 1：看完 `man bash`

# 什么是操作系统?



## 理解操作系统

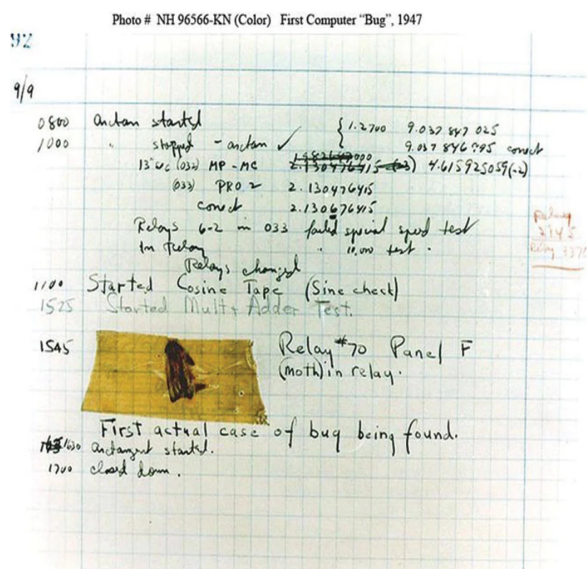
- 理解硬件（计算机）和软件（程序）的发展历史
- 夹在中间的就是操作系统

## 操作系统的一些发展历程

- 1940s的计算机软件

打印平方数、素数表、计算弹道.....

- 大家还在和真正的“bugs”战斗



- 1950s的计算机硬件：I/O设备的速度已经远低于CPU速度。⇒ 引入了中断机制

- 1960s的计算机软件：HLL(high-level language)开始出现。

## 1950s开始出现OS

---

### 1940s 的操作系统

没有操作系统。

能把程序放上去就很了不起了

- 程序直接用指令操作硬件
  - 不需要画蛇添足的程序来管理它
- 

### 1950s 的操作系统

库函数 + 管理程序排队运行的调度代码。

写程序（戳纸带）、跑程序都是非常费事的

- 计算机非常贵（\$50,000-\$1,000,000\$50,000-\$1,000,000），一个学校只有一台
  - 算力成为一种服务：多用户轮流共享计算机，operator 负责调度
- 

操作系统的概念开始形成

- 操作（operate）任务（jobs）的系统（system）
  - “批处理系统” = 程序的自动切换（换卡）+ 库函数 API
  - Disk Operating Systems (DOS)
    - 操作系统中开始出现“设备”、“文件”、“任务”等对象和 API

### 1960s 的计算机硬件

集成电路、总线出现

- 更快的处理器
- 更快、更大的内存；虚拟存储出现
  - 可以同时载入多个程序而不用“换卡”了
- 更丰富的 I/O 设备；完善的中断/异常机制

## 1960s 的操作系统

能载入多个程序到内存且调度它们的管理程序。

为防止程序之间形成干扰，操作系统自然地共享资源（如设备）以 API 形式管理起来

- 有了进程（process）的概念
- 进程在执行 I/O 时，可以将 CPU 让给另一个进程
  - 在多个地址空间隔离的程序之间切换
  - 虚拟存储使一个程序出 bug 不会 crash 整个系统

---

操作系统中自然地增加进程管理 API

- 既然可以在程序之间切换，为什么不让它们定时切换呢？
- Multics (MIT, 1965): 现代分时操作系统诞生

---

1970s 与现代计算机并无不同。

# 课程内容概述

---

操作系统：软件硬件之间的桥梁

- 本课程中的软件：多线程 Linux 应用程序
  - 本课程中的硬件：现代多处理器系统
- 

（设计/应用视角）操作系统为应用提供什么服务？

- 操作系统 = 对象 + API
  - 课程涉及：POSIX + 部分 Linux 特性
- 

（实现/硬件视角）如何实现操作系统提供的服务？

- 操作系统 = C 程序
  - 完成初始化后就成为 interrupt/trap/fault handler
- 课程涉及：xv6，自制迷你操作系统

# 如何学操作系统

---

## 最重要的：Get Your Hands Dirty

---

听课看书都不重要。独立完成编程作业即可理解操作系统。

应用视角（设计）：Mini Labs x 6

- 使用 OS API 实现“黑科技”代码

---

硬件视角（实现）：OS Labs x 5

- 自己动手实现一个真正的操作系统

---

全部 Online Judge

- 代码不规范 → `-Wall -Werror` 编译出错
- 代码不可移植 → 编译/运行时出错：`int x = (int)&y;`
- 硬编码路径/文件名 → 运行时出错：`open("/home/a/b", ..`

## Prerequisites

---

计算机专业学生必须具备的核心素质。

### 1. 是一个合格的操作系统用户

- 会 STFW/RTFM 自己动手解决问题
- 不怕使用任何命令行工具
  - vim, tmux, grep, gcc, binutils, ...

### 2. 不怕写代码

- 能管理一定规模（数千行）的代码
- 能在出 bug 时默念“机器永远是对的、我肯定能调出来的”
  - 然后开始用正确的工具/方法调试