中山大學 软件工程学院

SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

1924-2024 中山大學 世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

操作系统概述

SSE202/204: 操作系统原理

苏玉鑫

suyx35@mail.sysu.edu.cn

助教：龙玉丹 单诗雯 毛晨希 沈志轩 郑灿峰 胡伟峰

课程群

- 日常零碎消息
- 方便大家互加微信
  - 包括我和助教的
- 专业问题和讨论尽量在软工集市
  - 提出“蠢”问题没有心理负担
  - 微信群大概率只能人前显圣

群聊：操作系统原理课程群 25春

该二维码7天内(3月2日前)有效，重新进入将更新

苏玉鑫 软件工程学院副教授

- 分管本科教学的副院长

- 香港中文大学　　博士、博士后
- 浙江大学　　硕士
- 中山大学　　学士

B站：鸭大坑导

- 主要研究方向为利用人工智能技术提升系统软件可靠性与运行时性能，包括云原生系统、操作系统

- 发表人工智能、系统软件领域CCF-A类论文24篇

版权信息

- 部分内容来自：上海交通大学并行与分布式系统研究所操作系统课件
  - https://ipads.se.sjtu.edu.cn/courses/os/
- 其它参考资料：
  - 清华大学操作系统公开课
    - https://open.163.com/newview/movie/courseintro?newurl=ME1NSA351
    - 介绍标准内容，适合考研
  - 南京大学计算机软件研究所
    - http://jyywiki.cn/OS/2025/
    - https://space.bilibili.com/202224425/channel/collectiondetail?sid=192498
    - 比较有趣

主观题 10分

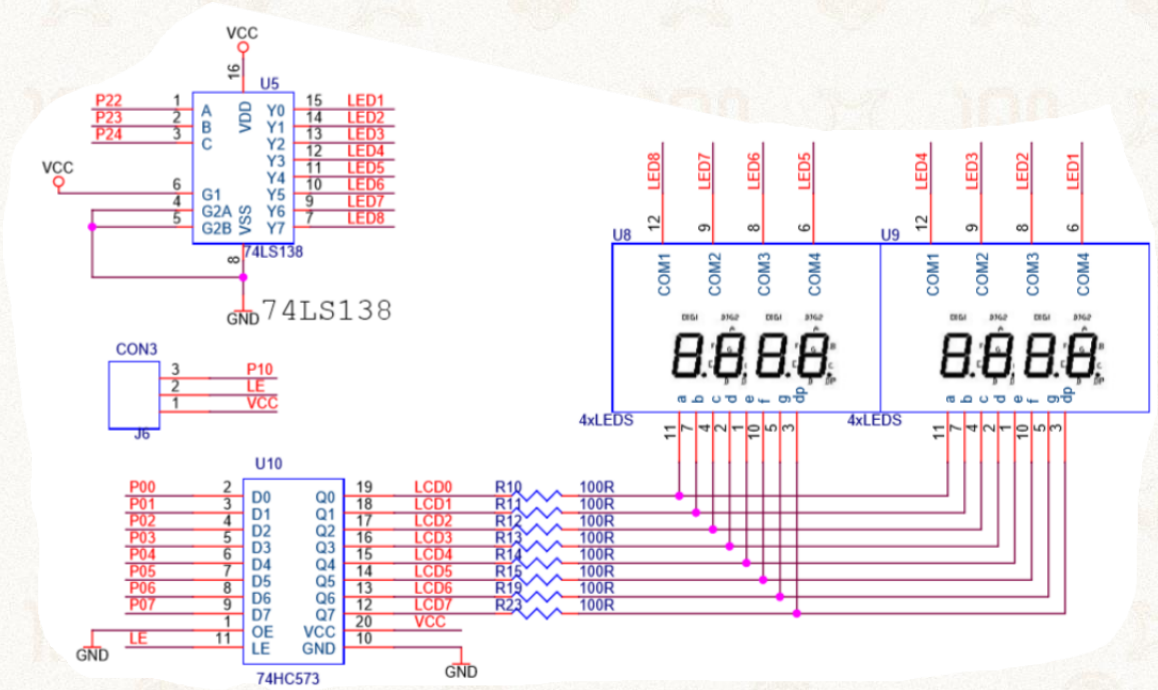
设置

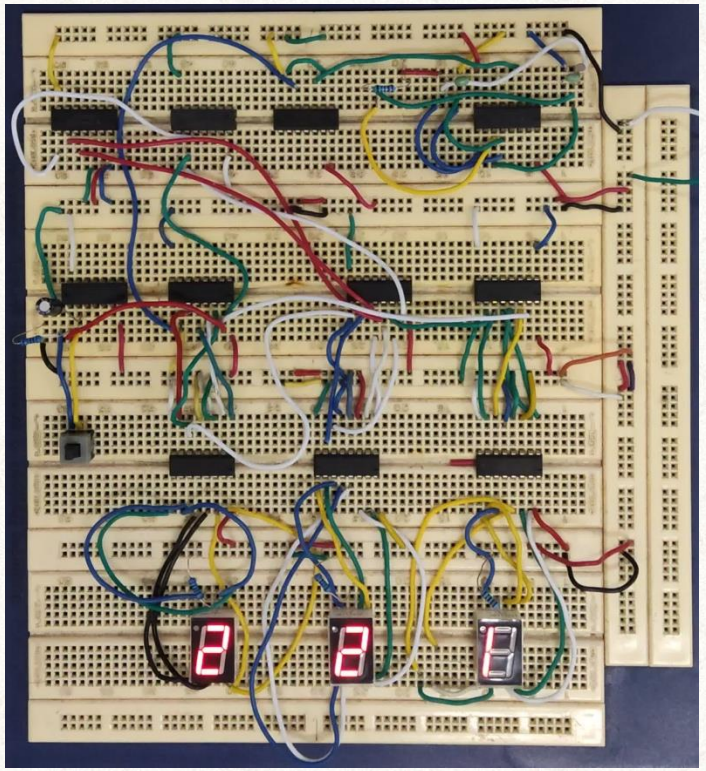
可以畅想一下，你觉得我们这门课要学什么？或者你想学到什么？

作答

大一的你是否有这样的疑惑：

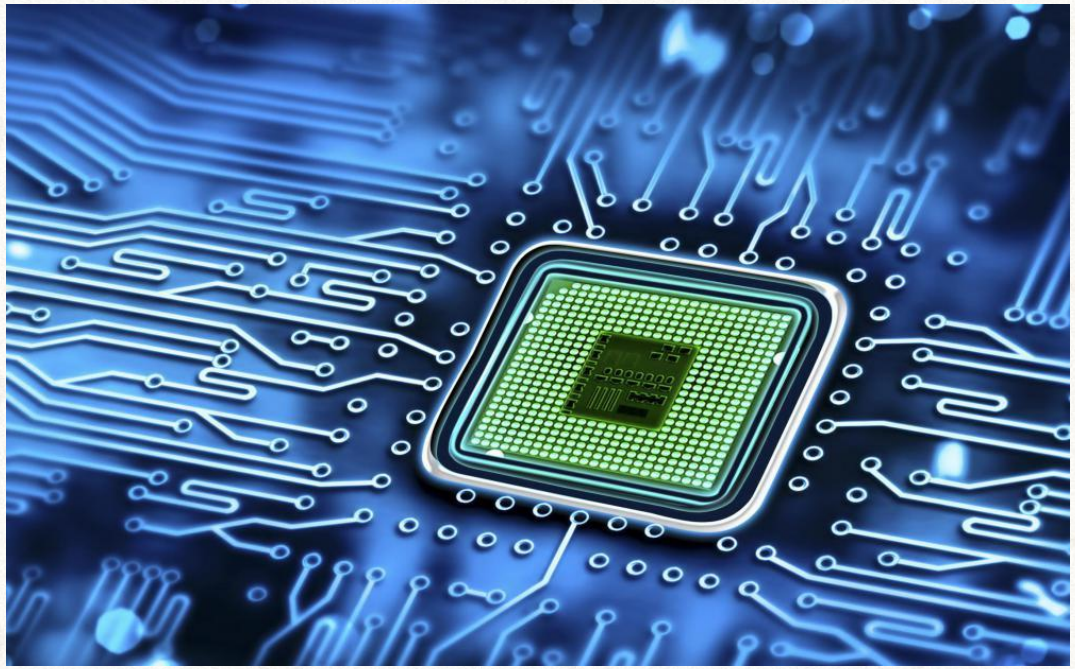
- 我为什么要学这些？

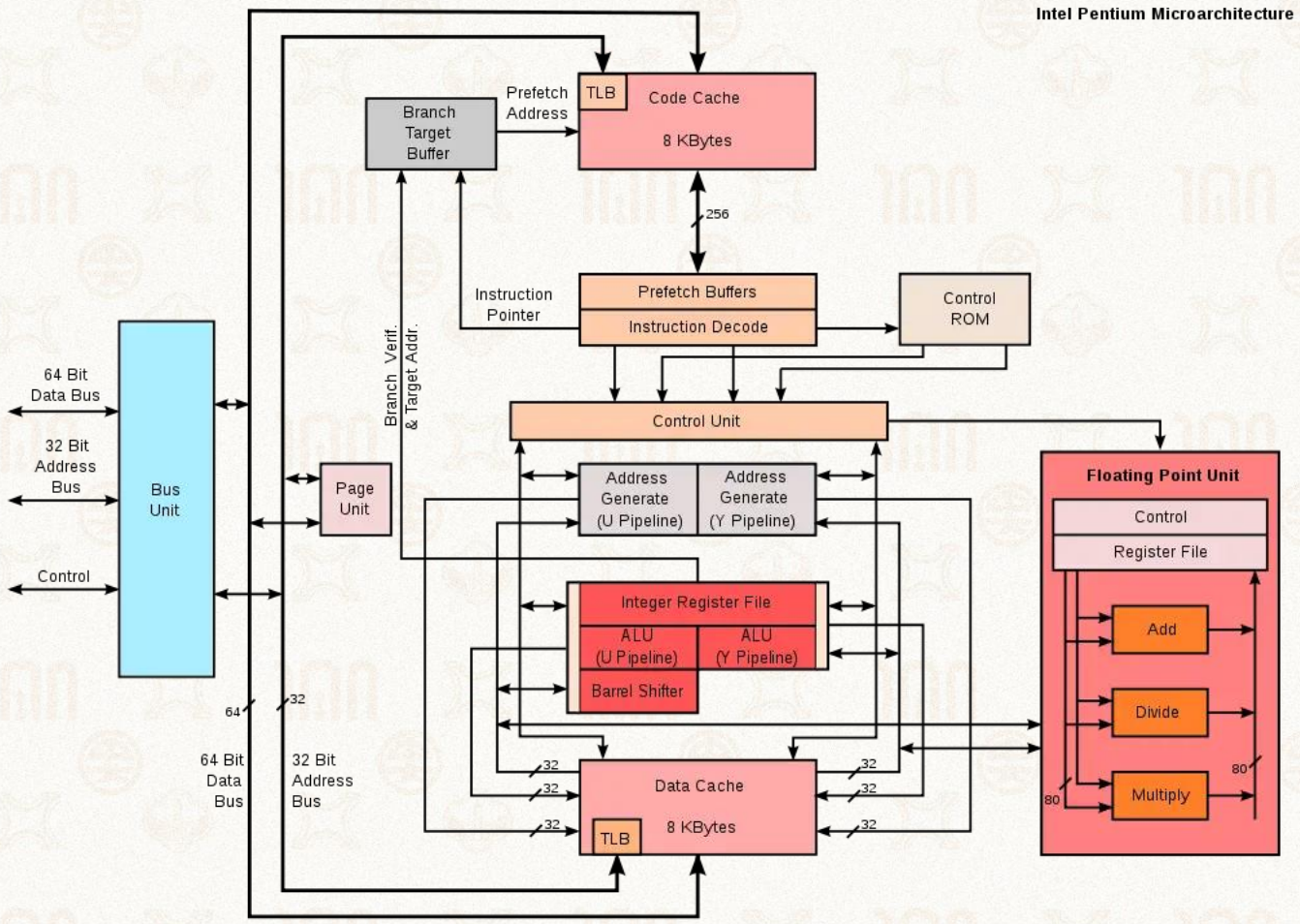




上个学期的你是否有这样的疑惑：

- 我为什么要学这些？

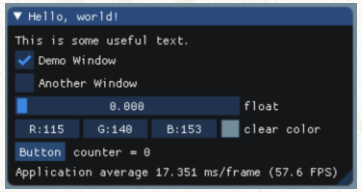




上个学期的你是否有这样的疑惑：

- 作为软工学生，我应该学这些：

应用程序开发

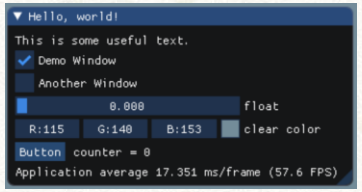


黑框、图形化界面、网页…

上个学期的你是否有这样的疑惑：

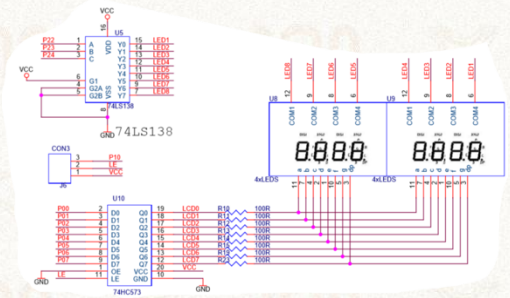
- 这两块内容有什么关联？

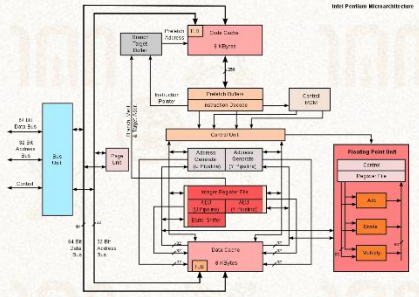
应用程序开发



黑框、图形化界面、网页…

硬件



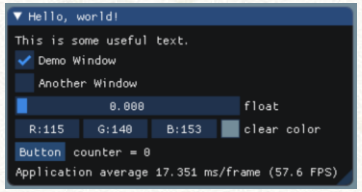


数字电路、计算机组成原理

上个学期的你是否有这样的疑惑：

- 这门课是用来终结割裂感的

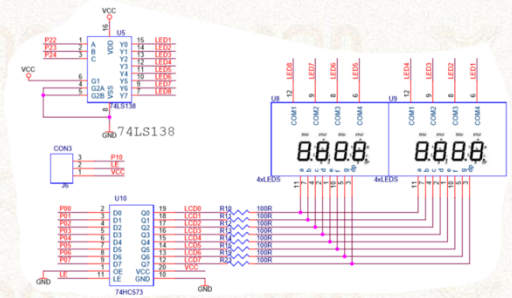
应用程序开发

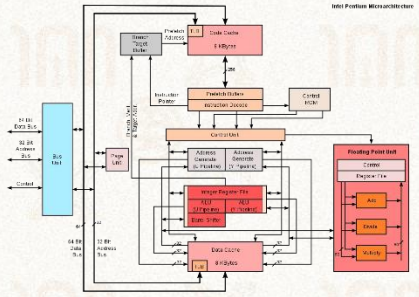


黑框、图形化界面、网页…

操作系统

硬件





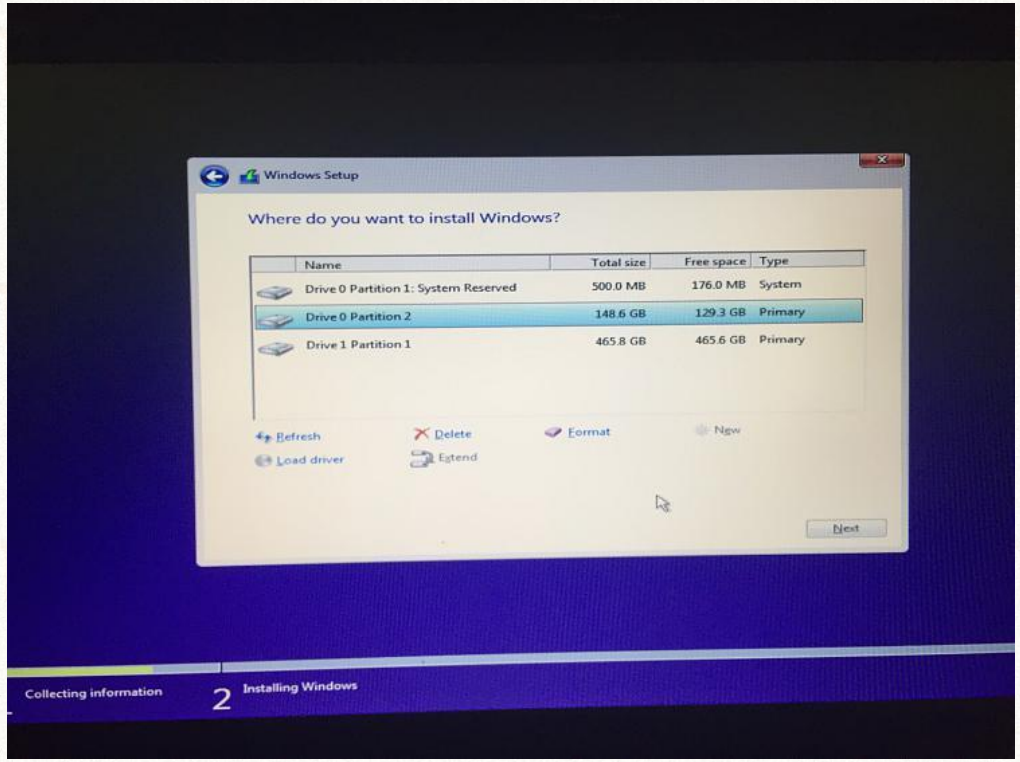
数字电路、计算机组成原理

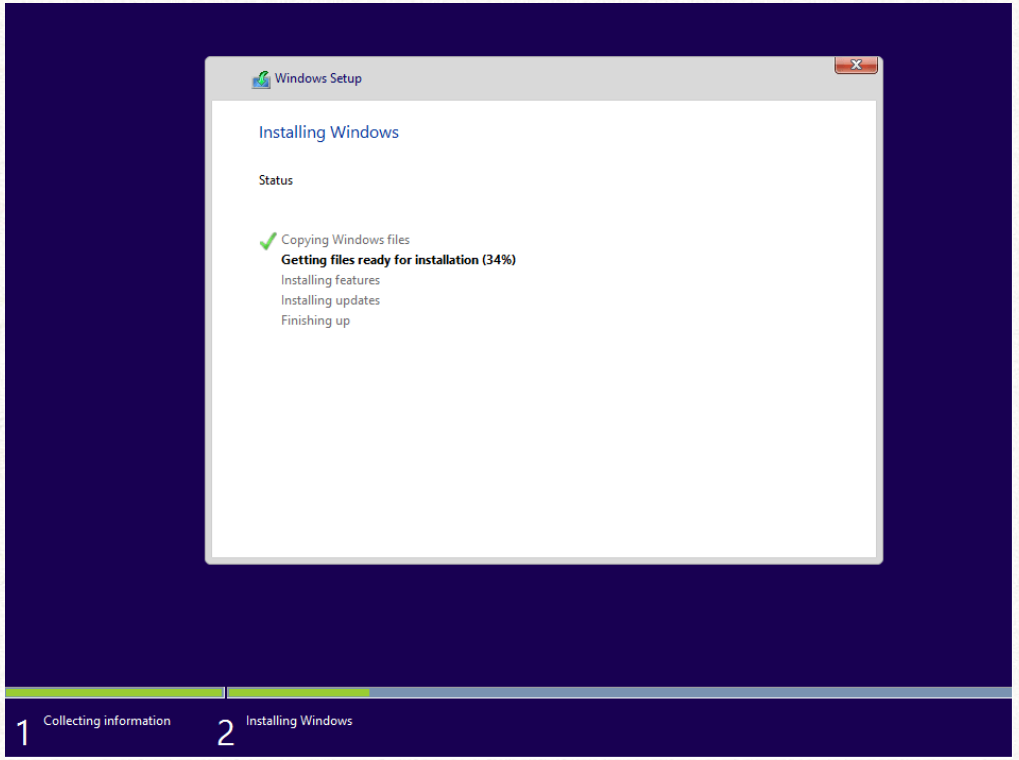
大纲

- 为什么要学习以及怎么学习操作系统
- 什么是操作系统
- 操作系统的历史

为了。。。

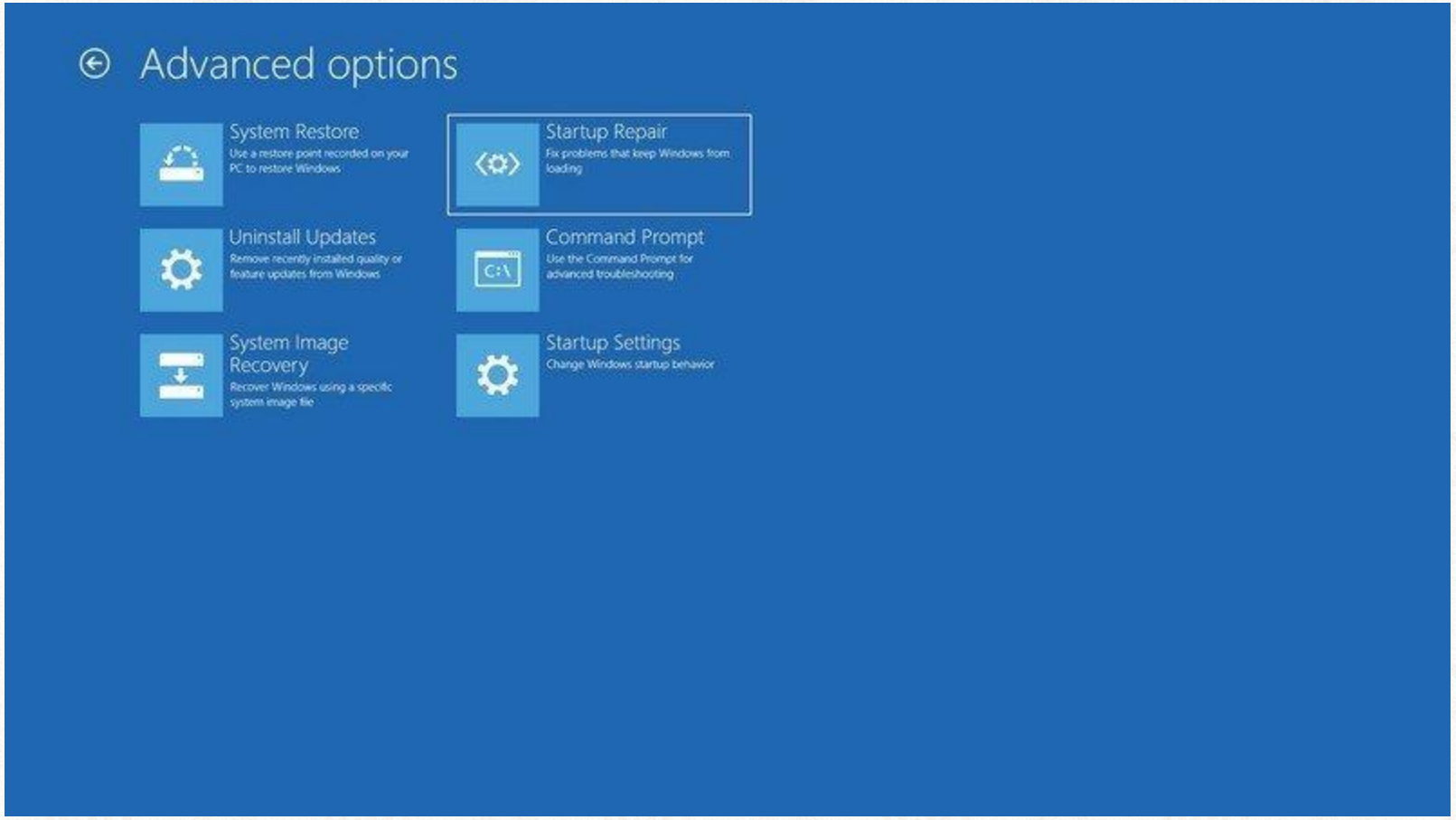
- 为了懂得如何重装操作系统?
  - 1节课足够





为了。。。

- 为了懂得蓝屏以后如何修复？然后帮其它专业同学修电脑？



单选题 1分

设置

哪边的速度更快些?

```
for(int i = 0; i < 10; i++) {
    a[i] = func(a[i]);
}
```

```
a[0] = func(a[0]);
a[1] = func(a[1]);
a[2] = func(a[2]);
a[3] = func(a[3]);
a[4] = func(a[4]);
a[5] = func(a[5]);
a[6] = func(a[6]);
a[7] = func(a[7]);
a[8] = func(a[8]);
a[9] = func(a[9]);
i += 10;
i < 10;
// ...
```

A 左边快

B 差不多

C 右边快

提交

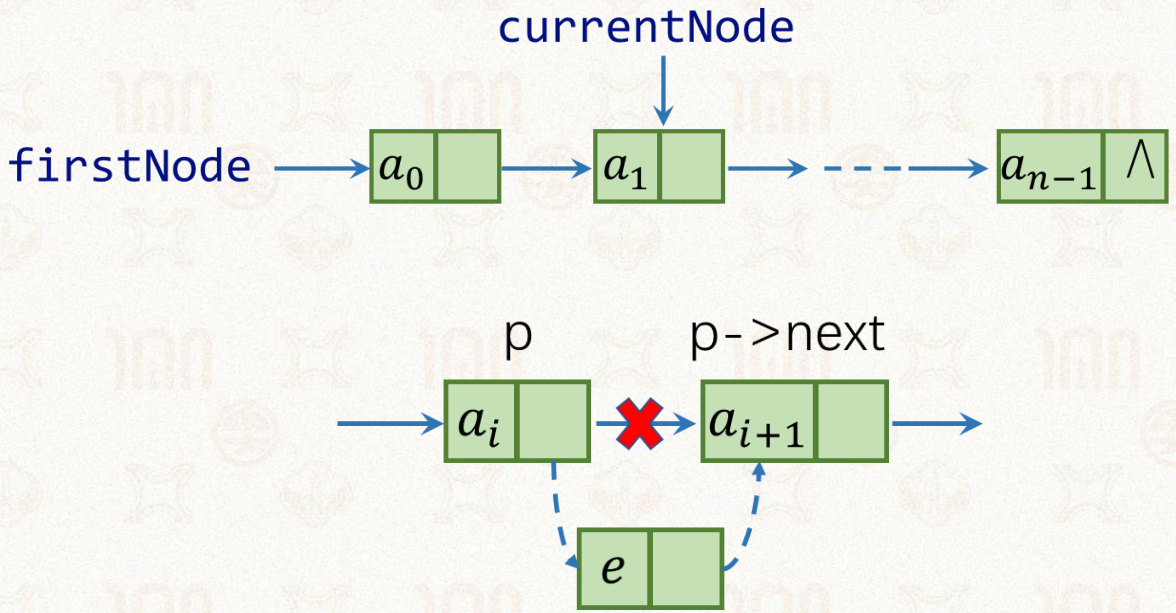
投票(匿名)

最多可选1项

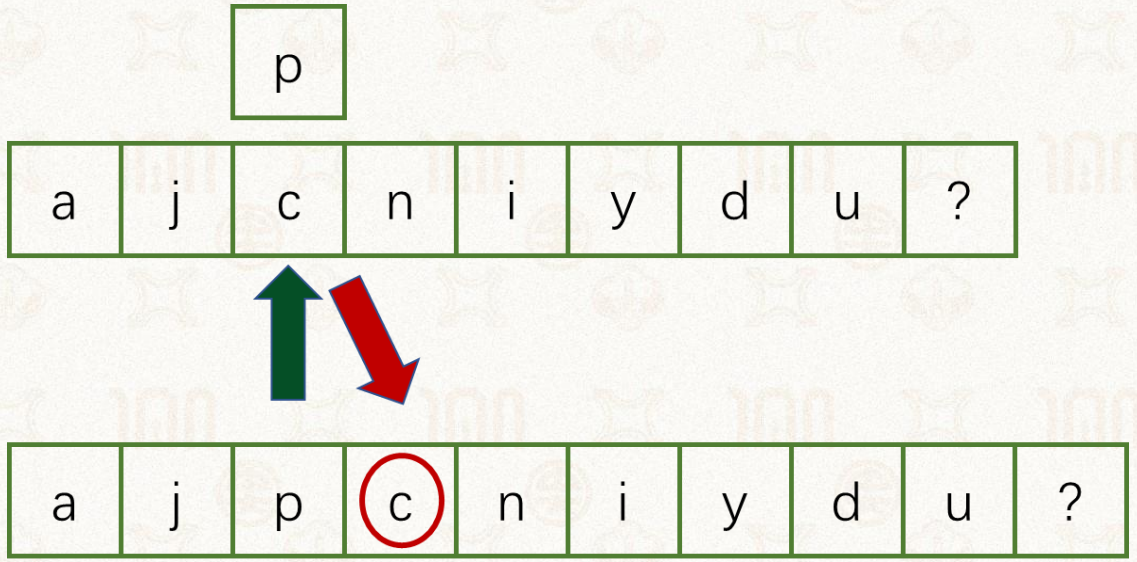
设置

为什么学习操作系统?
链表和顺序表的按值查找插入操作，谁更快一点?

A 链表



B 顺序表



提交

为什么学习操作系统?

- 可以成为一名更高效的程序员
  - 更快找到并消灭bug的能力
  - 理解并调试、提升程序性能的能力

- 提高程序性能的方法:

为什么学习操作系统?

- 拥有理解复杂系统设计与实现的能力
  - Linux kernel超过3000万行代码
    - 每年新增200万行代码
  - Windows 10据说超过1亿行代码

- 拥有构建一个操作系统的能力
  - 不急，实验课上就有

为什么学习操作系统?

- 操作系统是系统领域的基石
- 系统领域有大量的公司
  - 微软、谷歌、IBM、EMC、VMware、华为、阿里...
  - 谷歌的核心技术
    - K8s集群、GFS、MapReduce、BigTable
    - 都是系统领域的优秀工作
- 学好操作系统，for fun and profit
  - 优秀的公司，优秀的大学，更多的机会
  - 一线互联网公司的顶级offer
  - 华为的“天才少年”计划
  - 轻松跨越所谓的“35岁瓶颈”

为什么学习操作系统?

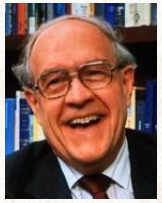
- 操作系统是一个成熟的领域
  - Windows曾经一统天下数十年
  - 让用户改变操作系统是很困难的
  - 我们是否需要新的操作系统?

- "新的"操作系统不断出现
  - Linux、Mac OS、Android、iOS、ROS...
  - 大疆用什么操作系统?
  - 谷歌数据中心用什么操作系统?
  - 我国的“揽月”着陆器用什么操作系统? ...

图灵奖与操作系统的演变

Maurice Wilkes
1967年图灵奖

EDSAC,1949
Multi-programming
第一台存储程序式电子计算机



Frederick Brooks
1999年图灵奖

IBM System/360,1964

Fernando J. Corbató
1990年图灵奖

CTSS,1961 & Multics,1969
分时操作系统

Unix,1971
多任务多用户操作系统

Ken Thompson & Dennis Ritchie
1983年图灵奖

Venus,1972
小型低成本交互式分时操作系统

Barbara Liskov
2008年图灵奖

2019
分布式操作系统

操作系统为什么难学?

- 深入事情本质：直接管理硬件细节
  - 好处：实现资源的高效利用，从根本上解决问题，做一个高效程序员（降维）
  - 挑战：需要理解与处理硬件细节，硬件甚至可能出错
  - “把复杂留给自己、把简单留给用户”
    - 对比：用户态写一个Hello World和在操作系统内核中输出一个Hello World

- 锻炼系统架构能力
  - 将复杂问题进行抽象与化简
  - 计算机科学中30%的原则是从操作系统中来的 (13/41, greatprinciples.org)

- 各种问题的交互与相互影响
  - fd = open (); … ; fork();

操作系统课程生存指南，一定要点开看看：http://jyywiki.cn/OS/OS_Guide

课程的特点：抽象与具体

- 许多课程强调抽象
  - 如：抽象数据类型、渐进分析等
  - 如：C++/Java/Python编程语言
  - 如：计算机网络多层结构

大量抽象的框图、架构图

- 这些抽象通常不可避免的带来限制
  - 尤其是当bug存在的时候
  - 需要理解底层的实现细节，才能突破限制

大量具体难懂的C语言代码

```
struct list_head {
    struct list_head *prev;
    struct list_head *next;
};
```

```
struct AnyType {
    struct list_head node;
    int a;
    double b;
    // ...
};
```

任意类型的双向链表

课程信息

- 理论（4学分，72学时）
  - 周一：一班5-6节、二班7-8节，D209
  - 周四：一班5-6节、二班7-8节，D209
- 实验（1学分，36学时)
  - 周三：一班1-2节、二班3-4节，A306

- 先修课程：
  - 数据结构与算法
    - 较多，操作系统是数据结构主要应用场景之一
  - 计算机组成原理
    - 较少，便于理解底层机制，没有过多要求

教学内容

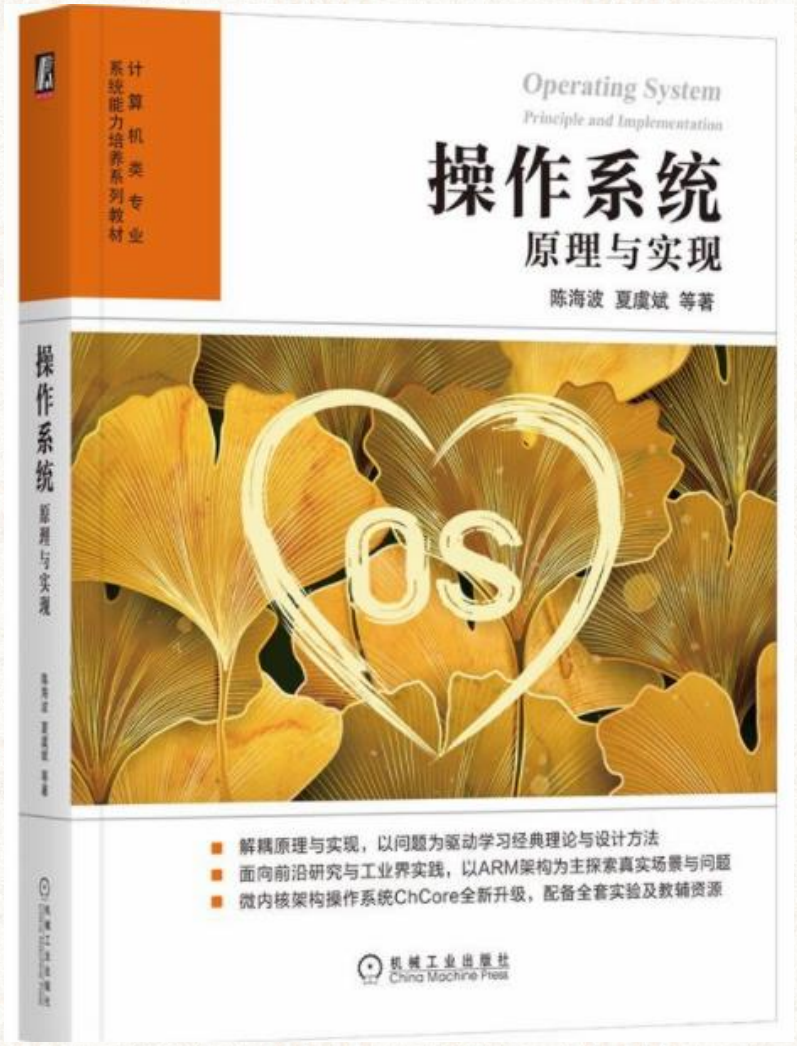
- 操作系统概述
- 硬件环境与软件抽象
- 操作系统结构
- 内存管理
- 进程管理
- 操作系统调度
- 进程间通信
- 同步原语
- 文件系统

基本原理

- 设备管理
- 系统虚拟化
- 多核与多处理器
- 文件系统崩溃一致性
- 网络协议栈与系统
- 轻量级虚拟化
- 操作系统安全
- 操作系统调测

进阶内容

教材



- 会按教材内容讲
  - 因为这本已经够“现代”、够“前沿”，不需要额外扩展
  - 作者陈海波
    - 国内操作系统研究领域的著名学者
    - 华为操作系统内核首席科学家
- 本书特色：
  - 经典理论+业界实践+前沿研究
- 辣眼睛的内容：
  - 选讲，不会考 🌶️

据多位学生反馈：找工作前多读几遍这本教材，面试加分不少！

实验课教学过程

理解

- 熟悉Linux系统的基本操作
- 理解课程所学知识点

实现

- 尝试编写和操作系统密切相关的复杂代码

科研探索

- 阅读论文，了解前沿进展

课程评分

理论课		
分数	考勤	4 %
	课后作业	36 %
	期末考试	60 %
	总共	100 %
实验课		
分数	考勤	10 %
	作业	40 %
	科研论文阅读报告	50 %
	总共	100 %

实验课：科研论文阅读报告

- 个人完成，但可以参考其它同学的内容（需引用）
  - 学期中段出详细评分标准
- 论文来源
  - 教材各章节后面的参考文献
  - 学期中段补充更多论文列表
- 报告在期末考试周之前交

大纲

- 为什么要学习以及怎么学习操作系统
- 什么是操作系统
- 操作系统的历史

多选题 1分

设置

你觉得以下哪些属于操作系统?

A Windows 10所包含的所有软件

B Linux内核以及所有设备的驱动

C 在Macbook上下载安装的第三方NTFS文件系统

D 华为Mate 60出厂时所有的软件

E 大疆无人机出厂时所有的软件

F 我国“揽月”着陆器上运行的软件

提交

什么是操作系统?

- 你觉得应当如何定义操作系统?

操作系统是在硬件和应用之间的软件层

应用

操作系统

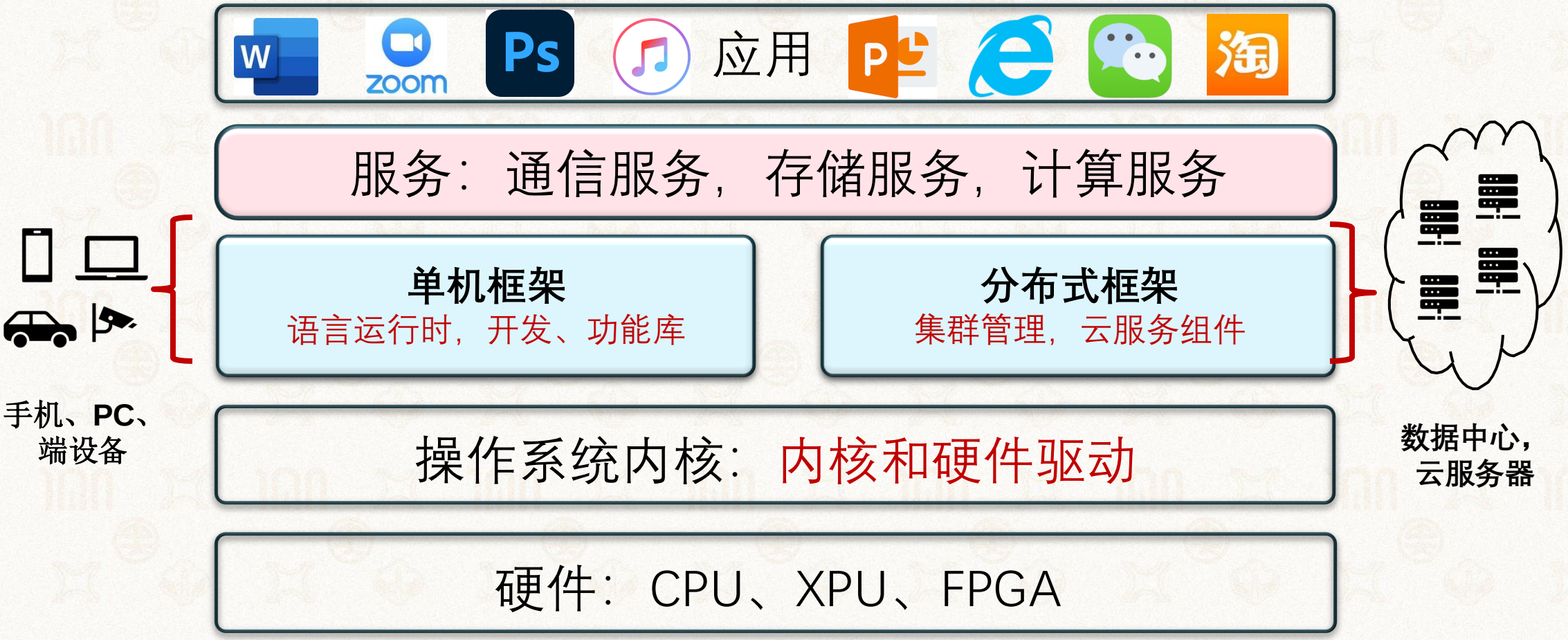
硬件

- **操作系统和应用的关系：**
  - 应用功能越来越多
  - 操作系统沉淀越来越多功能、 内涵与外延不断扩大
- **操作系统和硬件的关系：**
  - 意识 VS. 身体

"操作系统是管理硬件资源、控制程序运行、改善人机界面 和为应用软件提供支持的一种系统软件。"

[计算机百科全书(第2版)]

操作系统：对下管理硬件，对上构筑应用生态



大纲

- 为什么要学习以及怎么学习操作系统
- 什么是操作系统
- 操作系统的历史

从 Hello World 说起

```c
#include <stdio.h>

int main() {
    printf("Hello World!\n");
    return 0;
}
```

```
bash$ gcc hello.c -o hello

# 运行一个hello world程序
bash$./hello
Hello World!

# 同时启动两个hello world程序
bash$ ./hello & ./hello
[1] 144
Hello World!
Hello World!
[1]+ Done       ./hello
```

运行hello时，操作系统的作用？

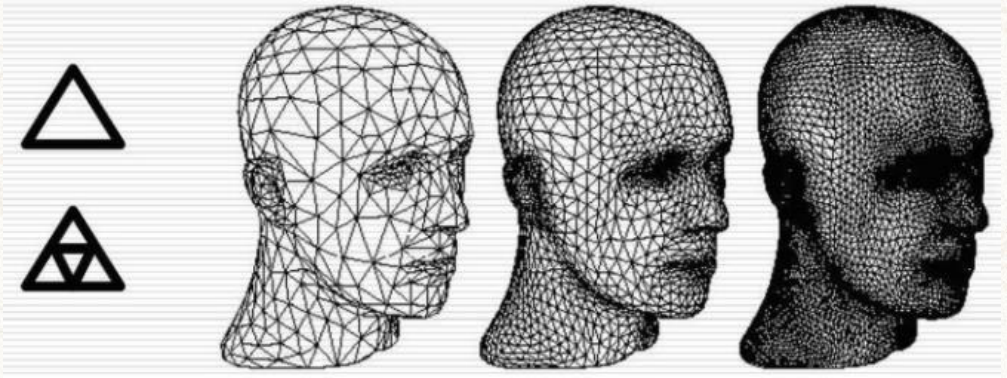
字符串打印

字符串打印

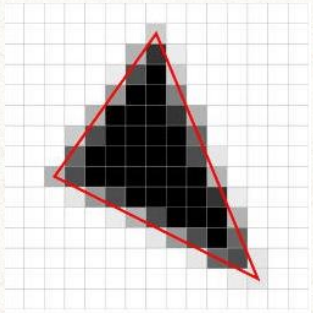
- `Hello World!` 怎么显示在电脑屏幕上?

复杂的图形:

模型三角化



光栅化



操作系统考虑的一些问题

- hello 这个可执行文件存储在什么位置？是如何存储的？
- hello 这个可执行文件是如何加载到 CPU 中运行？
- hello 这个可执行文件是如何将"Hello World!"这行字输出到屏幕？
- 两个hello 程序同时运行的过程中如何在一个 CPU 中运行？
- …

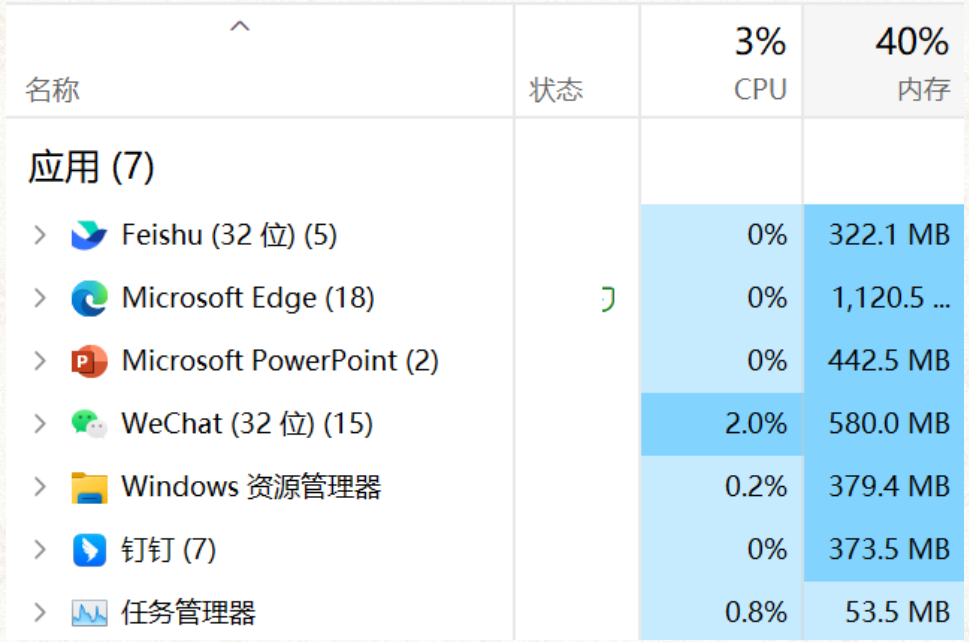
操作系统需要：1、服务应用；2、管理应用

操作系统为应用提供的一些服务

- 为应用提供计算资源的抽象
  - CPU：进程/线程，数量不受物理CPU的限制
  - 内存：虚拟内存，大小不受物理内存的限制
  - I/O设备：将各种设备统一抽象为文件，提供统一接口
- 为应用提供线程间的同步
  - 应用可以实现自己的同步原语（如spinlock）
  - 操作系统提供了更高效的同步原语（与线程切换配合，如pthread_mutex）
- 为应用提供进程间的通信
  - 应用可以利用网络进行进程间通信（如loopback设备）
  - 操作系统提供了更高效的本地通信机制（具有更丰富的语义，如pipe）
    - 例：Shell Pipe

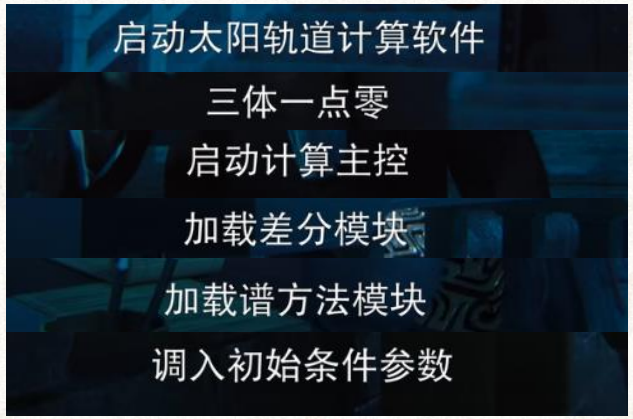
操作系统对应用的管理

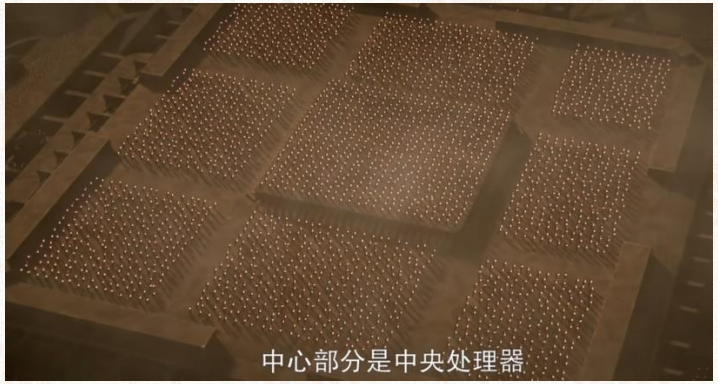
- 生命周期的管理
  - 应用的加载、迁移、销毁等操作
- 计算资源的分配
  - CPU：线程的调度机制
  - 内存：物理内存的分配
  - I/O设备：设备的复用与分配
- 安全与隔离
  - 应用程序内部：访问控制机制
  - 应用程序之间：隔离机制，包括错误隔离和性能隔离



《三体》电视剧第15集：人列计算机

- 28:38 - 工作内容：数值求解太阳运行的微分方程
- 30:01，32:14 - 门电路
- 38:00 - 计算机组成原理
- 38:29 - 有关操作系统部分：
- 41:29 - 应用程序



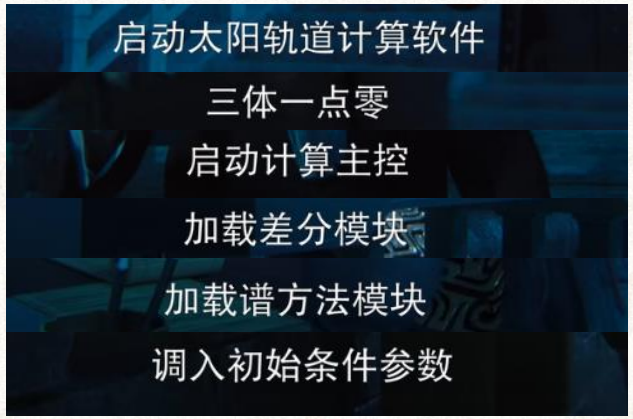


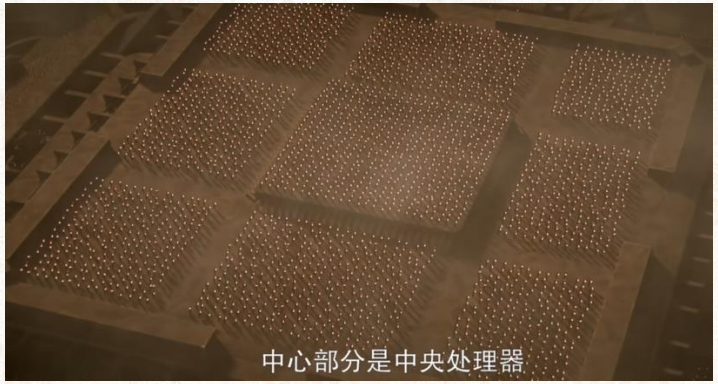
- 42:11 - 操作系统软件

https://v.qq.com/x/cover/mzc002007knmh3g/j0045k9gx33.html

《三体》电视剧第15集：人列计算机

- 28:38 - 工作内容：数值求解太阳运行的微分方程
- 30:01，32:14 - 门电路
- 38:00 - 计算机组成原理
- 38:29 - 有关操作系统部分：
- 41:29 - 应用程序





- 42:11 - 操作系统软件

问题来了：这个过程需不需要操作系统？

https://v.qq.com/x/cover/mzc002007knmh3g/j0045k9gx33.html

单选题 1分

设置

如果一台机器有且只有一个应用程序，开机后自动运行且不会退出，是否还需要操作系统?

A 需要

B 不需要

提交

单选题 1分

设置

如果一个应用希望自己完全控制硬件而不是使用操作系统提供的抽象，是否还需要操作系统?

A 需要

B 不需要

提交

操作系统 = 管理 + 服务

- 管理和服务的目标有可能存在冲突
  - 服务的目标：单个应用的运行效率最大化
  - 管理的目标：系统的资源整体利用率最大化
  - 例：单纯强调公平性的调度策略往往资源利用率低
    - 如细粒度的round-robin导致大量的上下文切换

操作系统的定义

- 操作系统的核心功能：
  - 将有限的、离散的资源，高效地抽象为无限的、连续的资源
- 从软件角度的定义：
  - 硬件资源虚拟化+管理功能可编程
- 从结构角度的定义：
  - 操作系统内核+系统框架

应用与操作系统的交互：系统调用

- 什么是系统调用？
  - 应用调用操作系统的机制，实现应用不能实现的功能

- 例如： printf() -> write()->sys_write()
  - write(1, "Hello World!\n", 13)

- 使应用调用操作系统的功能就像普通函数调用一样

Hello运行中的系统调用(strace)

```
execve("./hello.o", ["./hello.o"], 0x7ffc4ade0a00 /* 47 vars */) = 0
brk(NULL)                               = 0x55922c071000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd09149390) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=68663, ...}) = 0
mmap(NULL, 68663, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7efd7ae1a000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68,
880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7efd7ae18000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68,
880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7efd7ac26000
mprotect(0x7efd7ac4b000, 1847296, PROT_NONE) = 0
```

Hello运行中的系统调用(strace)

```
mmap(0x7efd7ac4b000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) =
0x7efd7ac4b000
mmap(0x7efd7adc3000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7efd7adc3000
mmap(0x7efd7ae0e000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) =
0x7efd7ae0e000
mmap(0x7efd7ae14000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7efd7ae14000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7efd7ae19540) = 0
mprotect(0x7efd7ae0e000, 12288, PROT_READ) = 0
mprotect(0x55922b152000, 4096, PROT_READ) = 0
mprotect(0x7efd7ae58000, 4096, PROT_READ) = 0
munmap(0x7efd7ae1a000, 68663)           = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL)                               = 0x55922c071000
brk(0x55922c092000)                     = 0x55922c092000
write(1, "Hello World!\n", 13Hello World!
)           = 13
exit_group(0)                           = ?
+++ exited with 0 +++
```

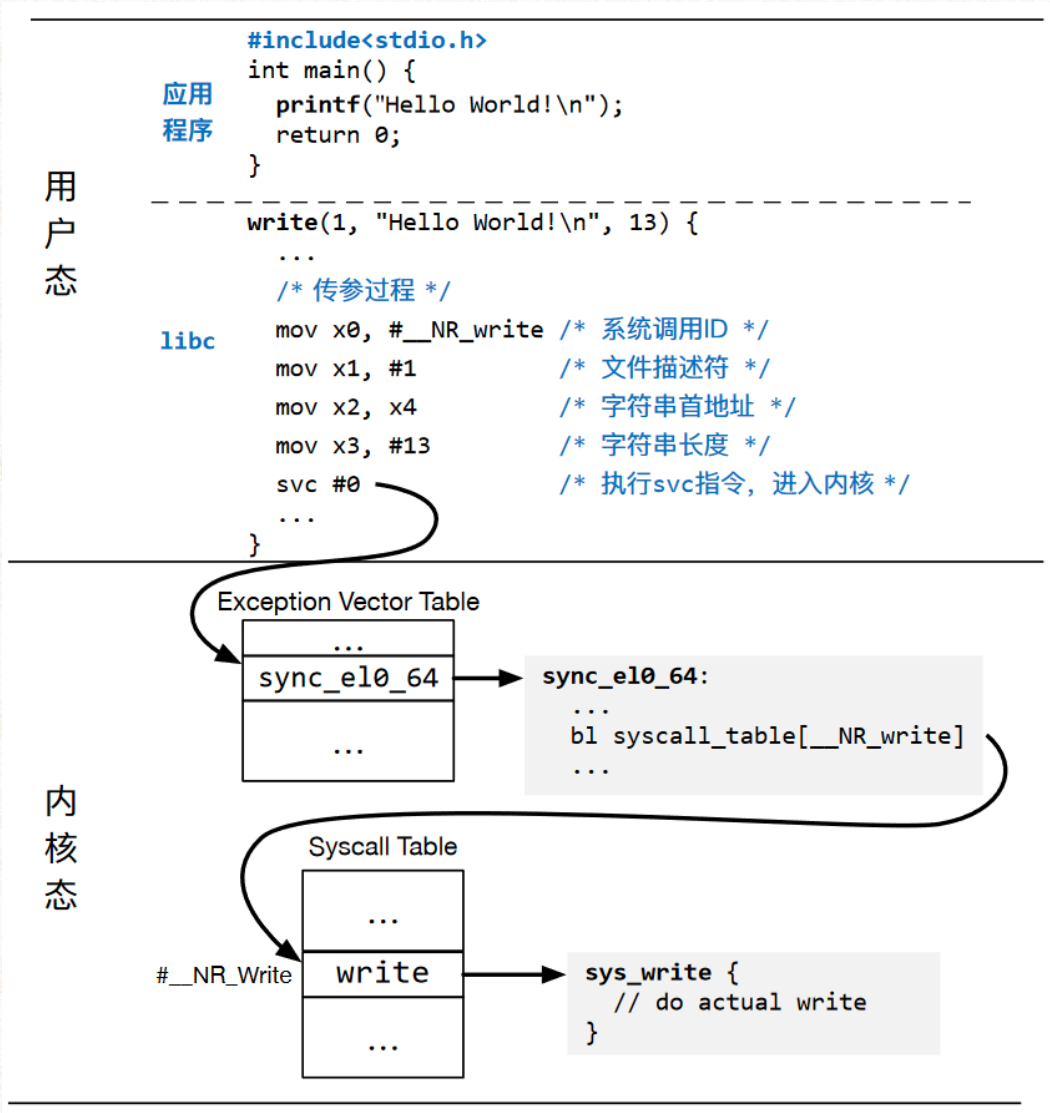
Hello运行中的系统调用(strace)

```
/* 运行hello程序 */
execve("./hello", ["./hello"], 0x7ffed5a79e80 /* 64 vars */) = 0
...

/*将``Hello World!\n''写到标准输出中，在这里，1代表标准输出，
  13代表一共写了13个字符。*/
write(1, "Hello World!\n", 13Hello World!) = 13

/* 执行结束后，hello程序退出*/
exit_group(0)
```

SVC 系统调用：使用异常向量表



操作系统的功能：管理

- 避免一个流氓应用独占所有资源

- 方法-1：每10ms发生一个时钟中断（时间片）
  - 调度器决定下一个要运行的任务

```
int main() {
    while (1);
}
```

- 方法-2：可通过信号等打断当前任务执行
  - 如：kill -9 1951

操作系统的功能：管理

- 如何卡死一个OS?
  - 例：可以fork出无数的进程

```
int main () {
    while (1) {
        fork();
    }
}
```

操作系统的功能：管理

- 如何卡死一个OS?
  - 例：可以fork出无数的进程

```
int main () {
    while (1) {
        fork();
    }
}
```

- 如何解决这个问题?
  - 资源配额：cgroup/Linux
  - 虚拟化：虚拟机
  - 万能方法：重启机器
  - 制度约束：AppStore的程序预审准入机制

大纲

- 为什么要学习以及怎么学习操作系统
- 什么是操作系统
- 操作系统的历史

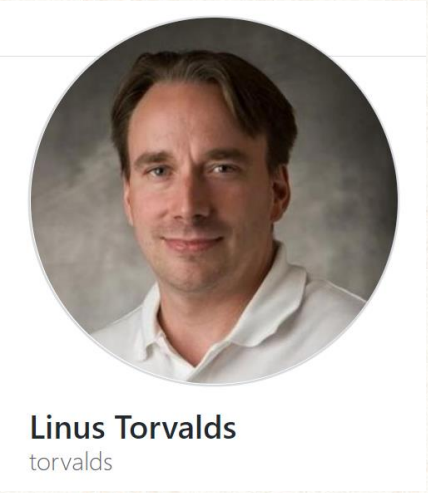
批处理操作系统：GM-NAA I/O

- Robert L. Patrick和Owen Mock于1956年建设
  - 运行在IBM 704上
  - 主要功能：批处理运行任务

通用操作系统：OS/360

- IBM System/360 OS，1964
  - 首个通用操作系统，首次将操作系统与计算机分离
  - 架构师：Gene Amdahl（Amdahl's Law)
  - 项目经理：Fred Brooks (《人月神话》，1999年图灵奖得主)
    - 开创软件工程学科

分时与多任务：Multics/Unix/Linux



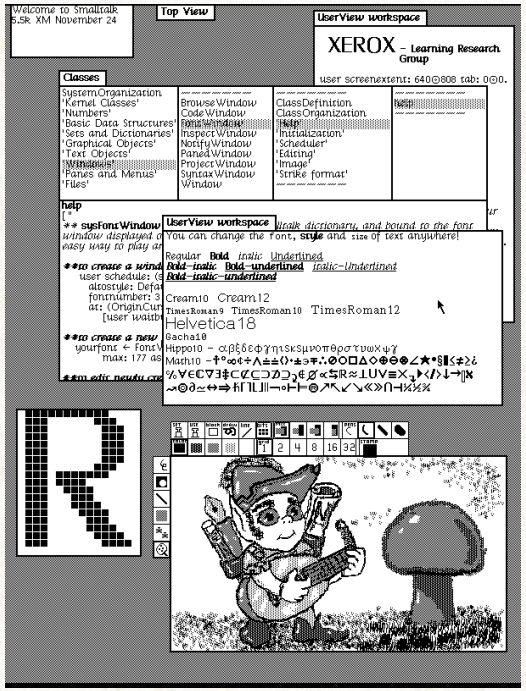
Multics: Fernando Corbató (1990年图灵奖) MIT/GE, 1964
: 分时，文件系统，动态链接等

Unix: Ken Thompson, Dennis Ritchie (1983年图灵奖), 1969
Shell，层次化文件系统

Linux: Linus Torvalds, 1991
最流行的开源操作系统

https://github.com/torvalds/linux

图形界面： Xerox Alto/MacOS/Windows



Xerox Alto （1973）：第一个图形化操作系统，首次使用鼠标 （ Chuck Thacker，2009年图灵奖）

Mac OS (Apple LISA, 1983):1979年乔布斯访问Xerox PARC，意识到GUI的重要性，买下了GUI进行研究

Windows 1.0 （1985）：基于图形界面的操作系统

https://zh.wikipedia.org/wiki/Xerox_Alto

主观题 10分

设置

关于这节课，有什么疑问?

作答

谢谢

微信：suyuxin
钉钉：苏玉鑫
B站：https://space.bilibili.com/502854403
软工集市课程专区：https://ssemarket.cn/new/course