

操作系统复习

蒋炎岩

南京大学 | 计算机软件研究所 | 系统与软件分析研究组



Hello World的一生



回到开学的第一天.....

- 你们学写的第一份代码，我们又回来了
 - Hello World从执行那一瞬间开始，从软件到硬件，究竟发生了什么？

```
#include <stdio.h>

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

- \$ gcc -o hello hello.c && ./hello



编译、链接

- [Q1] `gcc -o hello hello.c`做了什么？
- [Q2] `hello`是什么格式的文件？文件里有什么？



加载

- [Q3] ./hello回车为什么能执行Hello World程序？
- [Q4] ./hello执行的第一条指令是什么？
- [Q5] ./hello的main函数开始执行前，完成了什么？调用了哪些系统调用？



printf()

- [Q6] printf("Hello World\n")的printf实际代码位于何处？如何确定它的地址？
- [Q7] printf是线程安全的吗？如何保证printf()的线程安全？
- [Q8] 多个进程printf输出到终端/文件，会发生什么？
- [Q9] printf("Hello World")和printf("Hello World\n")有什么区别？



printf() – 进入库函数

- [Q10] hello程序执行过程中调用printf, 其中哪些操作是在用户态完成的、哪些必须进入内核态? printf使用哪些系统调用?
- [Q11] 在printf()和系统调用执行的过程中, memory layout是怎样的? %eip/%esp何时在哪些范围活动?



write系统调用：机制

- [Q12] 在x86系统上，write系统调用应该如何实现？

```
ssize_t write(int fd, const void *buf, size_t count);
```

- [Q13] 结合write系统调用理解 “操作系统不过是个C代码”
- [Q14] 同样一份代码(二进制文件)，为什么能输出到不同位置？
 - 输出到文件(./hello > a.txt)、终端(./hello)、磁盘(./hello > /dev/sda)
- [Q15] write如何知道要写入哪个文件系统、调用哪份代码、写入哪个物理设备？

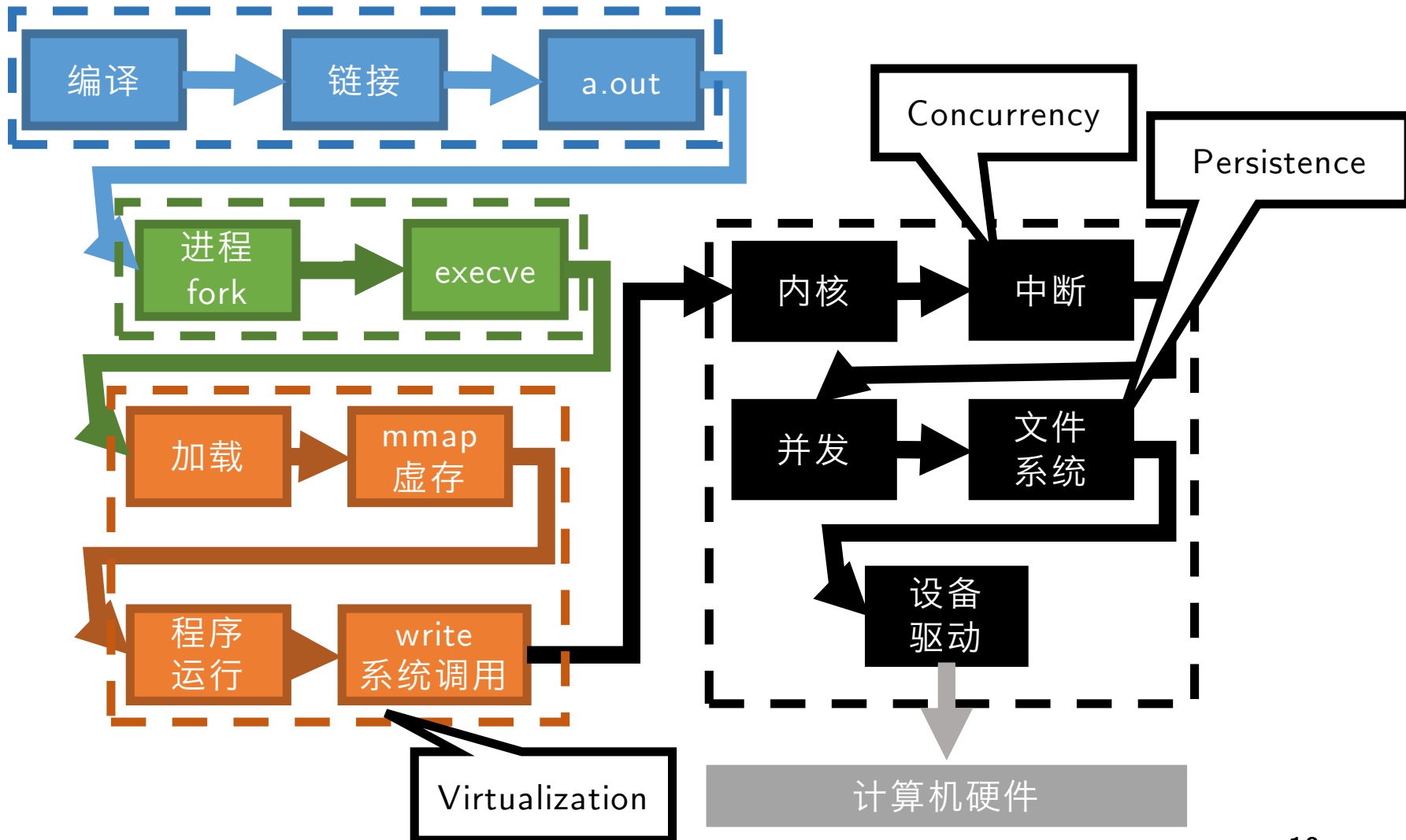


write系统调用：实现

- [Q16] 当write需要长时间等待(例如磁盘I/O)时，如何调度其他进程/线程？
- [Q17] 当系统中的多个线程都write同一个文件时，如何保证并发访问的正确性？
- [Q18] 仅使用write系统调用并不能保证数据的一致性，此时需要添加何种系统调用？



Operating Systems: Three Easy Pieces





Surprise! 就是这么简单

- 当你回想hello world执行的时候，整个操作系统就在你心中了
 - 所有的知识都按“为用户更好服务”的思路展开

