

2020 春

LINUX OPERATING SYSTEM

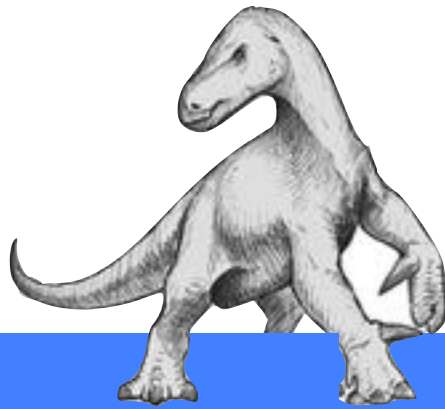
YANG

LINUX操作系统（双语）





双语课→课件内容中英混排



|Lecture 2

Operating-System Structures

本讲内容

 操作系统的服务

 为用户

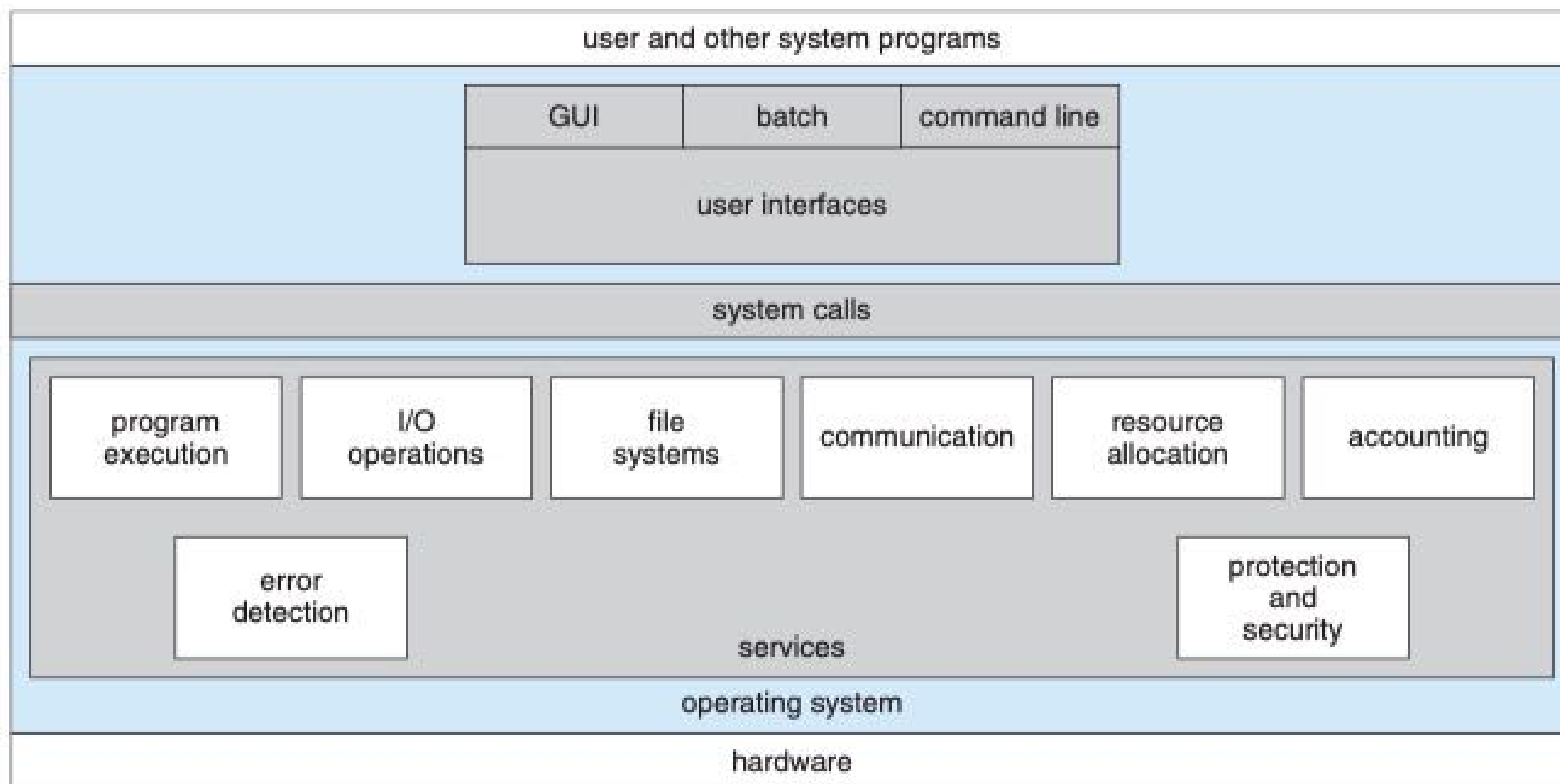
 为程序员

 操作系统的构建方式

 GNU/Linux

操作系统的服务

操作系统提供哪些服务？



USER INTERFACE

🧠 Almost all operating system have a user interface(UI). It offers a way for users to interface with OS.

🧠 CLI(Command Line Interface)

🧠 command interpreter(shell)

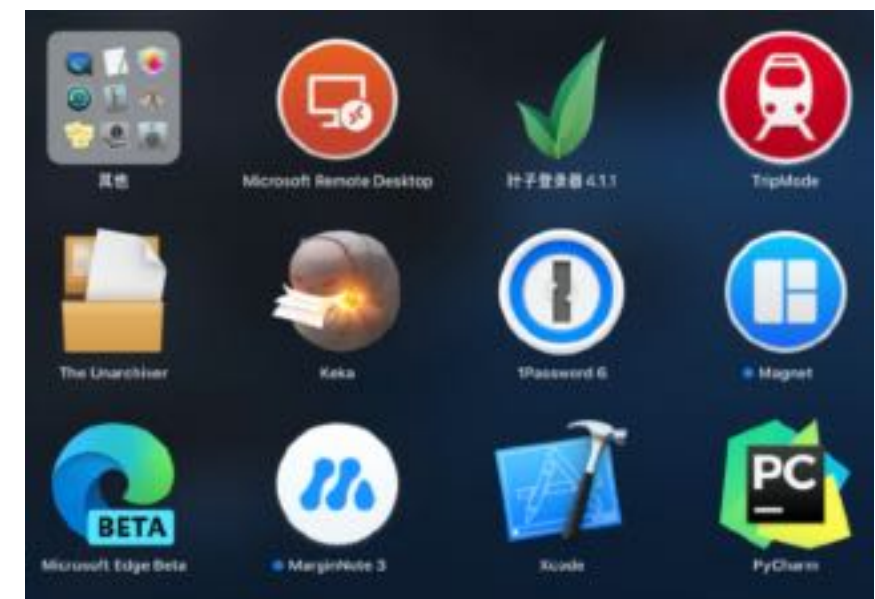
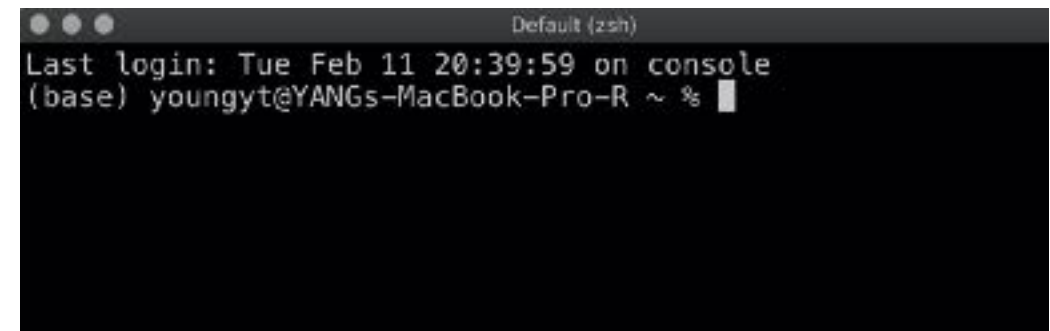
🧠 GUI(Graphic User Interface)

🧠 A user friendly graphical user interface.

🧠 Batch

🧠 It is a file which contains commands and directives.

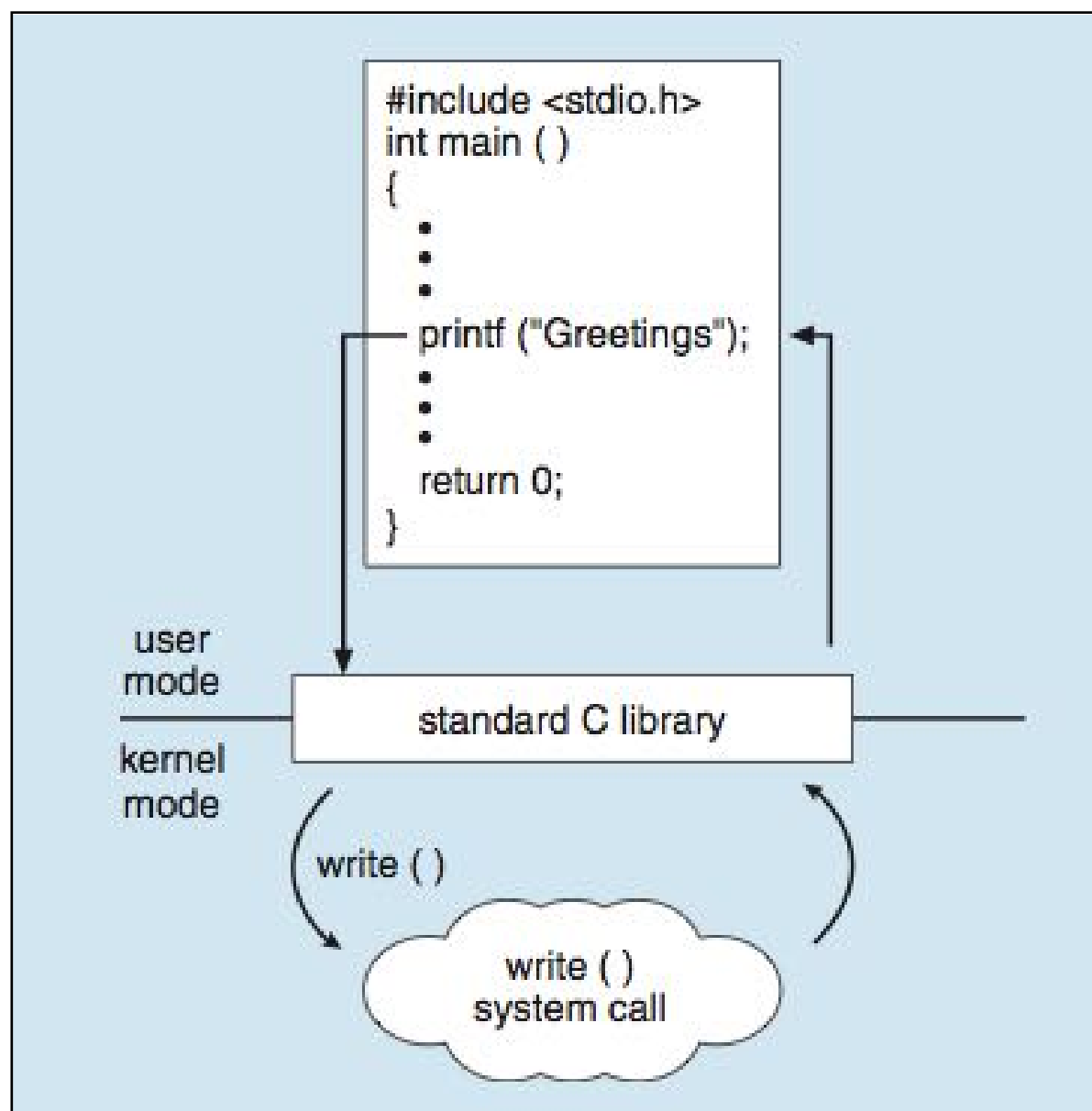
🧠 Demonstration 🙌 ...



系统调用 (SYSTEM CALLS)

- 💡 系统调用提供了访问和使用操作系统所提供的服务的接口。
- 💡 系统调用的实现代码是操作系统级的
- 💡 这个接口通常是面向程序员的
- 💡 API (Application Programming Interface): 指明了参数和返回值的一组函数。
- 💡 应用程序App的开发人员通过透过API间接访问了系统调用
- 💡 Windows API / POSIX API / JAVA API

标准C程序



- 💡 `printf`是standard C library提供的API
- 💡 `printf`函数的调用引发了对应的系统调用 `write`的执行
- 💡 `write`执行结束时的返回值传递回用户程序

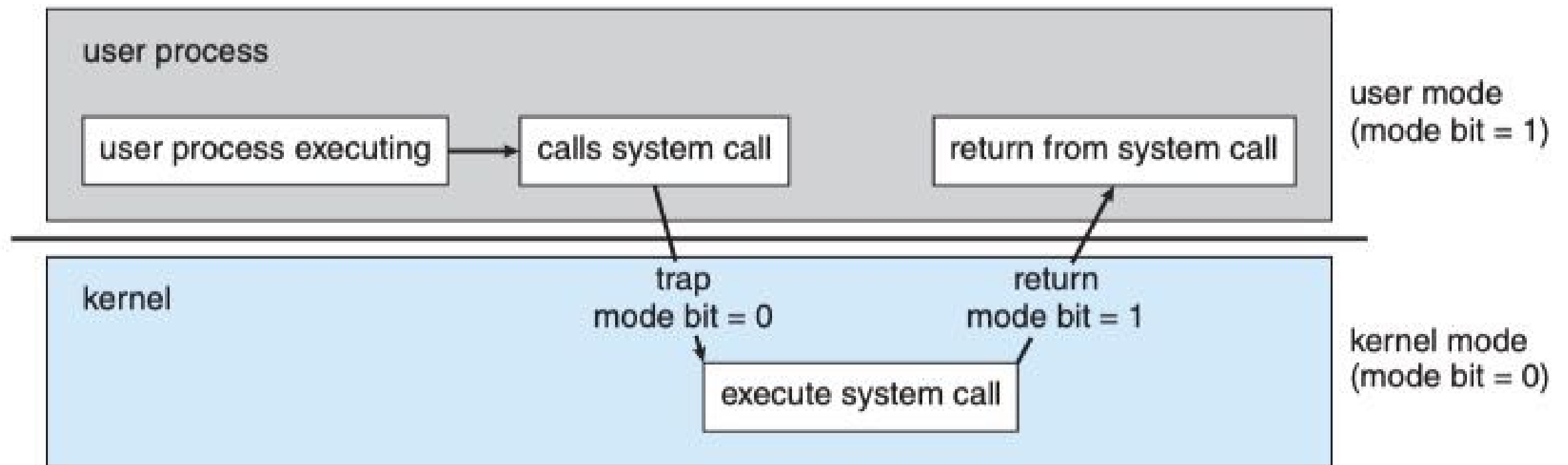
双重模式（DUAL MODE）

- 💡 现代计算机系统有一个特殊的硬件，用于划分系统的运行状态，至少需要两种单独运行模式：
 - 💡 用户模式（user mode）：执行用户代码
 - 💡 内核模式（kernel mode）：执行操作系统代码
- 💡 目的：确保操作系统正确的运行
- 💡 实现方式
 - 💡 用一个硬件模式位来表示当前模式：0表示内核模式，1表示用户模式。

运行模式的切换

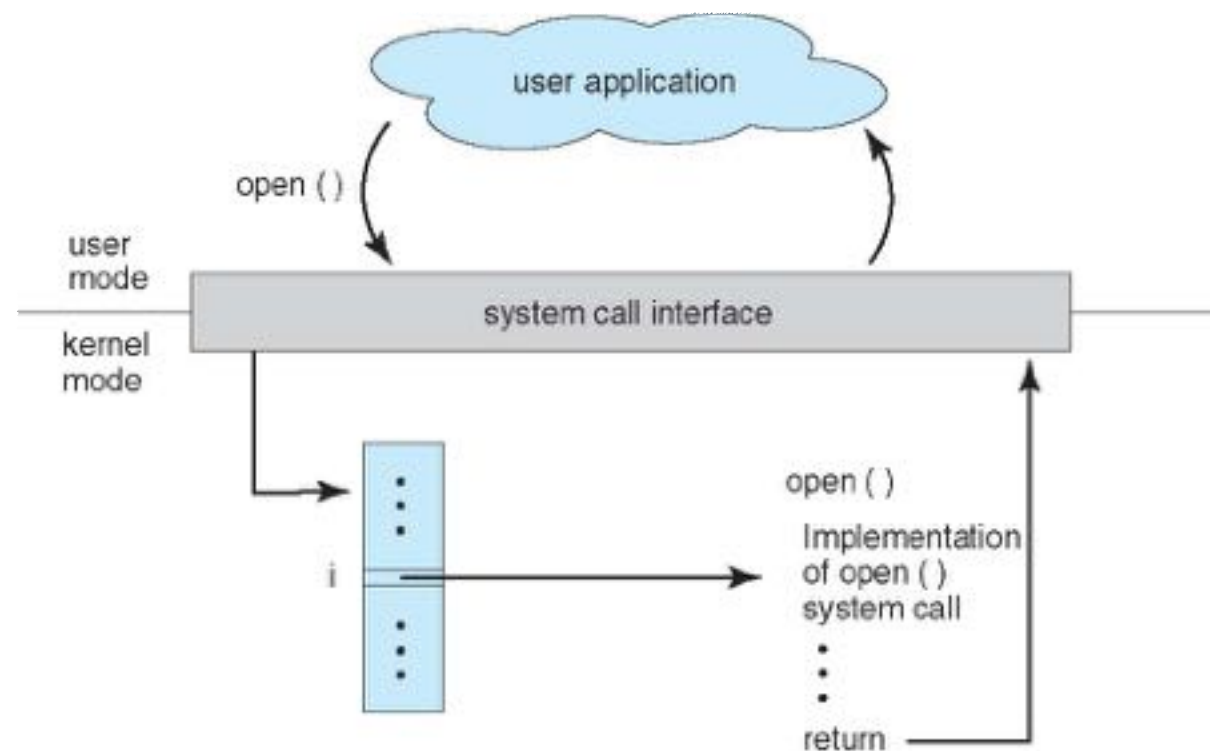
- 💡 系统调用需要在哪种模式下执行？
- 💡 你的应用程序运行在哪种模式下？
- 💡 调用API函数printf时，运行模式如何切换？

TRAP MECHANISM



系统调用的实现机制

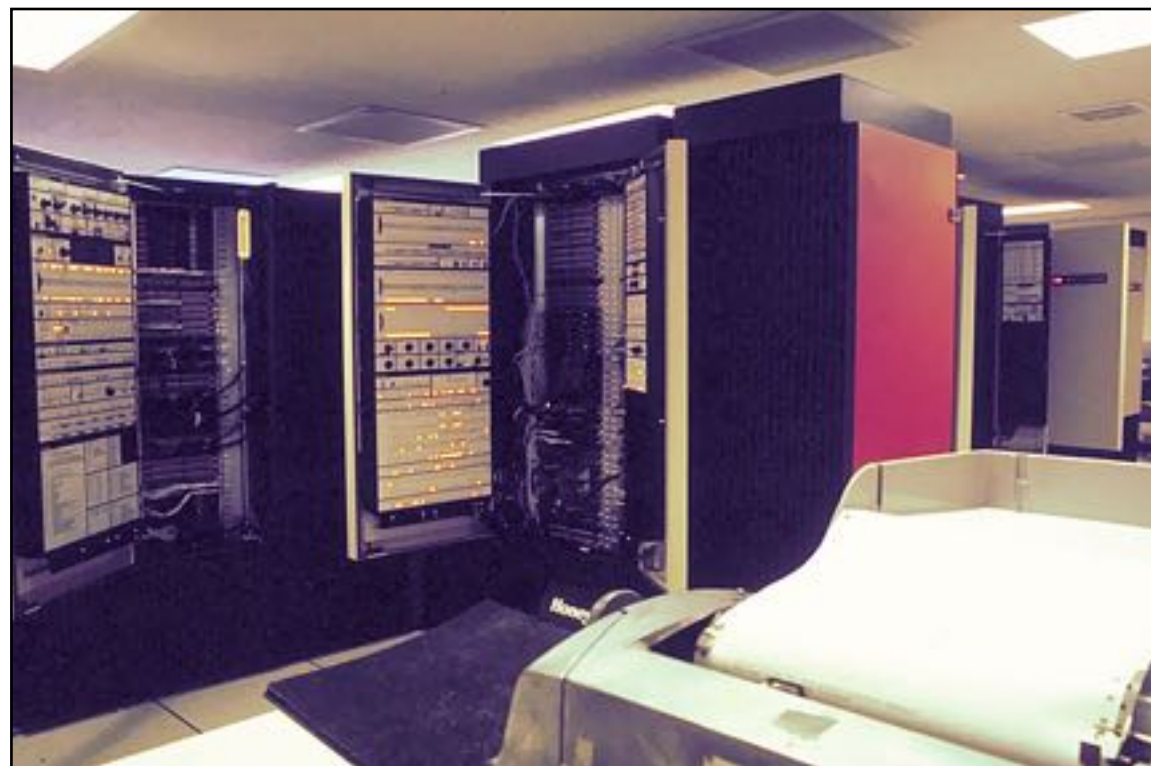
- 每个系统调用都有一个唯一的数字编号，被称为系统调用号。
- 用户代码调用API时，API中会向系统调用接口指明其所要用的系统调用号，操作系统内核中维护了一张索引表，依据这个调用号可以检索到访系统调用代码在内核中的位置。



操作系统的构建方式

MULTICS SYSTEM

- 由MIT、通用电气公司以及贝尔实验室三家于1969年推出了著名的**MULTICS**，即多路信息计算系统 (Multiplexed Information and Computer System)。作为一种通用的操作系统，MULTICS能有效地把计算机资源分配给多个远程用户程序，同时解决了安全和保密等问题。



操作系统设计成为一门学科

💡 多道程序设计的思想使用操作系统的复杂性变得难以控制:

💡 MULTICS:

💡 耗费了1866 people人力(www.multicians.org)

💡 图灵奖得主 Fernando Corbat ó 的获将演说: “On building systems that will fail”

💡 OS 360: 发布时就存在1000个已知bugs (APARs)

💡 “Anomalous Program Activity Report”

💡 操作系统设计逐渐形成一门重要的学科:

💡 如何处理硬件的复杂性?

💡 UNIX 基于Multics系统开发, 但已经大大的简化。

操作系统的设计思路

设计目标

 用户目标（user goal）

 系统目标（system goal）

机制与策略的分离

 机制（mechanism）：如何做

 策略（policy）：做什么

 微内核操作系统Mach、Darwin

GNU/LINUX

开源操作系统

🧠 Open-source, closed-source, hybrid operating system



🧠 Source code and binary code

Source code	Compiled binary code
<pre>\$VAT = 20; \$PRICE = 33; \$TOTAL = \$PRICE + (\$PRICE*\$VAT);</pre>	<pre>10000000000001010110010001010101010000000000 1010110010101001000100010011110010101000000 0000101011010001111001100001111000100000000 101100101000000000000000000000000000000000 00001010111000011001001010000000000010101110 00111000011010000000000010101110010010101010 10000000000010101110011010010001000100111100 1010100000000000101011101101011100000000000 00 01111001001010000000000010101111100110000110 1000000000001010111110110010000010000000000 1010111111010010000010000000000010101111111 0010000</pre>

🧠 What benefits does open source have?

历史

- 🧠 初衷：自由分享
- 🧠 阻碍：版权保护和商业利益
- 🧠 Richard Stallman于1983年设立GNU项目，旨在创建一个免费的、开源的、兼容Unix的操作系统。
- 🧠 GPL（GNU General Public License）：软件源码和二进制码一起发布，源码的任何修改应按同样的GPL许可来发布。
- 🧠 1990年第一个GNU核心Hurd完成开发
- 🧠 1991年Linus Torvalds开发出类Unix内核的Linux并于次年开源，GNU系统和Linux合并后成为今日的GNU/Linux

GNU/LINUX DISTRIBUTION

- 👤 **Linux Mint:** Requires low computer expertise, easy to install, easy to use and has a familiar-looking desktop for Windows users.
- 👤 **Debian:** For those seeking a truly free Linux distribution with no proprietary drivers, firmware or software, then Debian is for you. The grandfather of Linux.
- 👤 **Ubuntu:** A modern Linux distribution which is easy to install and easy to use.
- 👤 **openSUSE:** A stable and powerful Linux distribution. Not as easy to install as Mint and Ubuntu but a very good alternative nonetheless.
- 👤 **Fedora:** The most up to date Linux distribution with all new concepts incorporated at the earliest possible opportunity.
- 👤 **CentOS:** As with Fedora, CentOS is based on the commercial Linux distribution, Red Hat Linux. Unlike Fedora, it is built for stability.
- 👤 **Arch:** A rolling release distribution meaning that you don't have to install new versions of the operating system at any point because it updates itself. More difficult for the new user to get to grips with but very powerful.

准备实验环境和实验技能

🧠 环境安装

🧠 技能学习

🧠 常用命令：<https://zhuanlan.zhihu.com/p/36801617>

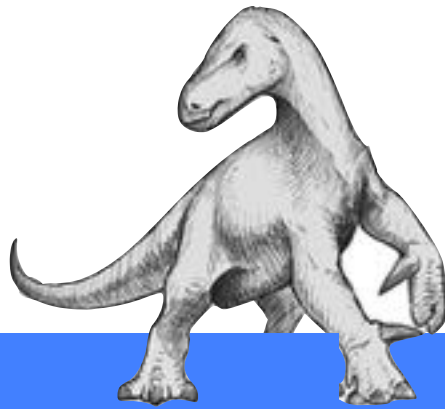
🧠 测试几个关键软件包是否正常

🧠 apt

🧠 gcc

🧠 预习教材第3章的3.1~3.3节

🧠 预告：下次上课时间为本周五（2.14）上午10：00



|Lecture 2

The End