

操作系统内核

- 基于Linux

第2讲

Linux操作系统内核实战环境搭建

主讲：杨文川

内容

- 1 Linux操作系统简介
- 2 Linux内核设计之道
- 3 Linux内核开发模式
- 4 Linux内核架构Overview
- 5 VMWare WorkStation的安装
- 6 利用VMWare虚拟机安装Linux
- 7 VMWare Tools的安装



内容导航：

1 Linux系统简介

Linux系统无处不在

- Linux基金会2017年发布的一组数据：
 - ✓ 90%的公有云应用在使用Linux系统
 - ✓ 62%的嵌入式市场在使用Linux系统
 - ✓ 99%的超级计算机在使用Linux系统
 - ✓ 82%的手机操作系统在使用Linux系统
 - ✓ 全球100万个顶级域名中超过90%都在使用Linux
 - ✓ 全球大部分的股票交易市场，都是基于Linux系统来部署的
 - ✓ 全球知名的淘宝网、亚马逊网、易趣网、沃尔玛等电子商务平台，都在使用Linux

Linux发展历史

- Linux诞生于1991年10月5日，它的产生和开源运动有着密切的关系。
 - ✓ 1983年，Richard Stallman发起GNU（GUN's Not UNIX）计划
 - ✓ 1991年，Linus Torvalds在一台386电脑上学习Minix操作系统，并动手实现了一个新的操作系统，然后在comp.os.minix新闻组上发布了第一个版本的Linux内核。
 - ✓ 1993年，Linux 0.99的代码已经有大约10万行。
 - ✓ 1994年，采用GPL（General Public License）协议的Linux 1.0正式发布
 - ✓ 1995年，Bob Young创办了Red Hat公司，以GNU/Linux为核心，把当时大部分的开源软件打包成一个发行版，这就是RedHat Linux发行版。
 - ✓ 1996年，Linux 2.0发布，该版本可以支持多种处理器，如alpha、mips、powerpc等，内核代码量大约是40万行。

- ✓ 1999年，Linux 2.2发布，支持ARM处理器。第一家国产Linux发行版——蓝点Linux系统诞生，它是第一个支持在帧缓冲上进行汉化的Linux中文版本。
- ✓ 2001年，Linux 2.4发布，支持对称多处理器SMP和很多外设驱动。同年，毛德操老师出版了《Linux 2.4内核源代码情景分析》。
- ✓ 2003年，Linux 2.6发布。与Linux 2.4相比，该版本增加了很多性能优化的新特性，使它成为**真正意义上的现代操作系统**。
- ✓ 2008年，谷歌正式发布Android 1.0，Android系统基于Linux内核来构建。在之后的十年里，Android系统占据了手机系统的霸主地位。
- ✓ 2011年，Linux 3.0发布。在长达8年的Linux 2.6开发期间，众多IT巨头持续为Linux内核，贡献了很多新特性和新的外设驱动。同年，全球最大的Linux发行版厂商Red Hat宣布营收达到10亿美元。
- ✓ 2015年，Linux 4.0发布。
- ✓ 2019年，Linux 5.0发布。

Linux发行版 – Red Hat

- Fedora Core
 - ✓ Fedora Core发行版是Red Hat公司的新技术测试平台，很多新的技术首先会应用到Fedora Core上，等测试稳定了才会加入Red Hat的RHEL版本中
- RHEL (Red Hat Enterprise Linux)
 - ✓ RHEL是面向服务器应用的Linux发行版，注重性能、稳定性和服务器端软件的支持。
 - ✓ 2018年4月Red Hat公司发布的RHEL 7.5操作系统，提升了性能，增强了安全性。
- CentOS (Community Enterprise Operating System)
 - ✓ 根据RHEL的源代码重新编译而成
 - ✓ CentOS不包含封闭源代码的软件
 - ✓ CentOS可以**免费使用**，并由社区主导

Linux发行版 – Debian系列

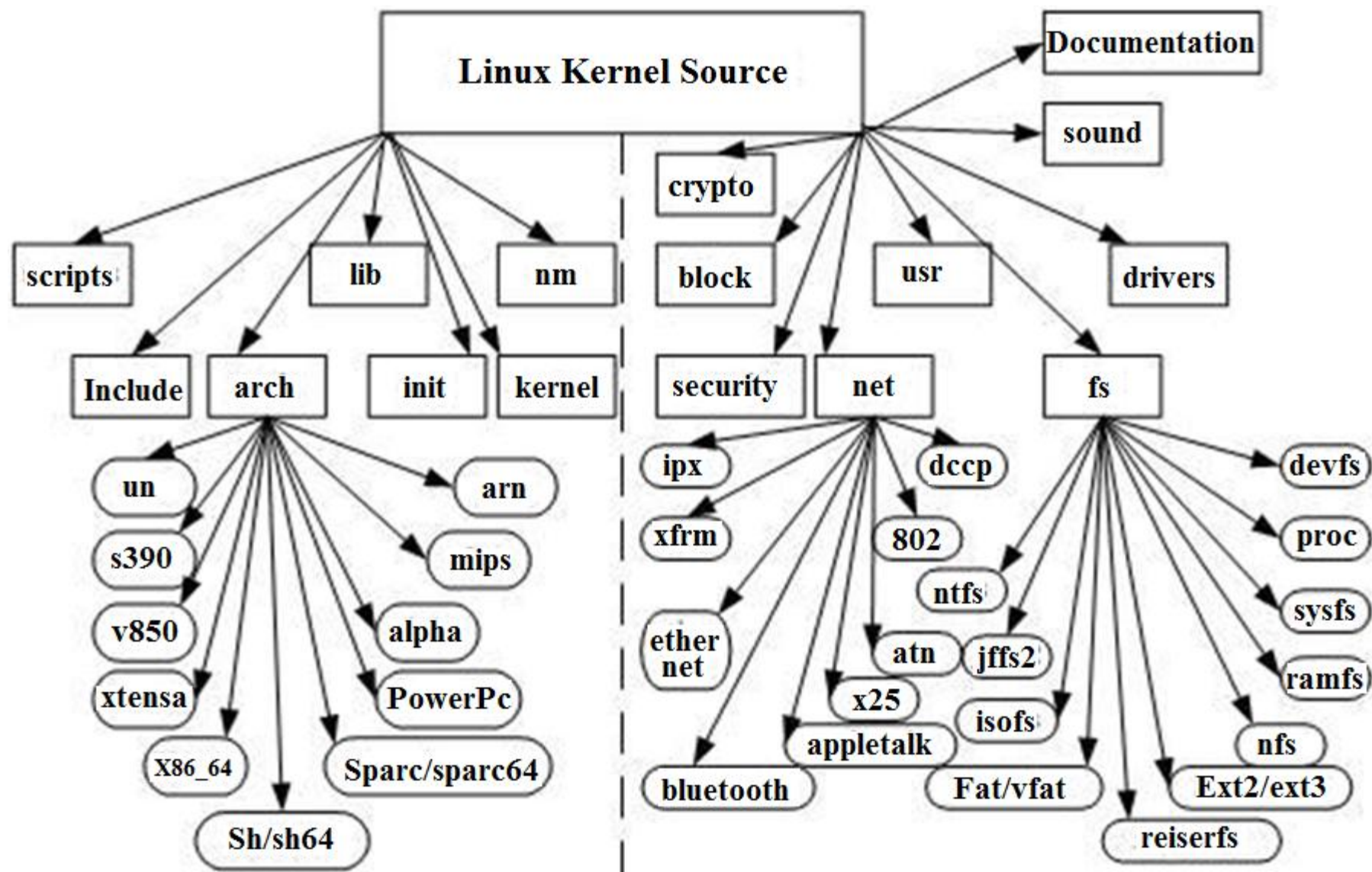
➤ Debian Linux

- ✓ Debian由Ian Murdock在1993年创建，是一个致力于创建自由操作系统的合作组织。因为Debian项目以Linux内核为主，所以Debian一般指的是Debian GNU/Linux。Debian能风靡全球的主要原因是其特有的apt-get/dpkg软件包管理工具，该工具被誉为所有Linux软件包管理工具中最强大、最好用的一个。

➤ Ubuntu Linux

- ✓ Ubuntu的中文音译是“乌班图”，是以Debian为基础打造的以桌面应用为主的Linux发行版。Ubuntu注重提高桌面的可用性以及安装的易用性等方面，因此经过这几年的发展，Ubuntu已经成为最受欢迎的桌面Linux发行版之一。

Linux内核源码目录结构

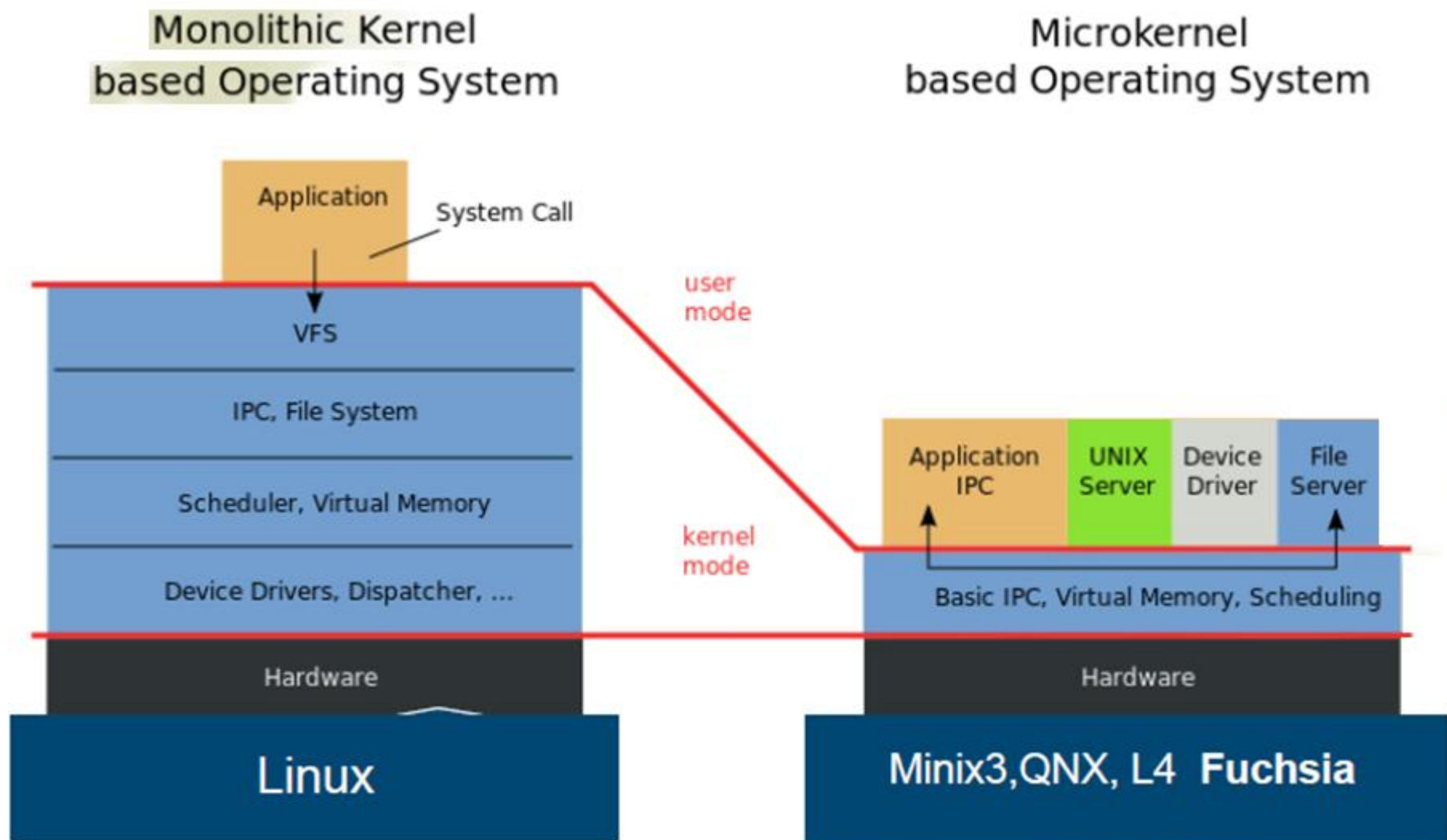




2 Linux内核结构及内核模块

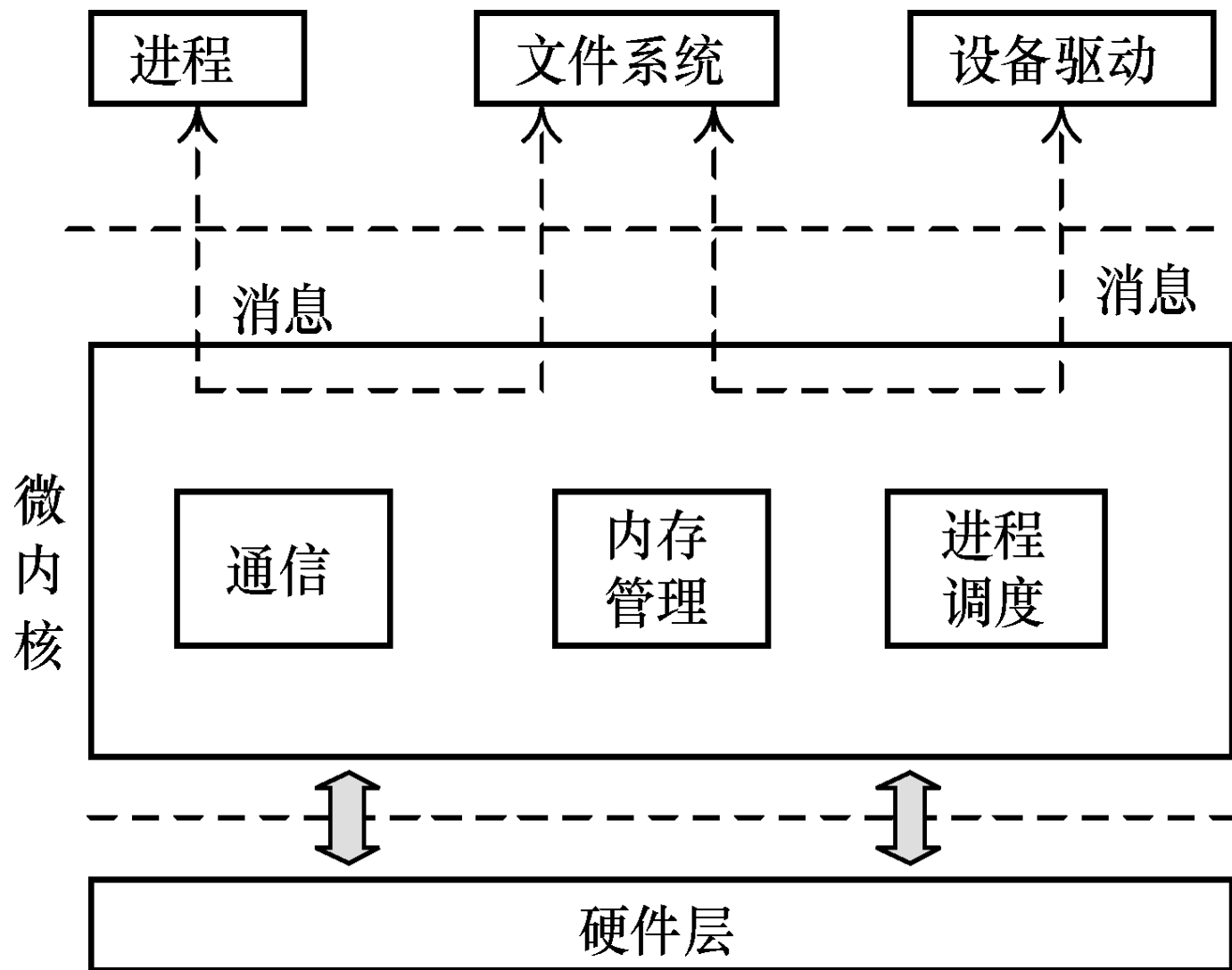
宏内核与微内核

- 操作系统内核的设计，在历史上存在两大阵营：
- 一个是宏内核，
- 一个是微内核



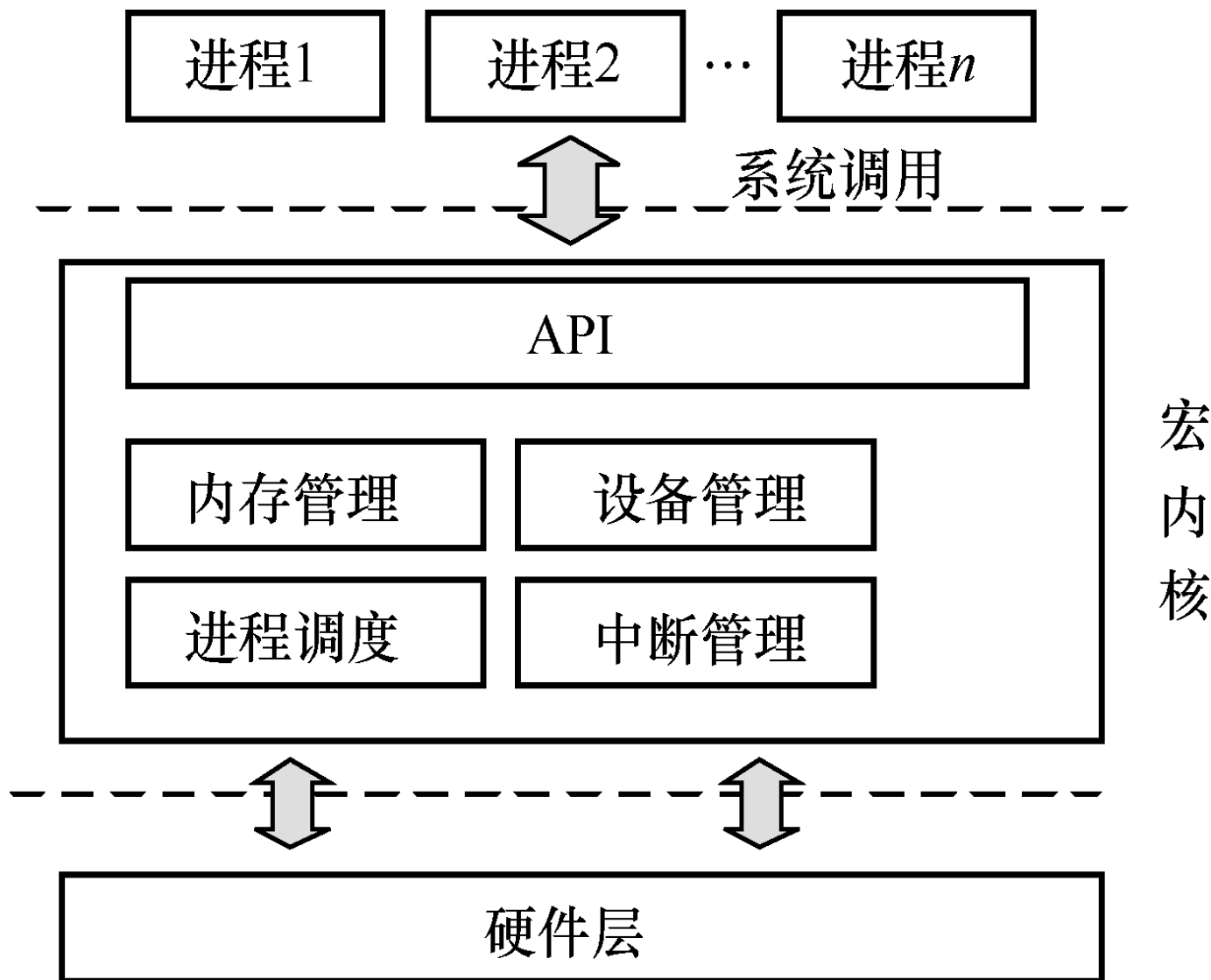
微内核

- 微内核：把操作系统分成多个独立的功能模块，
- 每个功能模块之间的访问，需要通过消息来完成



宏内核

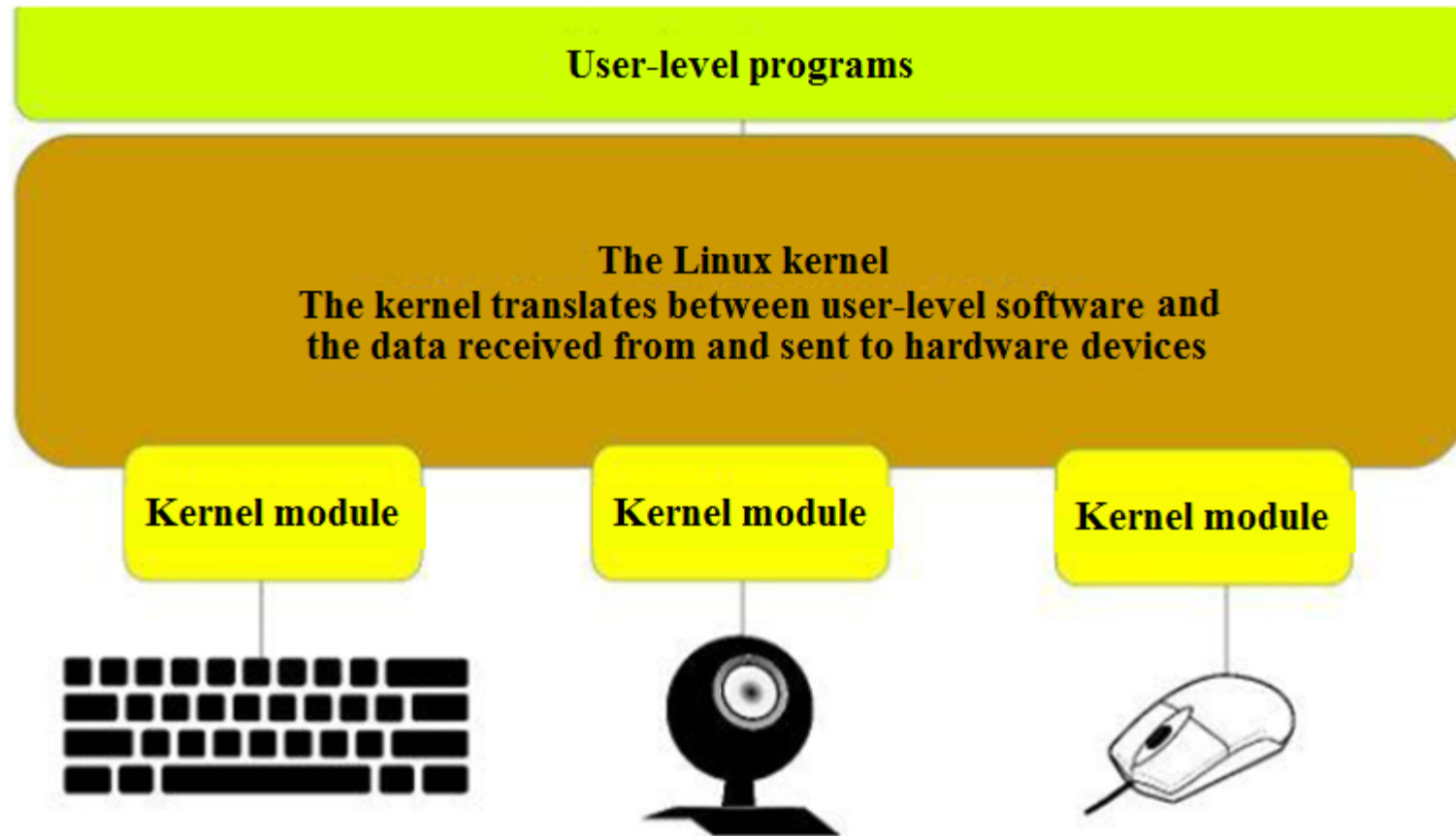
- 宏内核：指所有的内核代码都编译成一个二进制文件，
- 所有的内核代码，都运行在一个大内核地址空间里，
- 内核代码可以直接访问和调用，效率高并且性能好。



Linux设计之道

- Linus在设计之初，并没有使用当时学术界流行的微内核架构，而是采用实现方式比较简单的宏内核架构。
 - ✓ Linux在当时是业余作品
 - ✓ Linus本人更喜欢宏内核的设计
 - ✓ 宏内核架构的优点，是设计简洁和性能比较好
 - ✓ 微内核架构最大的问题，是高度模块化带来的交互的冗余和效率的损耗
- 所有的理论设计，放到现实的工程实践中都是一种折中的艺术。
- Linux内核融合了宏内核和微内核的优点。
 - ✓ 模块化设计
 - ✓ 抢占式内核、
 - ✓ 动态加载内核模块

可加装的Linux内核模块(LKM)



- A loadable kernel module (LKM) is a mechanism for adding code to, or removing code from, the Linux kernel at run time.

Linux内核模块编程入门

```
1 #include <linux/module.h>
2 #include <linux/kernel.h>
3 #include <linux/init.h>
4 /*
5  * 模块的初始化函数lkp_init()
6  * __init是用于初始化的修饰符
7  */
8 static int __init lkp_init(void)
9 {
10     printk("<1>Hello,world!from the kernel space...\n");
11     return 0;
12 }
13 /*
14  * 模块的退出和清理函数lkp_exit()
15  */
16 static void __exit lkp_exit(void)
17 {
18     printk("<1>Goodbye,world!leaving kernel space...\n");
19 }
20
21 module_init(lkp_init);
22 module_exit(lkp_exit);
23 /*
24  * 模块的许可证声明GPL
25  */
26 MODULE_LICENSE("GPL");
```

任何模块都要包含的三个头文件：

#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/init.h>

说明：

module.h头文件包含了对模块的版本控制；

kernel.h包含了常用的内核函数；

init.h包含了宏__init和__exit，宏__init告诉编译程序相关的函数和变量仅用于初始化，编译程序将标有__init的所有代码存储到特殊的内存段中，初始化结束就释放这段内存。

在此使用了printk()函数，该函数是由内核定义的，功能和C库中的printf()类似，它把要打印的日志输出到终端或系统日志。代码见samplec/15module/helloworld.c和Makefile

Linux内核模块编程入门

- 内核模块的Makefile文件

```
1 obj-m:=module_example.o           #产生module_example模块的目标文件
2 CURRENT_PATH := $(shell pwd)      #模块所在的当前路径
3 LINUX_KERNEL := $(shell uname -r) #linux内核源代码的当前版本
4 LINUX_KERNEL_PATH := /usr/src/linux-headers-$(LINUX_KERNEL)
5                                   #linux内核源代码的绝对路径
6 all:
7     make -C $(LINUX_KERNEL_PATH) M=$(CURRENT_PATH) modules #编译模块
8 clean:
9     make -C $(LINUX_KERNEL_PATH) M=$(CURRENT_PATH) clean  #清理模块
```

- 内核模块不是独立的可执行文件，但在运行时其目标文件被链接到内核中，只有超级用户才能加载和卸载模块。
- `obj-m :=`这个赋值语句的含义是说明要使用目标文件`module_example.o`建立一个模块，最后生成的模块名为`module_example.ko`。
- `.o`文件是经过编译和汇编，而没有经过链接的中间文件。
- 注：makefile文件中，若某一行是命令，则它必须以一个Tab 键开头。

Linux内核模块编程入门

- 模块插入命令：`$insmod module_example.ko`
- 模块删除命令：`$rmmod module_example`
- 查看模块信息的命令：`$dmesg`
- 动手写一个自己的Linux内核模块，
- 代码在samplec/15module中



Linux内核模块与C应用的对比

	C语言应用程序	内核模块程序
使用函数	Libc库	内核函数
运行空间	用户空间	内核空间
运行权限	普通用户	超级用户
入口函数	main()	module_init ()
出口函数	exit()	module_cleanup()
编译	gcc -c	make
连接	gcc	insmod
运行	直接运行	insmod
调试	gdb	kdebug, kdb, kgdb等



3 Linux内核开发模式

Linux内核开发模式

- Linux内核的开发模式完全由社区来主导。
 - ✓ Linus是Linux内核的最大的维护者和导演
 - ✓ 每个子模块都有各自维护者
 - ✓ 在邮件列表为讨论主战场，比如LKML

Kernel Version	Release Date	Days of Development
3.19	2015-02-08	63
4.0	2015-04-12	63
4.1	2015-06-21	70
4.2	2015-08-30	70
4.3	2015-11-01	63

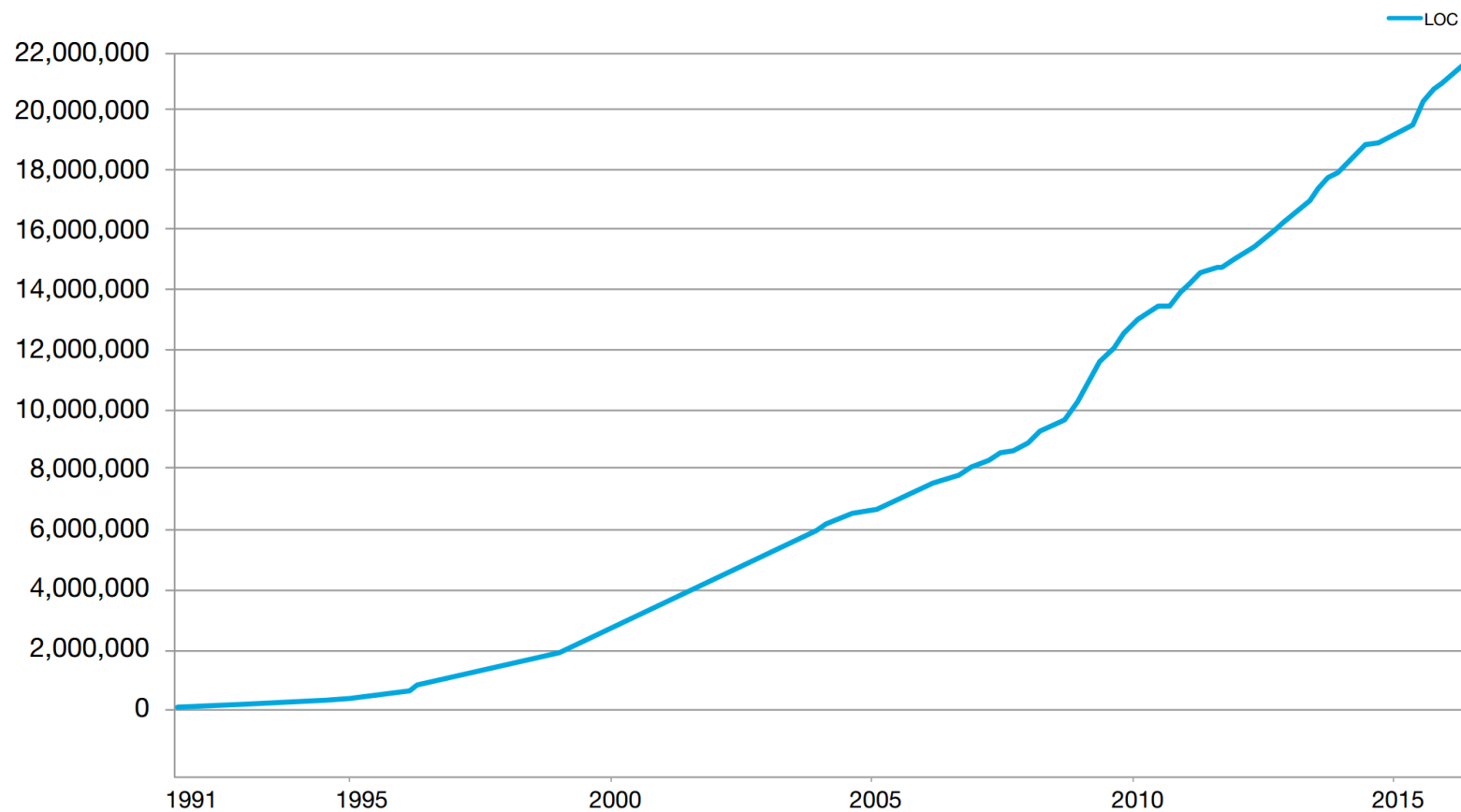
Kernel Version	Release Date	Days of Development
4.4	2016-01-10	70
4.5	2016-03-13	63
4.6	2016-05-15	63
4.7	2016-07-24	70

大约60-70天一个新内核版本发布

Linux内核行数的变化

- Linux内核从1991年的1w行代码发展到超过200w行代码

Total Lines of Code in the Linux Kernel



Linux内核开发参与者

- Linux内核参与的开发者分布全球，数以千计的开发者，几个顶尖的IT公司

Kernel Version	Developers	Companies
3.19	1,451	230
4.0	1,458	214
4.1	1,539	238
4.2	1,591	251
4.3	1,625	211

Kernel Version	Developers	Companies
4.4	1,575	220
4.5	1,537	231
4.6	1,678	243
4.7	1,582	221

主要的内核开发贡献者

Those developers are:

Name	Changes	Percent
H Hartley Sweeten	5,960	1.1%
Al Viro	5,433	1.0%
Takashi Iwai	4,723	0.8%
Mark Brown	3,960	0.7%
David S. Miller	3,950	0.7%
Mauro Carvalho Chehab	3,943	0.7%
Tejun Heo	3,852	0.7%
Johannes Berg	3,707	0.7%
Russell King	3,467	0.6%
Thomas Gleixner	3,233	0.6%
Hans Verkuil	3,119	0.6%
Greg Kroah-Hartman	3,117	0.6%
Ingo Molnar	2,873	0.5%
Joe Perches	2,778	0.5%
Christoph Hellwig	2,697	0.5%

Name	Changes	Percent
Eric Dumazet	2,633	0.5%
Axel Lin	2,604	0.5%
Dan Carpenter	2,562	0.5%
Geert Uytterhoeven	2,460	0.4%
Laurent Pinchart	2,381	0.4%
Alex Deucher	2,340	0.4%
Bartlomiej Zolnierkiewicz	2,279	0.4%
Trond Myklebust	2,269	0.4%
Paul Mundt	2,268	0.4%
Daniel Vetter	2,224	0.4%
Ben Skeggs	2,216	0.4%
Arnd Bergmann	2,199	0.4%
Lars-Peter Clausen	2,176	0.4%
Arnaldo Carvalho de Melo	2,107	0.4%
Ralf Baechle	2,097	0.4%

主要的参与公司

The most active companies over the 3.19 to 4.7 development cycles were:

Company	Changes	Percent
Intel	14,384	12.9%
Red Hat	8,987	8.0%
none	8,571	7.7%
unknown	7,582	6.8%
Linaro	4,515	4.0%
Samsung	4,338	3.9%
SUSE	3,619	3.2%
IBM	2,995	2.7%
consultants	2,938	2.6%
Renesas Electronics	2,239	2.0%
Google	2,203	2.0%
AMD	2,100	1.9%
Texas Instruments	1,917	1.7%
ARM	1,617	1.4%
Oracle	1,528	1.4%

Company	Changes	Percent
Outreachy	1,524	1.4%
Vision Engraving Systems	1,456	1.3%
Free Electrons	1,453	1.3%
NXP Semiconductors	1,445	1.3%
Mellanox	1,404	1.3%
Atmel	1,362	1.2%
Broadcom	1,237	1.1%
NVidia	1,146	1.0%
Code Aurora Forum	1,033	0.9%
Imagination Technologies	963	0.9%
Huawei Technologies	937	0.8%
Facebook	877	0.8%
Pengutronix	790	0.7%
Cisco	692	0.6%
Qualcomm	656	0.6%



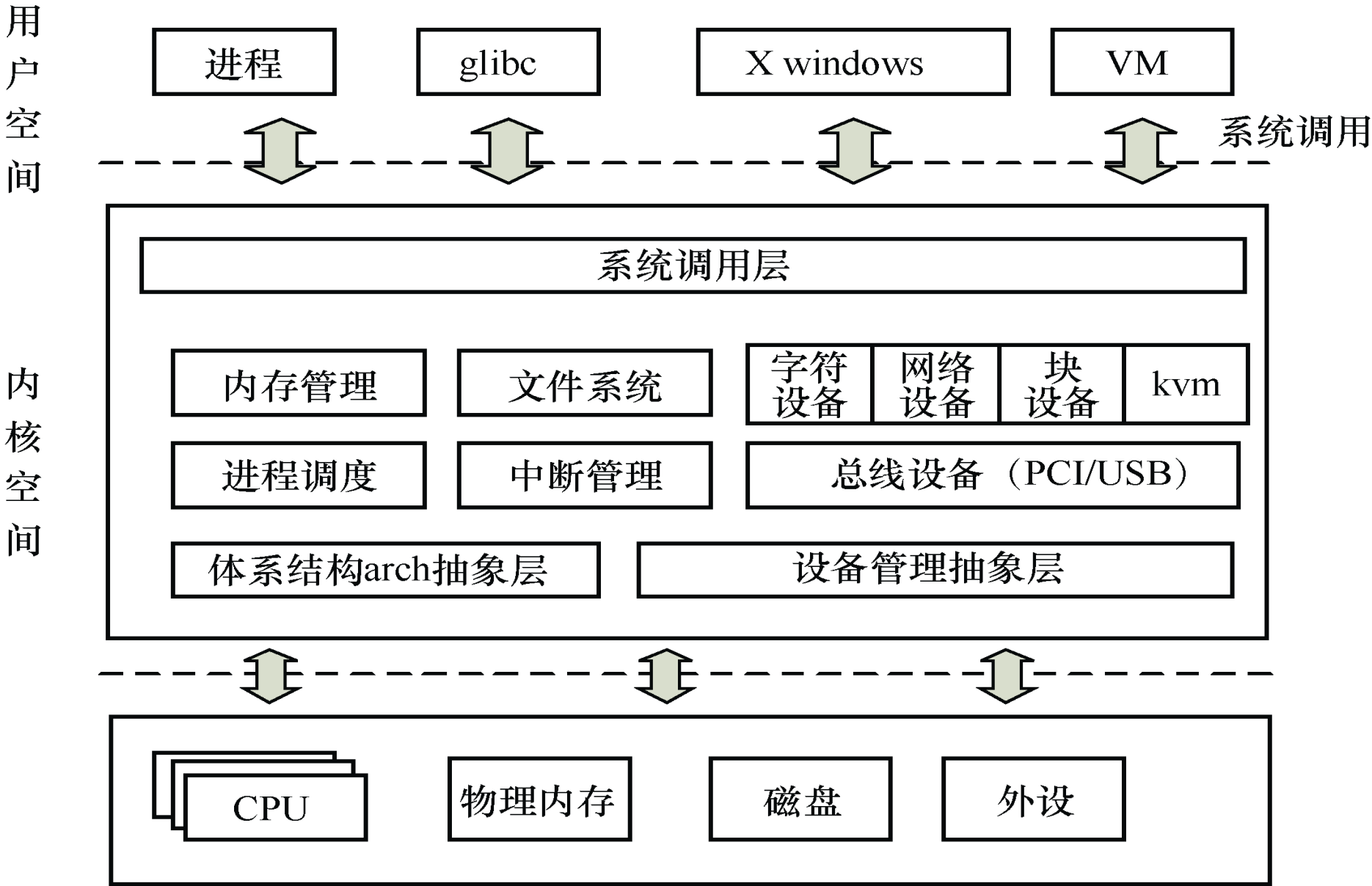
内容导航：

4 Linux内核OverView

Linux内核架构概貌

Linux内核把系统分成两个空间：
用户空间和内核空间

下面介绍其中各层



➤ 系统调用层

- ✓ Linux内核为内核态和用户态之间的切换，设置了软件抽象层，叫作系统调用（System Call）层，
- ✓ 其实每个处理器体系结构设计中，都提供了一些特殊的指令，来实现内核态和用户态之间的切换。
- ✓ Linux内核充分利用了这种硬件提供的机制，来实现系统调用层。
- ✓ 系统调用层最大的目的，是让用户进程看不到真实的硬件信息。
- ✓ 当用户需要读取一个文件的内容时，编写用户进程的程序员，不需要知道这个文件具体存放在磁盘的哪个扇区里，只需要调用open()、read()或mmap()等函数即可。

处理器体系结构抽象层

- ✓ Linux最初的设计只支持x86体系结构，后来不断扩展，到现在已经支持几十种体系结构
- ✓ Linux内核为不同体系结构的实现，做了很好的抽象和隔离，也提供了统一的接口来实现。
- ✓ 比如，在内存管理方面，Linux内核把和体系结构相关部分的代码，都存放在arch/xx/mm目录里，把和体系结构不相关的代码，都存放在mm目录里，从而实现完好的分层。

进程管理

- ✓ 进程是现代操作系统中非常重要的概念，包括上下文切换（Context Switch）以及进程调度（Scheduling）
- ✓ 每个进程运行时，都感觉完全占有了全部的硬件资源，但是进程不会长时间占有硬件资源
- ✓ 操作系统利用进程调度器，让多个进程并发执行
- ✓ **注意：** Linux内核并没有严格区分进程和线程，而常用task_struct数据结构来描述
- ✓ Linux内核的调度器的发展经历了好几代，从很早的 $O(n)$ 调度器，到Linux 2.6内核中的 $O(1)$ 调度器，再到现在的CFS公平算法调度器
- ✓ 目前比较热门的讨论是关于性能和功耗的优化，比如ARM阵营提出了大小核体系结构，至今在Linux内核实现中还没有体现，因此类似EAS（Energy Awareness Scheduling）这样的调度算法是一个研究热点

内存管理

- ✓ 内存管理模块是Linux内核中最复杂的模块，它涉及物理内存的管理和虚拟内存的管理
- ✓ 虚拟内存的模块有：反向映射、页面回收、KSM (Kernel Samepage Merging)、mmap映射、缺页中断、共享内存、进程虚拟地址空间管理等
- ✓ 物理内存管理包括：物理内存初始化，页面分配器（Page Allocator），伙伴系统，slab分配器等等。

中断管理

- ✓ 中断管理包含处理器的异常（Exception）处理和中断（Interrupt）处理
- ✓ **异常**通常是指，如果处理器在执行指令时检测到一个反常条件，处理器就必须暂停下来处理这些特殊的情况，如常见的缺页异常（Page Fault）
- ✓ **中断**异常一般是指，外设通过中断信号线路来请求处理器，处理器会暂停当前正在做的事情来处理外设的请求
- ✓ Linux内核在**中断管理**方面有上半部和下半部之分。
 - ✓ **上半部**是在关闭中断的情况下执行的，因此处理时间要求短、平、快；
 - ✓ **下半部**是在开启中断的情况下执行的，很多对执行时间要求不高的操作，可以放到下半部来执行。
 - ✓ Linux内核为下半部提供了多种机制，如软中断、Tasklet和工作队列等。

设备管理

- ✓ 设备管理，对于任何的一个操作系统来说都是重中之重。
- ✓ Linux内核之所以这么流行，就是因为它支持的外设，是所有开源操作系统中最多的。
- ✓ 每当大公司有新的芯片诞生时，第一个要支持的操作系统是Linux，也就是尽可能地在Linux内核社区里推送。
- ✓ Linux内核的设备管理是一个很广泛的概念，包含的内容很多，如：
- ✓ ACPI、设备树、设备模型kobject、设备总线（如PCI总线）、字符设备驱动、块设备驱动、网络设备驱动等。

文件系统

- ✓ 为了支持各种各样的文件系统，Linux抽象出了一个称为**虚拟文件系统**（VFS）层的软件层，这样Linux内核就可以很方便地集成多种文件系统。
- ✓ 一个优秀的操作系统必须包含优秀的文件系统，但是文件系统有不同的应用场合，如基于闪存的文件系统F2FS、基于磁盘存储的文件系统ext4和XFS等

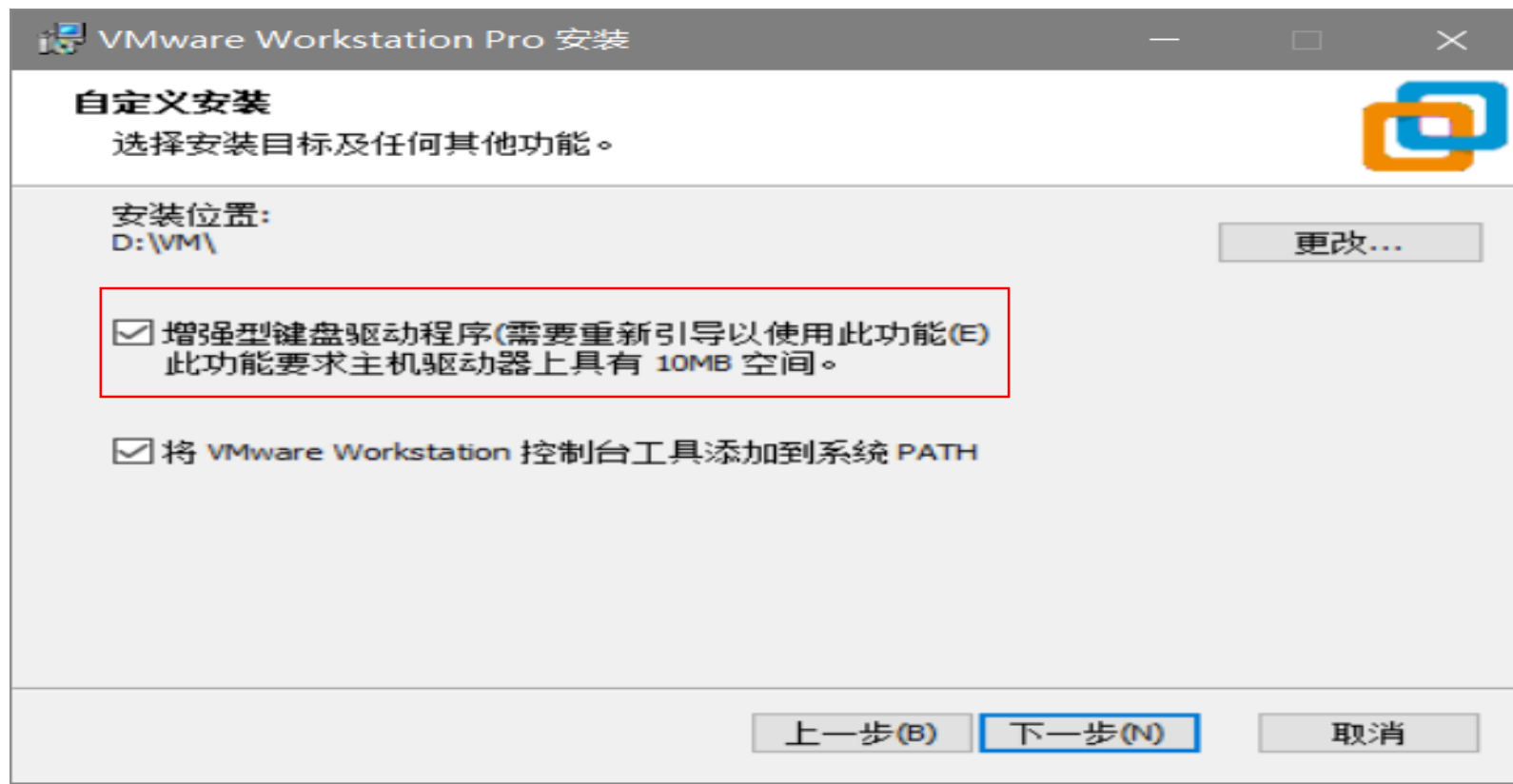


内容导航：

5 VMWare WorkStation的安装

VMWare WorkStation的下载和安装

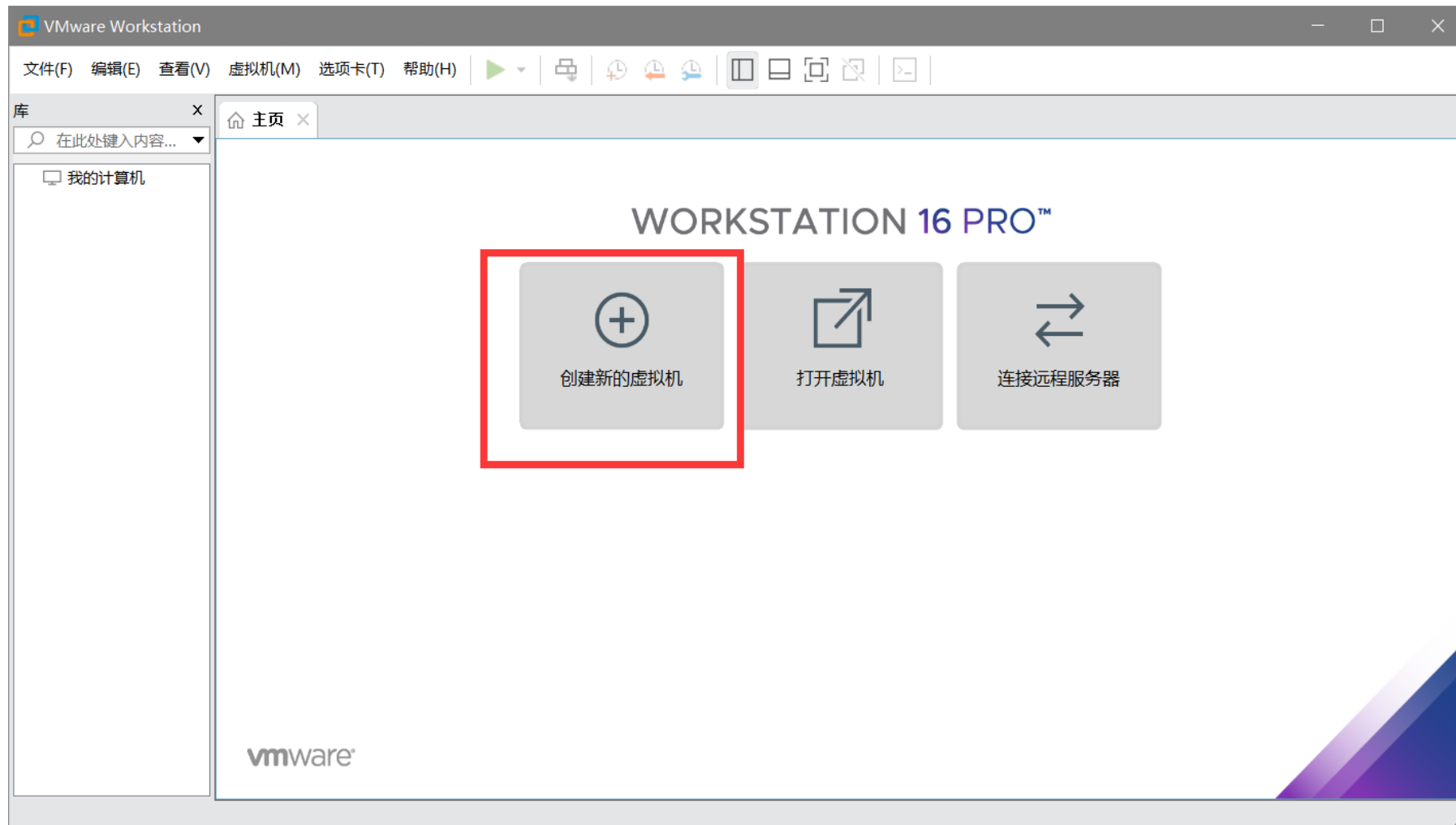
- 版本：
 - VMware Workstation 15 Pro 或 16 Pro
- 下载地址：
 - <https://www.vmware.com/cn/products/workstation-pro/workstation-pro-evaluation.html>
- 详细的安装过程参见课程附件



编辑虚拟机设置



新建虚拟机





6 利用VMWare虚拟机安装Linux

Ubuntu Linux的下载

- Ubuntu Linux 下载地址：
 - <https://ubuntu.com/download/desktop>
- 版本可自行选择
 - 以Ubuntu Linux 20为例

下载 Ubuntu 桌面

Ubuntu 20.04.3 LTS

下载适用于台式机和笔记本电脑的最新LTS版本的 Ubuntu。LTS 代表长期支持——这意味着五年，直到 2025 年 4 月，免费安全和维护更新，保证。

[Ubuntu 20.04 LTS 发行说明](#)

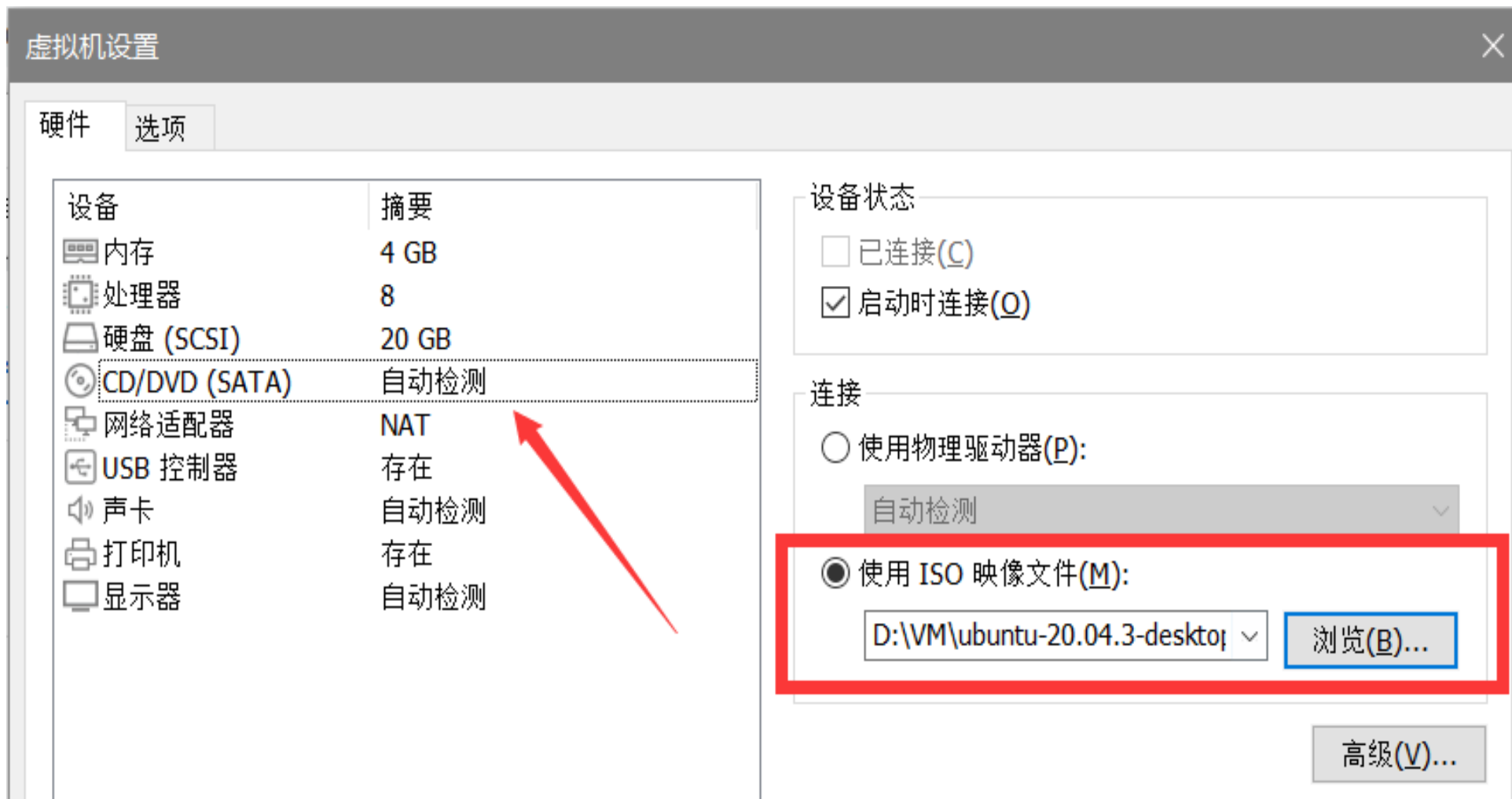
推荐的系统要求：

- ✓ 2 GHz 双核处理器或更好
- ✓ 4 GB 系统内存
- ✓ 25 GB 可用硬盘空间
- ✓ 上网很有帮助
- ✓ 用于安装程序介质的 DVD 驱动器或 USB 端口

下载

对于其他版本的 Ubuntu 桌面，包括种子、网络安装程序、本地镜像列表和过去的版本，[请参阅我们的替代下载](#)。

安装Linux映像文件(ubuntu-20.04.3-desktop-amd64.iso)



Aug 23 01:22



安装完成



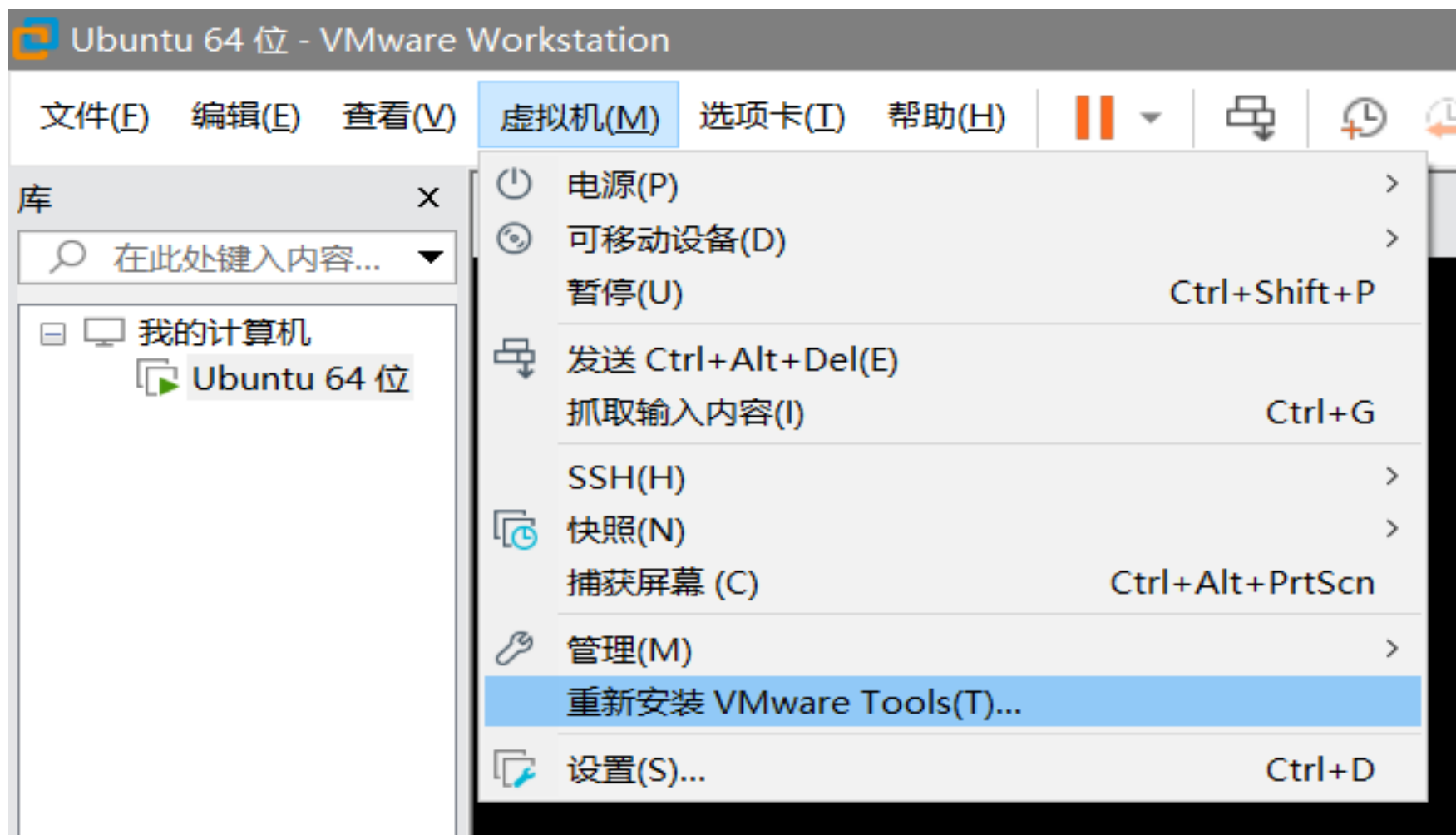
安装完毕。您需要重新启动计算机以使用新安装的系统。

现在重启



7 VMWare Tools的安装

安装VMWare Tools



输入安装命令

```
wenjie@wenjie-virtual-machine:~/桌面/vmware-tools-distrib$ sudo su  
[sudo] wenjie 的密码:  
root@wenjie-virtual-machine:/home/wenjie/桌面/vmware-tools-distrib# ./vmware-install.pl
```

安装成功提示

1. Manually start /usr/bin/vmware-user
2. Log out and log back into your desktop session
3. Restart your X session.

Enjoy,

.. --the VMware team

最后，重启虚拟机软件或电脑！



内容导航：

课程思政

课程思政

运用科学、辩证的观点和思想方法分析问题

20世纪90年代初期，基于MINIX和UNIX思想而研发的开源Linux系统面市，其是一款支持多用户、多任务、多线程和多内核的操作系统，不仅能够运行UNIX工具软件、应用程序和网络协议，还具有稳定的系统性能。发展至今，Linux已有上百种不同的发行版本。

在Linux系统稳步发展过程中，Windows系统亦不分昼夜地进行着功能完善、界面美化以及版本更新等工作。进入21世纪后，微软公司的Windows系统在个人计算机领域基本占领了垄断地位。

由垄断所导致的潜在安全问题是各国相关部门尤为关心的核心问题，而解决该问题（即去微软公司化）的主流途径便是采用开源Linux系统。

运用科学、辩证的观点和思想方法分析问题

但是，要想简单地通过采用Linux系统实现去微软公司化，实属不易。例如：2004年，德国慕尼黑政府宣布将政府办公计算机中所采用的Windows系统换为Linux系统，然而10年之后试验并未获得预期的效果。

从此类案例中可以看出，单纯地通过采用Linux系统+开源软件的模式来降低运维信息化成本，效果并不理想，后期维护工作量大，原因是采用Linux系统实现去微软公司化缺乏相应的产业基础。

但我们也绝对不应放弃反垄断！在此情形下，我们知识分子与科技人才更应运用科学、辩证的观点和思想方法分析问题：究竟应当如何解决操作系统垄断问题，以及如何在解决该问题的过程中，通过发挥我们自身的价值，助力研发自主可控的国产操作系统？



课后练习题

- 1、Linux目前流行的有哪些版本，请简单做一个介绍。
- 2、请简述操作系统内核的微内核设计的主要思想。
- 3、操作系统内核的宏内核设计的主要思想是什么，请简述。
- 4、请简述Linux操作系统中的用户空间的含义。
- 5、Linux操作系统中的内核空间的含义是什么，请简述。

谢谢！

THANKS