

1.2 Linux内核结构及内核模块

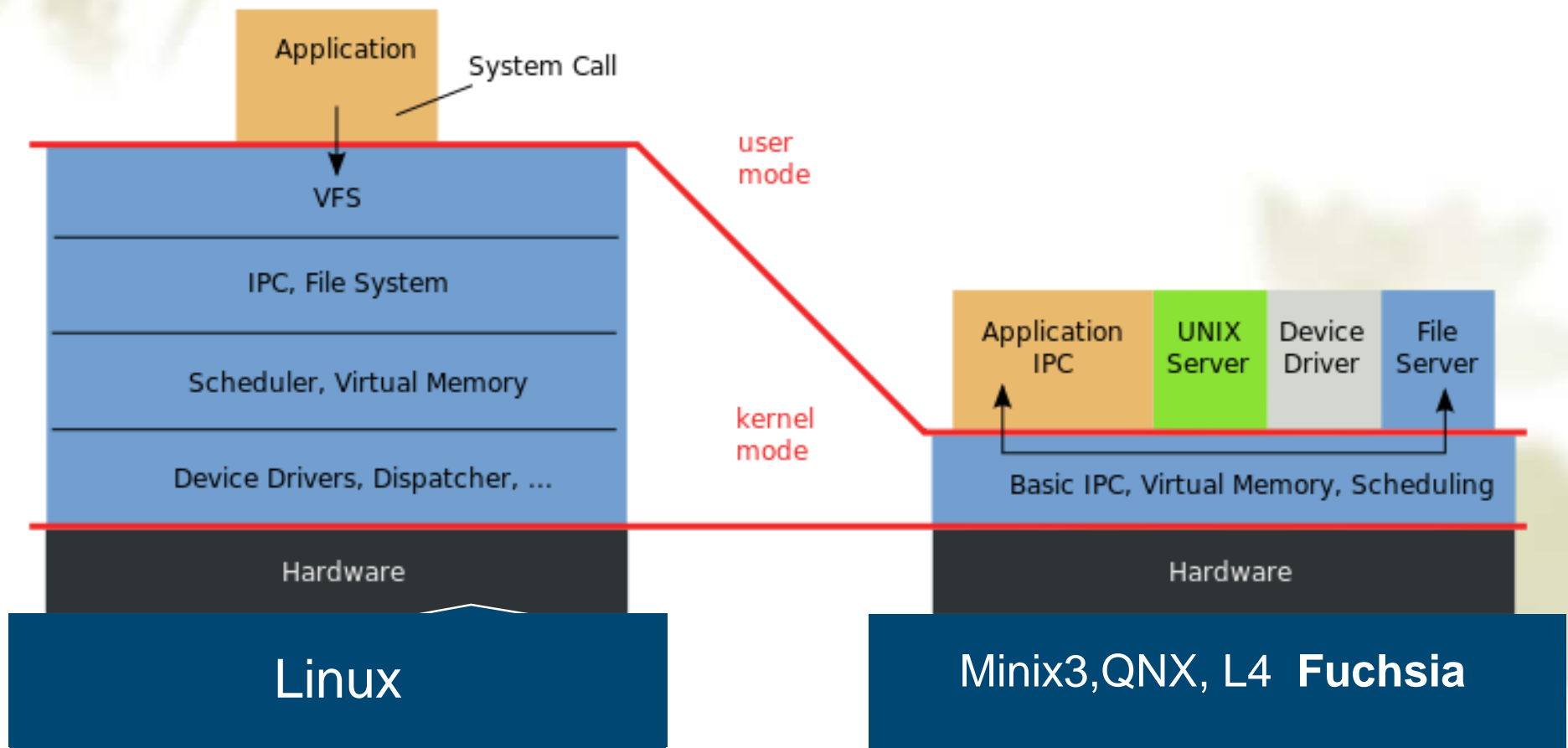


西安邮电大学

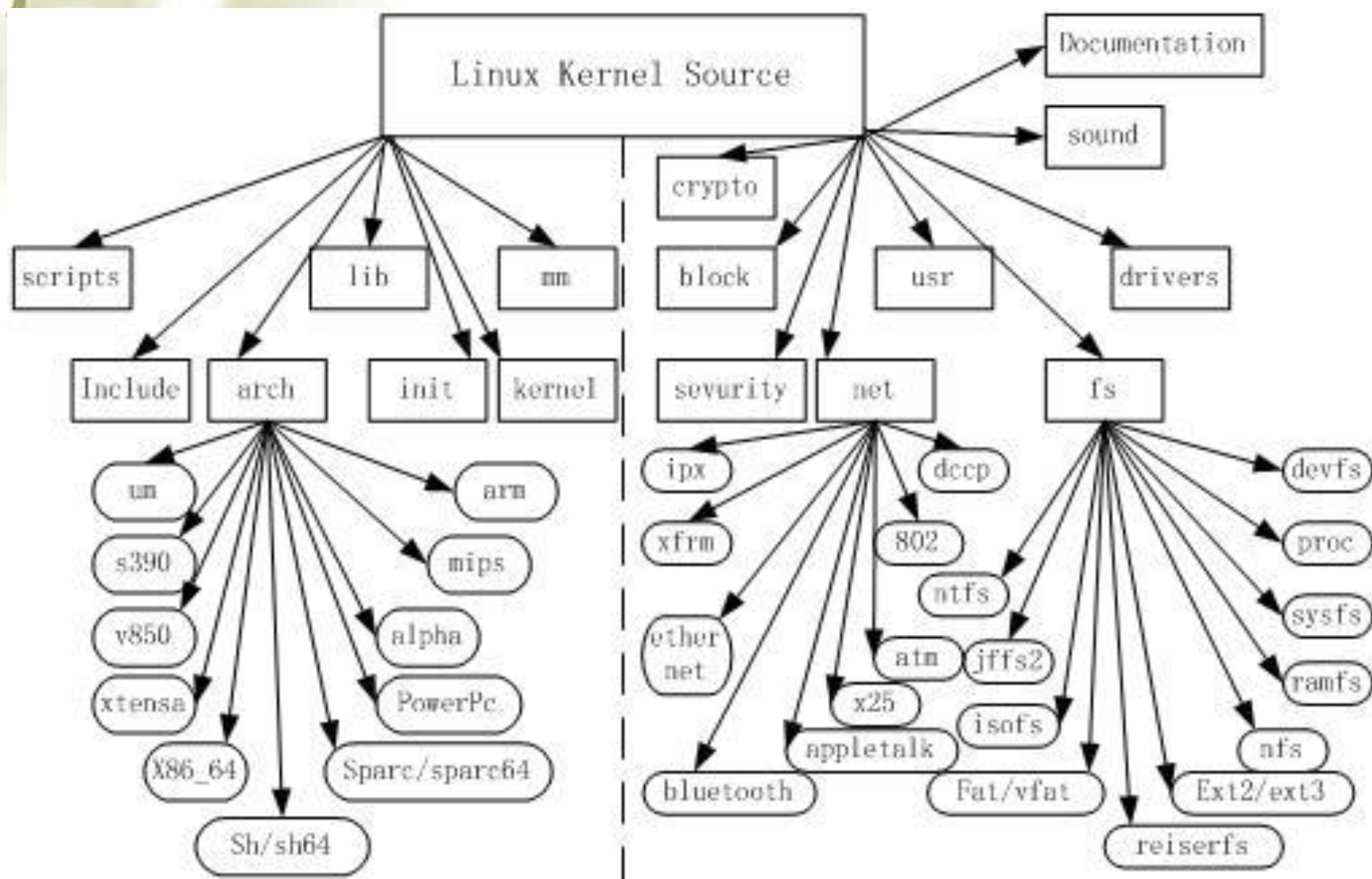
单内核与微内核

Monolithic Kernel
based Operating System

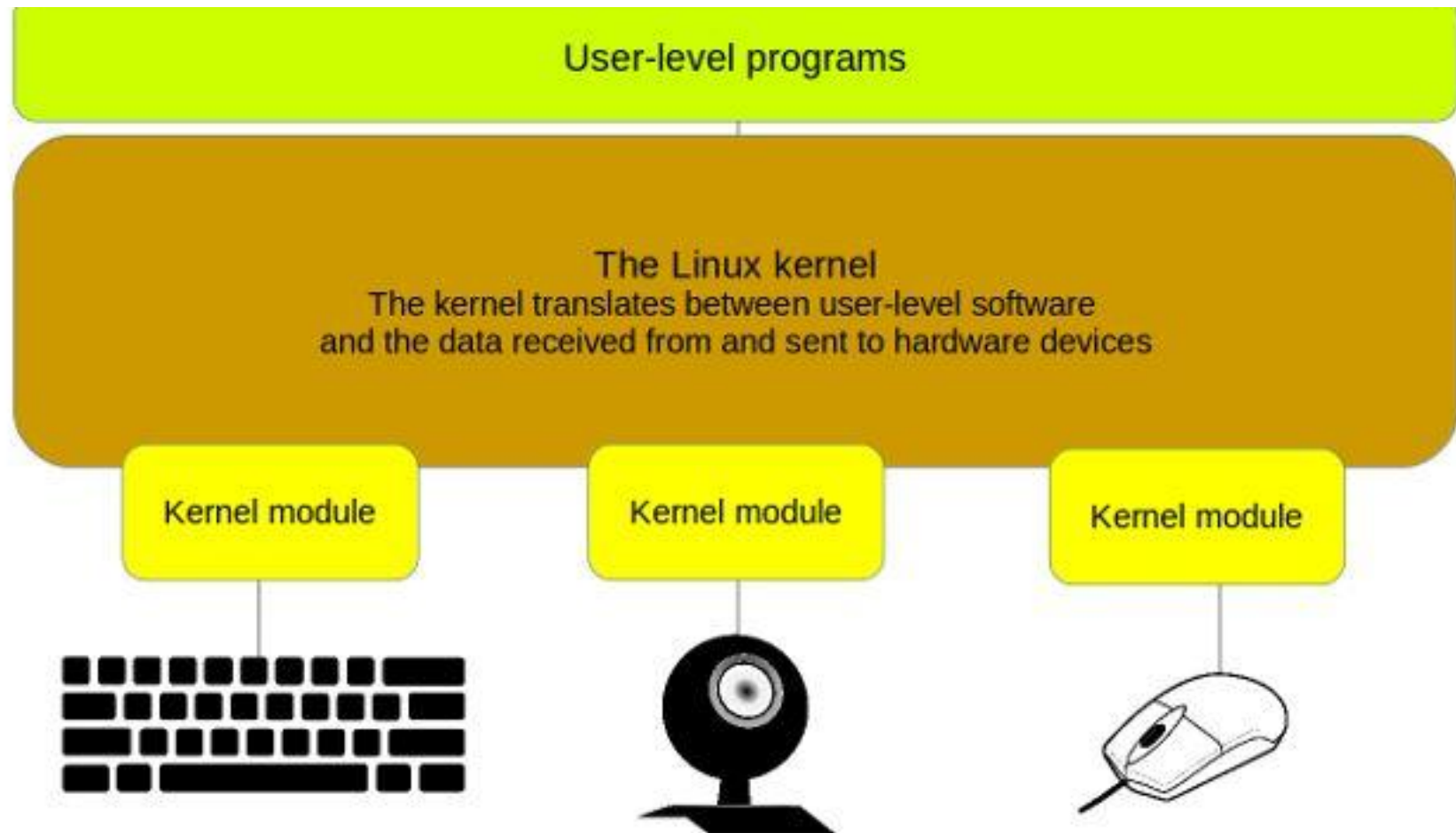
Microkernel
based Operating System



Linux内核源码目录结构



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\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\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。字符串中的<1>是输出的级别，表示立即在终端输出。

Linux内核模块编程入门

◆ 内核模块的Makefile文件

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。

```
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
```

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
```

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

动手写一个自己的Linux内核模块



谢谢大家！



THANK YOU