01、 boot/boot.asm：

//加载文件入内存需要读软盘

//LABEL\_START初始化堆栈；清屏；显示 "Booting "

//ReadSector读软盘扇区

//寻找LOADER.BIN（遍历根目录区所有的扇区，将每一个扇区加载入内存，寻找文件名为LOADER.BIN的条目）；

LABEL\_FILENAME\_FOUND=>得到LOADER.BIN的起始扇区号

LABEL\_GOON\_LOADING\_FILE每读一个扇区就在 "Booting " 后面打一个点

//DispStr显示字符串

//GetFATEntry（通过扇区号找到对应的FAT项的值）

//jmp BaseOfLoader:OffsetOfLoader 跳入loader

未改动

02、 boot/loader.asm：

//寻找KERNEL.BIN的代码与boot.asm中寻找LOADER.BIN是相似的

只不过boot.asm中是跳转到已加载到内存中的LOADER.BIN的开始处执行代码，而loader.asm则是跳入保护模式

//保护模式下显示字符串由函数DispStr完成，实模式下由函数DispStrRealMode完成

//添加有关于内存使用情况的代码=>进入保护模式后显示内存信息=>启动分页机制

//初始化内核（遍历每一个 Program Header，根据Program Header中的信息来确定把什么放进内存，放到什么位置，以及放多少。)

//正式进入内核(jmp SelectorFlatC:KernelEntryPointPhyAddr)

未改动

03、 boot/include/fat12hdr.inc：

//FAT12磁盘的引导扇区BPB等头信息

//为boot.asm和loader.asm所共享

未改动

04、 boot/include/load.inc：

//加载时段地址等的宏定义

//为boot.asm和loader.asm所共享

未改动

05、 boot/include/pm.inc：

//描述符类型、选择子类型、分页机制需要的常量等宏定义

未改动

06、 include/console.h：

//CONSOLE控制台的宏定义

未改动

07、 include/const.h：

//宏定义PUBLIC、PRIVATE用来区分全局和局部

//GDT\_SIZE

//8259A相应端口的宏定义

TTY数量改为1

#define DEFAULT 0x07 /\* 0000 0111 \*/

08、 include/global.h：

//gdt、idt

声明了is\_find;is\_show;find\_index;find\_content[1024];

09、 include/keyboard.h：

//键的宏定义

//键盘缓冲区的结构体定义

未改动

10、 include/keymap.h：

//扫描码解析数组

未改动

11、 include/proc.h：

//STACK\_FRAME进程表结构体

//PROCESS进程

//定义任务堆栈

删除了最后testB、testC的栈声明

NR\_TASKS数量改为2（TTY和TESTA）

12、 include/protect.h：

//存储段描述符/系统段描述符结构体

//中断向量的宏定义

//门描述符结构体

//描述符、选择子等的宏定义

未改动

13、 include/proto.h：

//总之就是很多的函数声明

未改动

14、 include/sconst.inc：

//一些寄存器的地址

未改动

15、 include/string.h：

//memcpy的函数声明

//memset的函数声明

未改动

16、 include/tty.h：

//TTY相关定义

未改动

17、 include/type.h：

//u8、u16、u32等类型定义=>增强代码可读性

未改动

18、 kernel/clock.c：

//clock\_handler时钟中断处理

//milli\_delay延迟函数

//init\_clock初始化时钟

修改clock\_handler()方法

19、 kernel/console.c：

//init\_screen=>（set\_cursor）

//is\_current\_console判断是否为当前控制台

//out\_char=>（flush=>（set\_cursor；set\_video\_start\_addr））

//select\_console切换控制台=>（set\_video\_start\_addr切换控制台）

//scroll\_screen屏幕滚动=>（set\_cursor；set\_video\_start\_addr）

if else删除

out\_char增加判断

20、 kernel/global.c：

//变量没有extern关键字

//进程表等的声明

删除了task\_table中的TestB、TestC

21、 kernel/i8259.c：

//init\_8259A=>设置8259A

//spurious\_irq=>打印IRQ号

//put\_irq\_handler=>设定中断处理程序，比如设定时钟中断处理程序put\_irq\_handler(CLOCK\_IRQ, clock\_handler)

未改动

22、 kernel/kernel.asm：

//切换堆栈，更换GDT（用到了start.c中的cstart方法）

//中断和异常=>硬件中断；异常：调用异常处理函数exception\_handler=>save=>restart=>ring->ring1

//kernel\_main执行main.c的kernel\_main函数

定义了global close\_int、global open\_int即close\_int:cli open\_int:sti开中断和关中断

23、 kernel/keyboard.c：

//keyboard\_handler键盘中断处理程序

//init\_keyboard=>设定键盘中断处理程序，开键盘中断（在kernel\_main中调用）

//keyboard\_read对扫描码解析，交给in\_process处理

//get\_byte\_from\_kbuf从键盘缓冲区中读取下一个字节（关闭中断cli，结束后打开sti）

//kb\_wait等待输入缓冲区空

删去了对左右ctrl、左右alt、numlock、scrolllock的定义，有关部分删去

同时不需要控制灯的亮灭，删去set\_leds()方法，有关部分删去

同时不需要小键盘的处理，删去pad部分

24、 kernel/main.c：

//kernel\_main（清空屏幕；初始化进程表；调用init\_clock以及init\_keyboard；调用restart）

//TestA进程A（设置进程延迟时间）

删除开始的输出

删除了优先级的设置

修改了testA，将延迟时间改为20000ms，删除了testB和testC

增加了清屏操作

增加了判断操作

25、 kernel/pro.c：

//sys\_get\_ticks系统调用得到当前总共发生的时钟中断的次数

不需要进程调度，删去schedule()方法，相应修改clock.c

26、 kernel/protect.c：

//异常处理函数exception\_handler

//初始化段描述符init\_descriptor

//由段名求绝对地址seg2phys

//初始化中断门：init\_idt\_desc

//设置IDT：init\_prot=>1、init\_8259A；2、几乎只调用init\_idt\_desc，所有描述符被初始化为终端门

未改动

27、 kernel/start.c：

//cstart函数=>更换GDT，初始化GDT和IDT

删除了输出

28、 kernel/syscall.asm：

//系统调用

未改动

29、 kernel/tty.c：

//in\_process对键的处理=>(put\_key)

//task\_tty=>（init\_tty；tty\_do\_read；tty\_do\_write）

删去了对多余的键的处理

添加了对Tab键的处理以及一些判断条件

添加了show\_result和recover方法

30、 lib/klib.c :

//disp\_int显示整数

//itoa将整数转换为字符串

//delay设置时间间隔

未改动

31、 lib/kliba.asm：

//disp\_str打印字符串

//out\_byte写端口

//in\_byte读端口

//disp\_color\_str异常处理用颜色突出显示

//disable\_irq/enable\_irq屏蔽中断和允许接受中断；disable\_int/enable\_int

未改动

32、 lib/string.asm：

//memcpy内存拷贝函数

//memset设置内存空间的值

//strcpy字符串复制函数

未改动

33、 bochsrc：

删除了一些不必要的属性

将boot中的a改为floppy

34、 Makefile：

CFLAGS后增加-fno-stack-protector

buildimg中增加make realclean;bochs -f bochsrc;删去sudo umount /mnt/floppy

删去default start position中的nop :@echo "why not \`make image' huh? :)"提示

将image改为run以符合make run的要求

35、 a.img：

默认存在