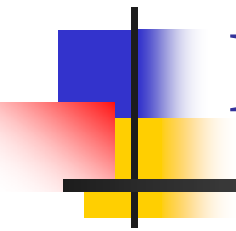
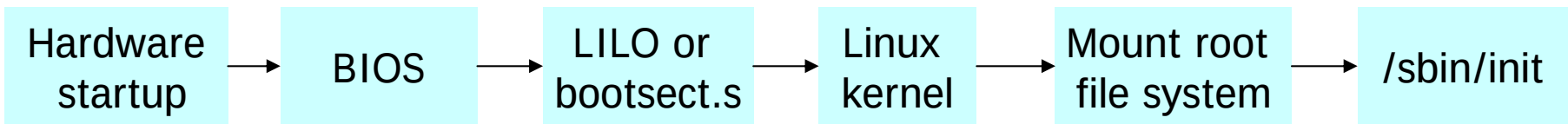


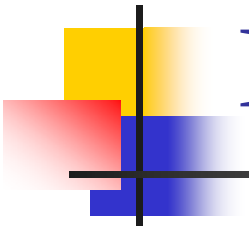
Introduction to Linux start-up

Hao-Ran Liu

Boot process overview



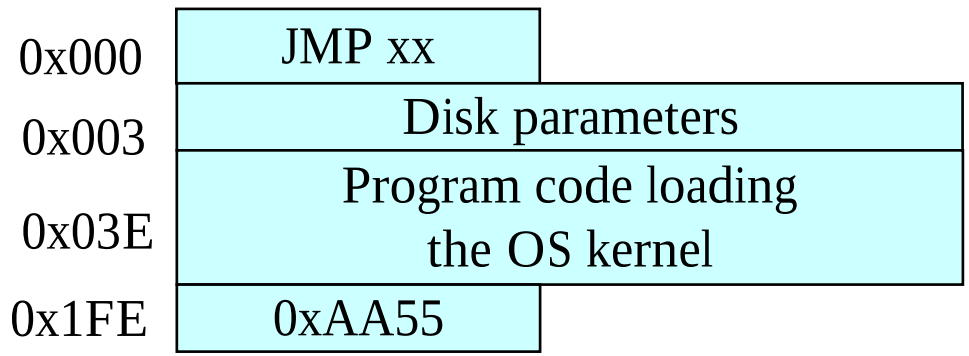
LILO is a versatile boot loader,
but it is functional-equivalent to bootsect.s in Linux (version 2.4 or before)



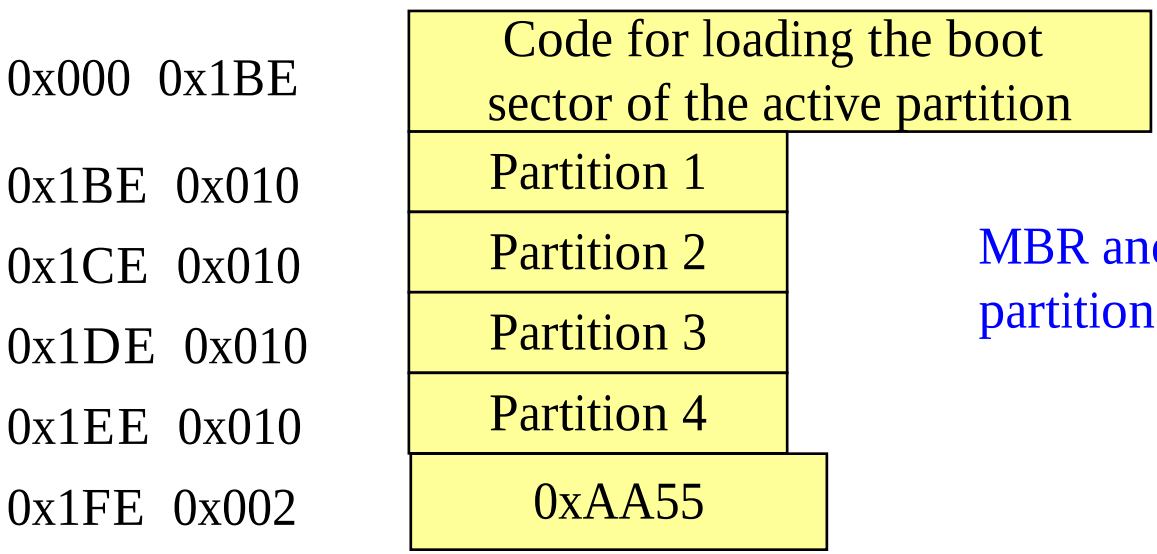
Boot process

- BIOS
  - reads the first sector of the boot disk (floppy, hard disk, …, according to the BIOS parameter setting)
  - the boot sector (512 bytes) will contain program code for loading the operating system kernel (e.g., Linux Loader, LILO)
  - boot sector ends with 0xAA55
- Boot disk
  - Floppy: the first sector
  - Hard disk: the first sector is the master boot record (MBR)

Boot sector and MBR



Boot Sector (Floppy)



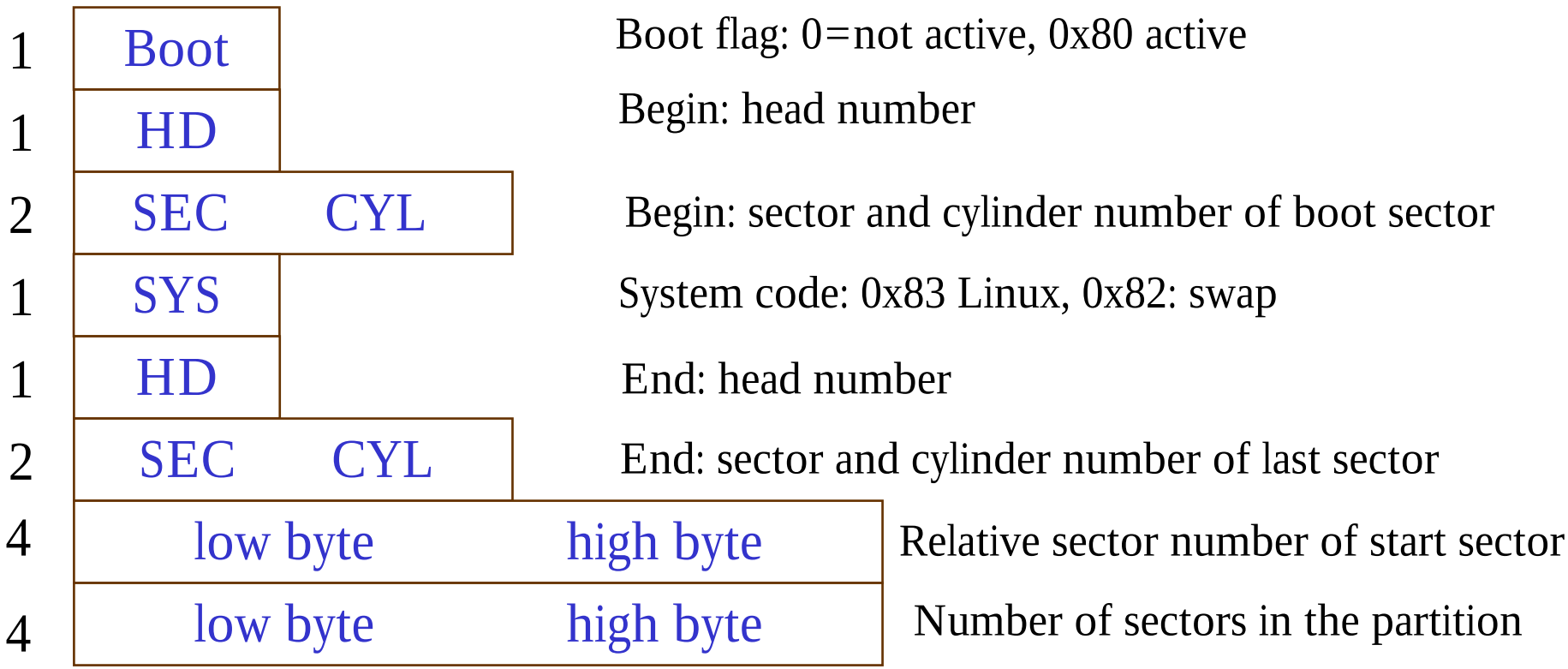
MBR and extended partition table (Hard disk)

MBR (*Master Boot Record*)

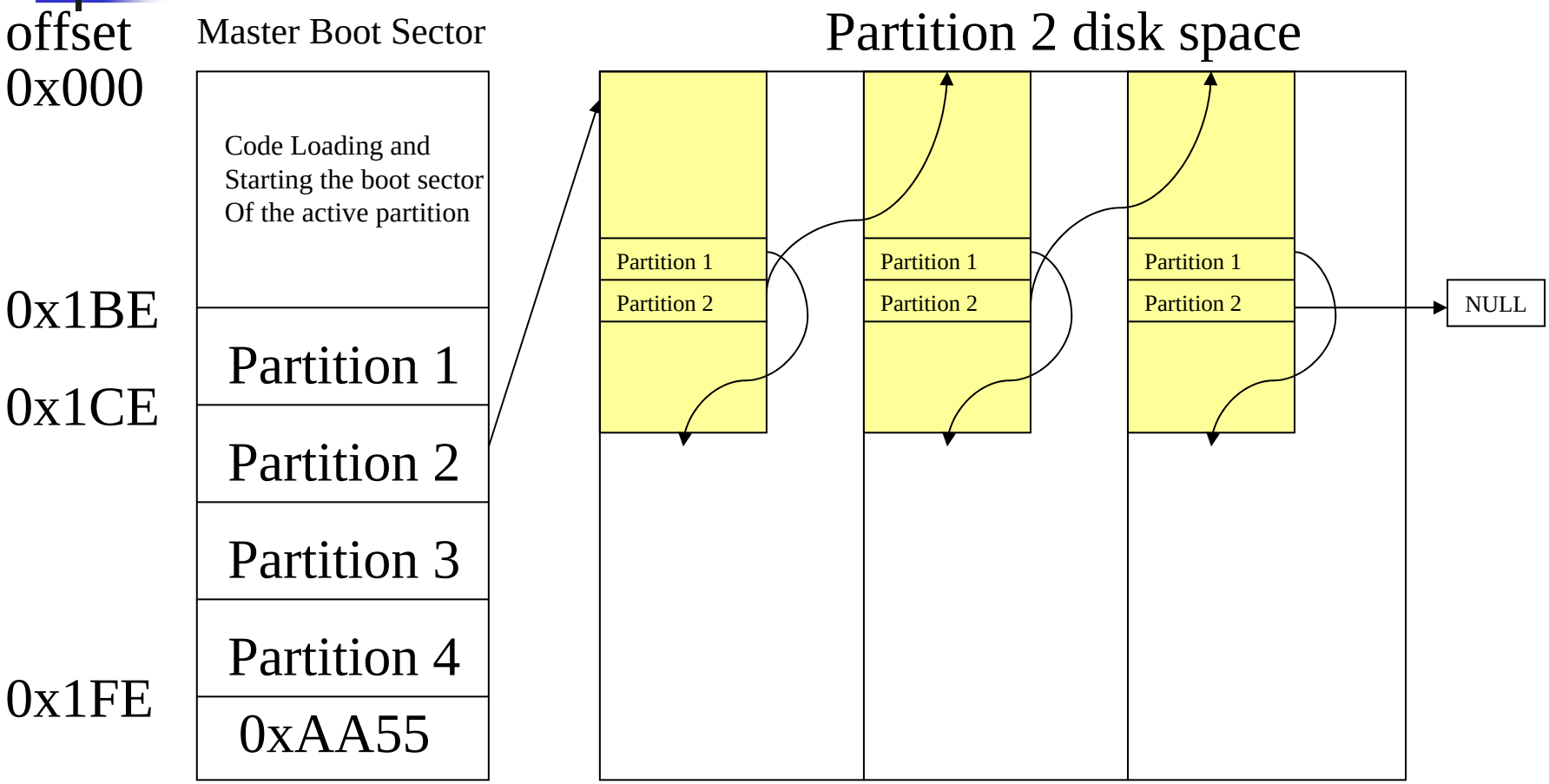
Four primary partitions
- only 4 partition entries
- Each entry is 16 bytes

- Extended partition
  - If more than 4 partitions are needed
  - The first sector of extended partition is same as MBR
  - The first partition entry is for the first logical drive
  - The second partition entry points to the next logical drive (MBR)
- The first sector of each primary or extended partition contains a boot sector

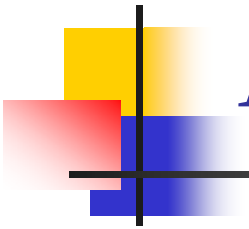
Structure of a Partition Entry



Extended partition table



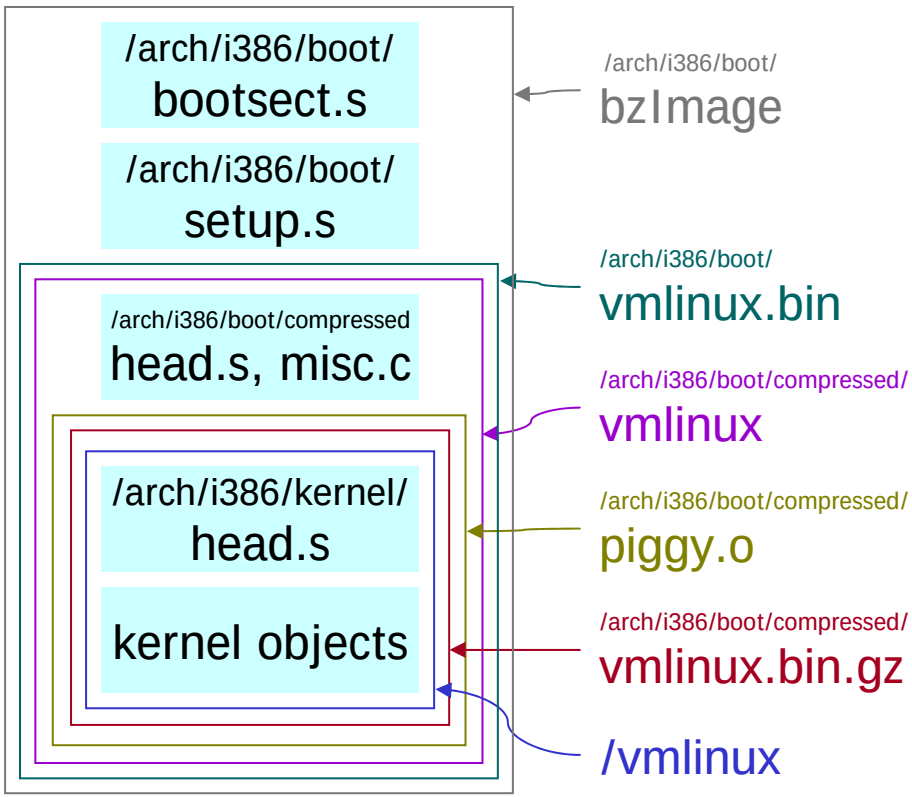
The structure of MBR and Extended Partition are the same.



Active Partition

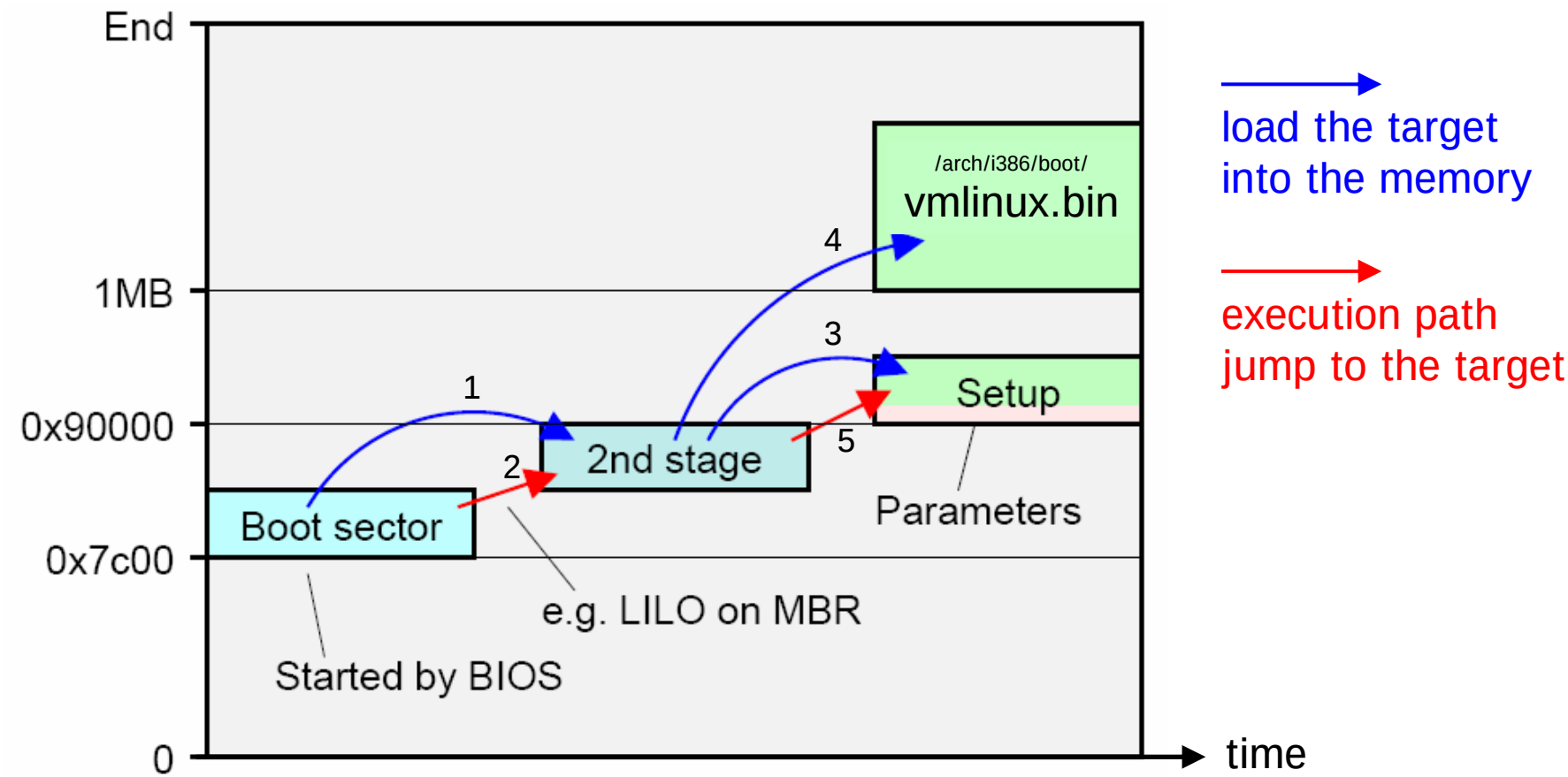
- Booting is carried out from the active partition which is determined by the boot flag
- Operations of MBR
  - determine active partition
  - load the boot sector of the active partition
  - jump into the boot sector at offset 0

Linux 2.6 kernel image

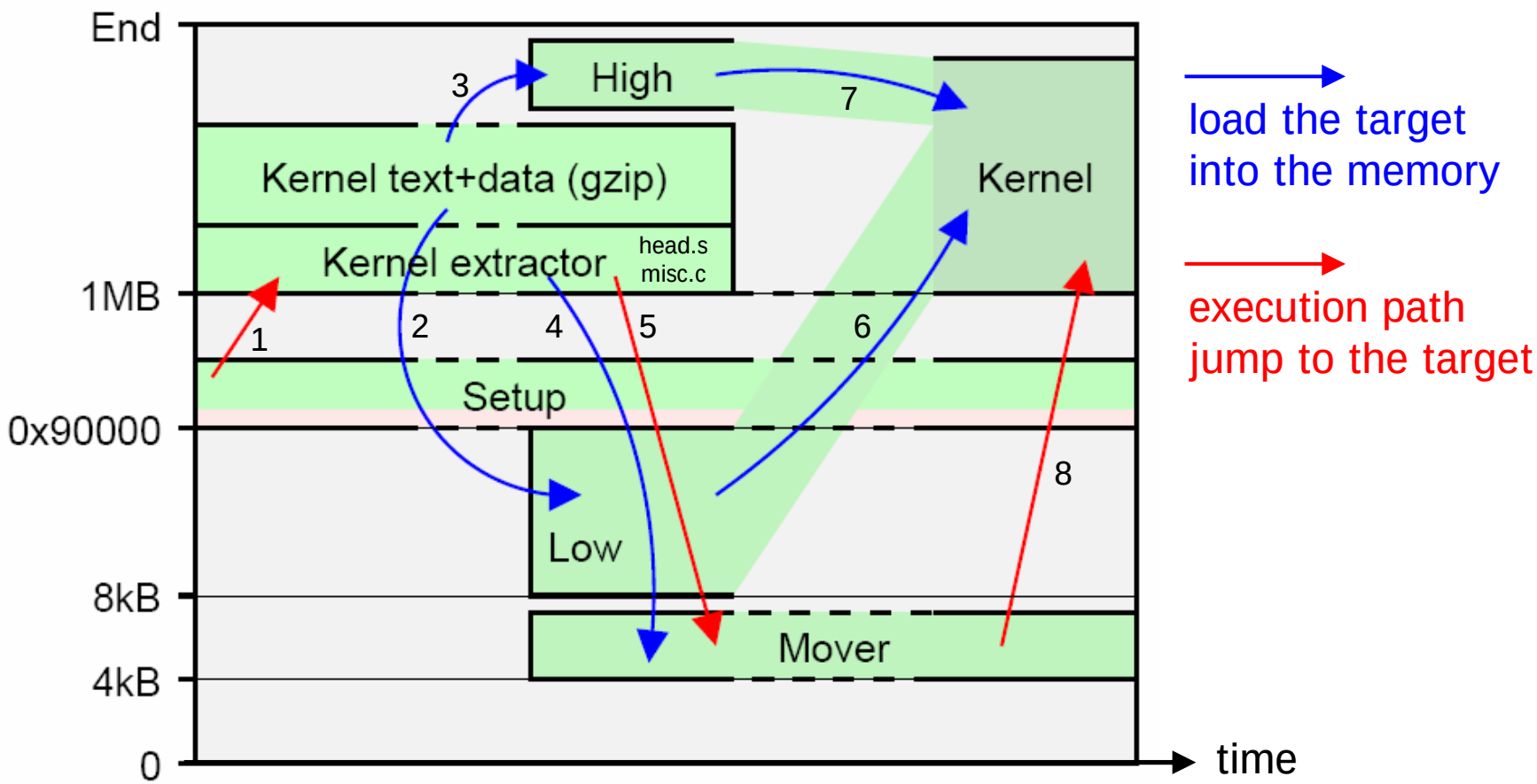


- bootsect.s: Linux floppy boot loader (only in version <= 2.4)
- setup.s: hardware initialization, switch to protected mode
- First head.s, misc.c: decompress kernel
- Second head.s: enable paging, setup GDT, jump to C routine start_kernel()

Loading a bzImage



Starting a bzImage

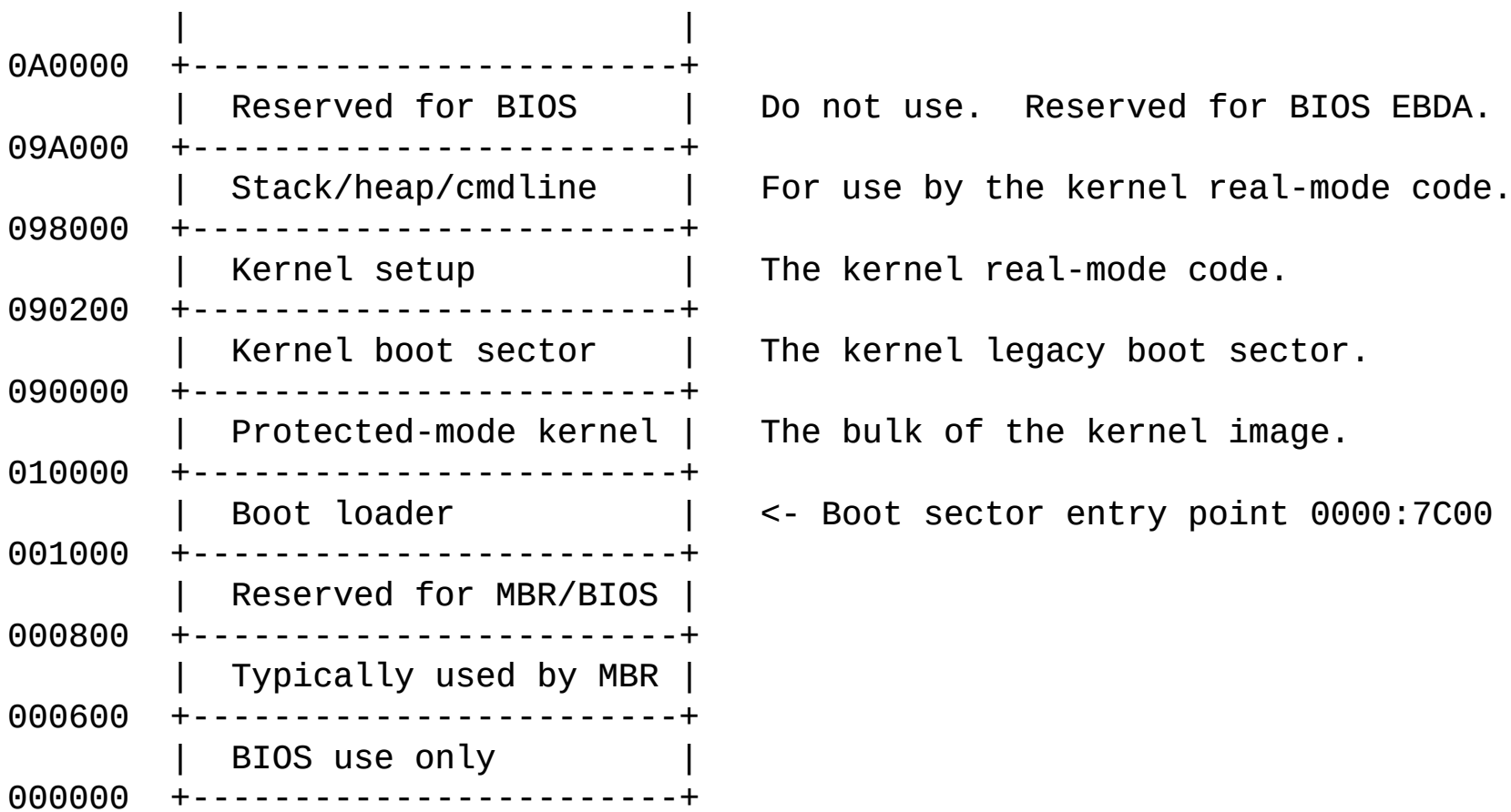


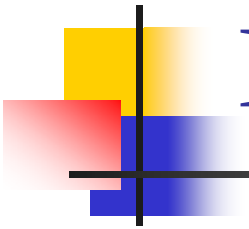
zImage and bzImage

- Both zImage and bzImage are compressed with gzip
- The only difference
  - zImage is loaded low and has a size limit of 512KB

	zImage	bzImage
Boot loader (bootsect.s) places /arch/i386/boot/ vmlinux.bin at	between 0x10000~0x90000	above 0x100000
Decompressed kernel (/arch/i386/boot/compressed/ vmlinux.bin)'s final address	0x100000	0x100000

The Linux/i386 boot protocol

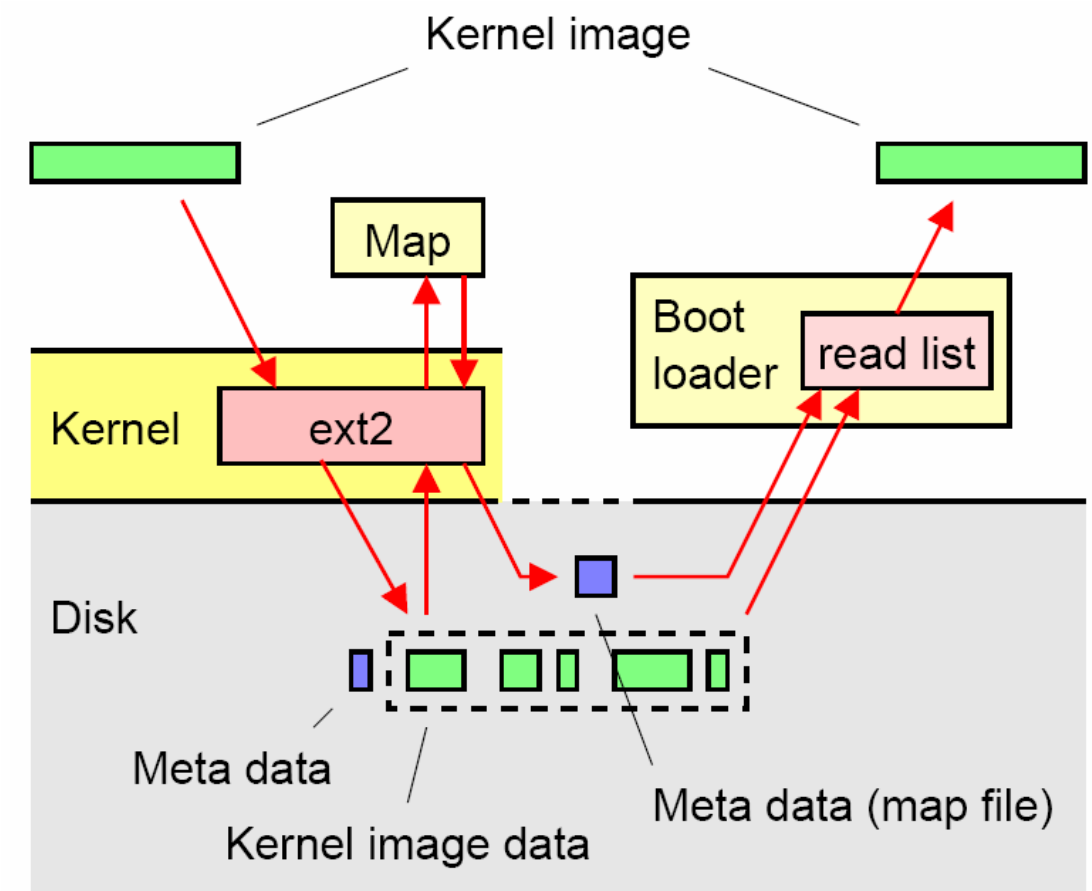




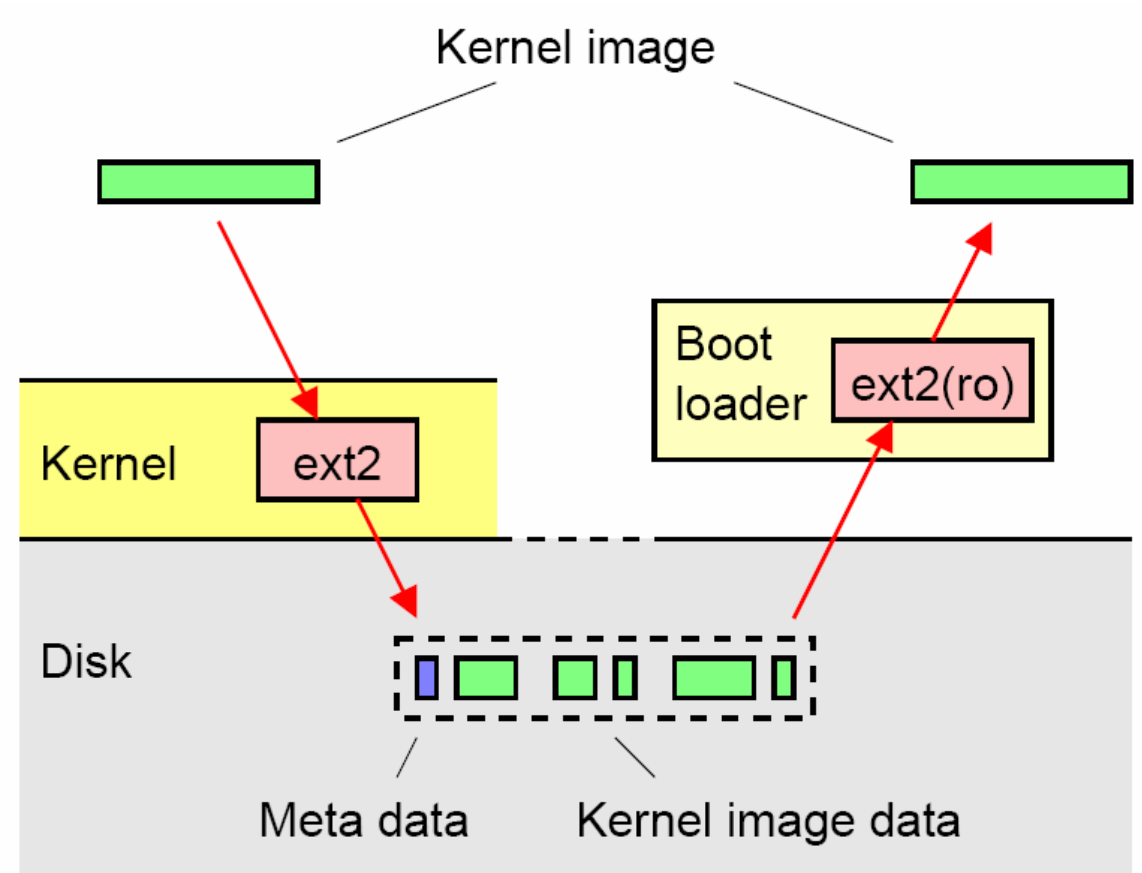
LILO vs. GRUB

- LILO
  - Boot any file system
  - Need regeneration of a map file if kernel changes
  - Friendly to file system developers
- GRUB
  - Boot only known file system
  - No map file!
  - Friendly to normal users

LILO – file system unaware



GRUB – file system aware



Starting the kernel

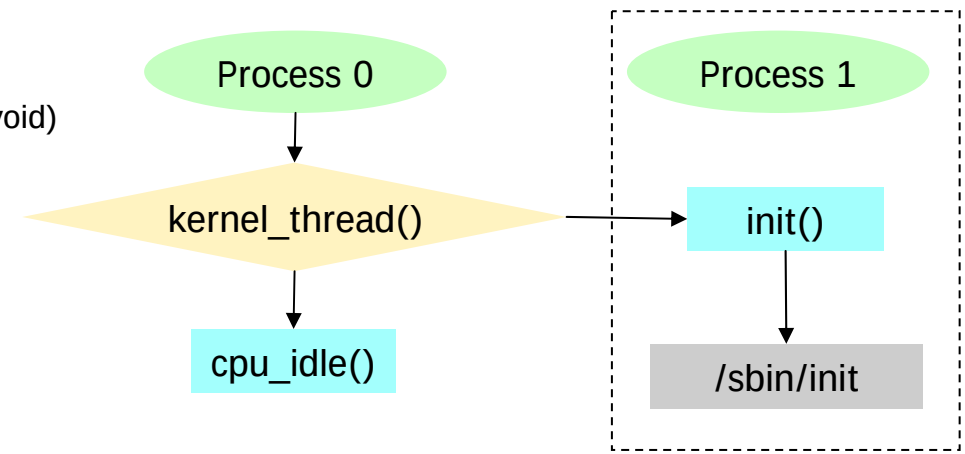
- After head.s and misc.c decompress kernel, it calls the first C routine start_kernel()
- Many hardware-independent parts of the kernel are initialized here.

```
asmlinkage void __init start_kernel(void)
{
  char * command_line;
  printk(linux_banner);
  setup_arch(&command_line);
  parse_options(command_line);
  trap_init();
  init_IRQ();
  sched_init();
  time_init();
  softirq_init();
  console_init();
...
```

Spawn first process - init

- The original process now running is process 0
  - It generates a kernel thread (process 1) executing init() function
  - Subsequently, it is only concerned with using up unused computing time - cpu_idle()

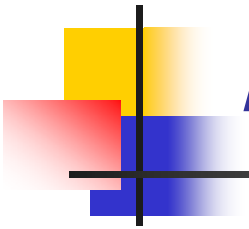
```
asmlinkage void __init start_kernel(void)
{
...
  kernel_thread(init, NULL, ...)
  cpu_idle();
}
```



init() function

- Carry out the remaining initialization and open file descriptors 0, 1, 2 for the first user program being exec'ed later
- Try to execute a boot program specified by the user or one of the programs /sbin/init, /etc/init, or /bin/init
- If none of these programs exists, try to start a shell so that the superuser can repair the system. If this is not possible too, the system is stopped

```
static int init() {
{
  do_basic_setup();
  ...
  if (open "/dev/console", O_RDWR, 0) < 0)
    printk("Warning: unable to open an initial console.\n");
  (void) dup(0);
  (void) dup(0);

  if (execute_command)
    execve(execute_command, argv_init, envp_init);
  execve("/sbin/init", argv_init, envp_init);
  execve("/etc/init", argv_init, envp_init);
  execve("/bin/init", argv_init, envp_init);
  execve("/bin/sh", argv_init, envp_init);
  panic("No init found...");
}
```

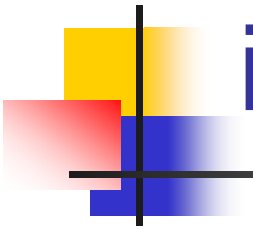


/sbin/init – parent of all processes

- Init configures the system and create processes according to /etc/inittab
- A Runlevel is:
  - A software configuration of what services to be started
  - A state that the system can be in
- /etc/inittab defines several runlevels

Description of runlevels

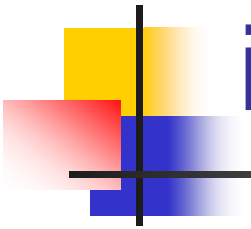
Runlevel	Description
0	Halt - used to halt the system
1	Single-user text mode
2	Not used
3	Full multi-user text mode
4	Not used
5	Full multi-user graphic mode (with an X-based login screen)
6	Reboot – used to reboot the system
S or s	Used internally by scripts that run in runlevel 1
a,b,c	On-demand run levels - typically not used.



inittab syntax

- An entry in the inittab has this format:
  - **id : runlevels : action : process**
  - **id:** an unique id to identify the entry
  - **runlevels:** a list of runlevels for which the specified action should be taken
  - **action:** describes which action should be taken
  - **process:** specifies the program to be executed

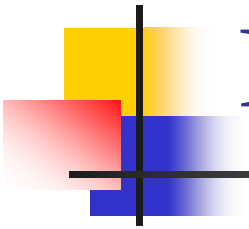
Please refer to man page inittab(5) for details of the action field



inittab example

```
# default runlevel is 3
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

# the start script of each runlevel
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```



References

- Werner Almesberger, Booting Linux: the history and the future, Ottawa Linux Conference, 2000