

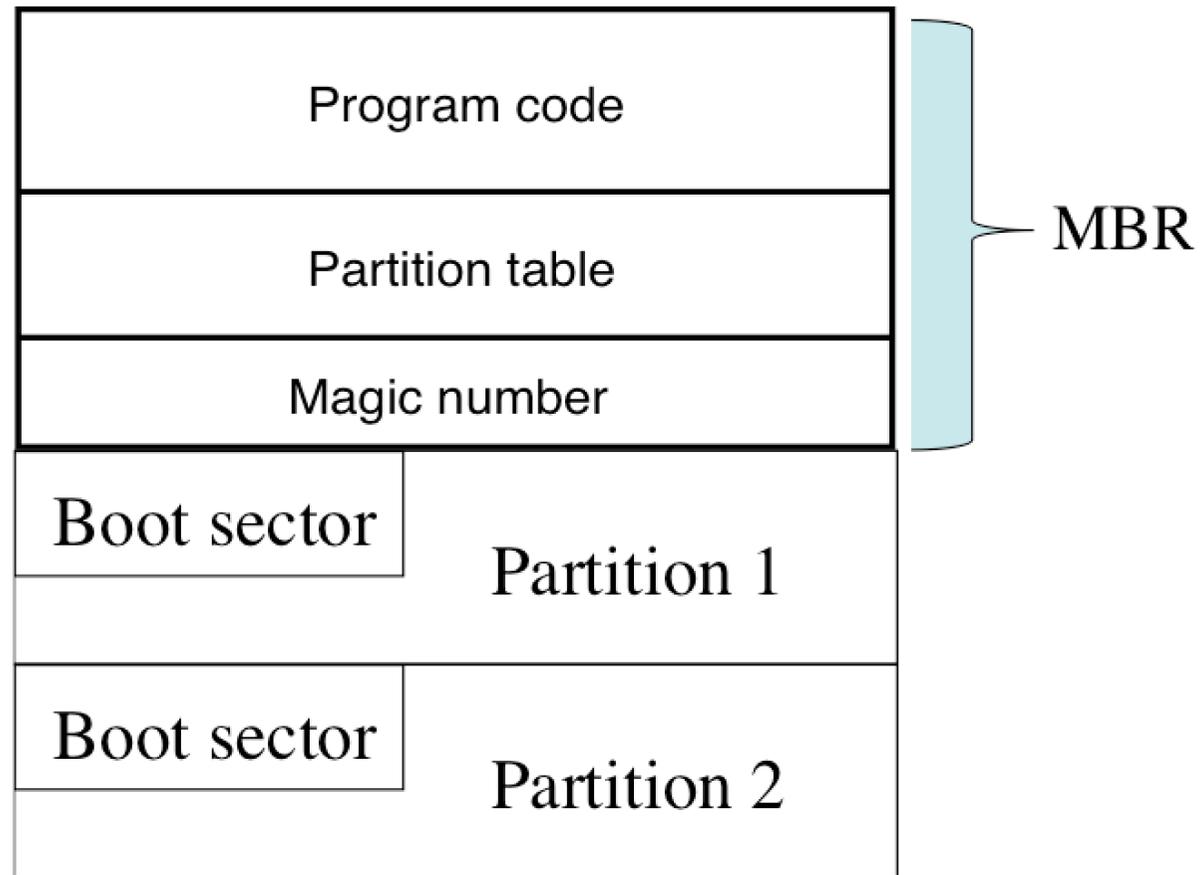
# Overview

- Last Lecture
  - Basic System Administration
- This Lecture
  - System installation
- Next Lecture
  - Post installation

# Installation of Linux

- Major Linux distributions
  - Ubuntu, Red Hat, Debian, Fedora, S.u.S.E., etc
- Why need to install?
  - Copy Linux onto the hard disk
  - Properly place the boot program, the Linux kernel, and startup scripts so that the computer can boot up from the hard disk
- Select an installation method
  - CD-ROM
  - NFS

# Disk layout after installation



# BIOS or EFI

- BIOS (Basic I/O System) or EFI (Extensible Firmware Interface)
  - A program that is written in ROM
  - Provides the lowest level interface to peripheral devices and controls the first step of the boot procedure
  - BIOS tests the system, looks for and checks peripherals and then looks for a device (floppy, hard disk, or CD-ROM) to use to boot the system
  - Master Boot Record (MBR) is the sector that the BIOS reads in and starts when a hard disk is used to boot.

# Disk Partitioning

- A hard disk can be divided into several partitions. Each partition functions as if it were a separate hard disk.
- If you have one hard disk, and want to have, say, two operating systems on it, you can divide the disk into two partitions. Each operating system uses its partition as it wishes and doesn't touch the other one. This way the two operating systems can co-exist peacefully on the same hard disk. Without partitions one would have to buy a hard disk for each operating system.

# Disk Partitioning (cont.)

- Why partition a disk?
  - Easy maintenance of file systems
    - If a partition is broken, just mount the backup partition.
  - Easy for system upgrade
    - Can avoid formatting other file systems when a distribution is to be upgraded
- How to partition?
  - Depends on the ultimate purpose for the system
    - Some investigation on space requirements for each filesystem is strongly recommended

# Disk Partitioning (cont.)

- The partitioning scheme or table
  - is not built into the hardware, or even into the BIOS. It is only a convention that many operating systems follow.
  - As a safety precaution, it is a good idea to write down the partition table on a piece of paper, so that if it ever corrupts you don't have to lose all your files.
- Partition types (aka file system types)
  - Linux (ext2, ext3, ext4), swap, DOS FAT, etc
- Partitioning a hard disk
  - **fdisk**, **cdisk**
- Magic number in MBR
  - 0xAA55
  - Used by BIOS to distinguish a bootable disk and a non-bootable disk

# Disk Partitioning (cont.)

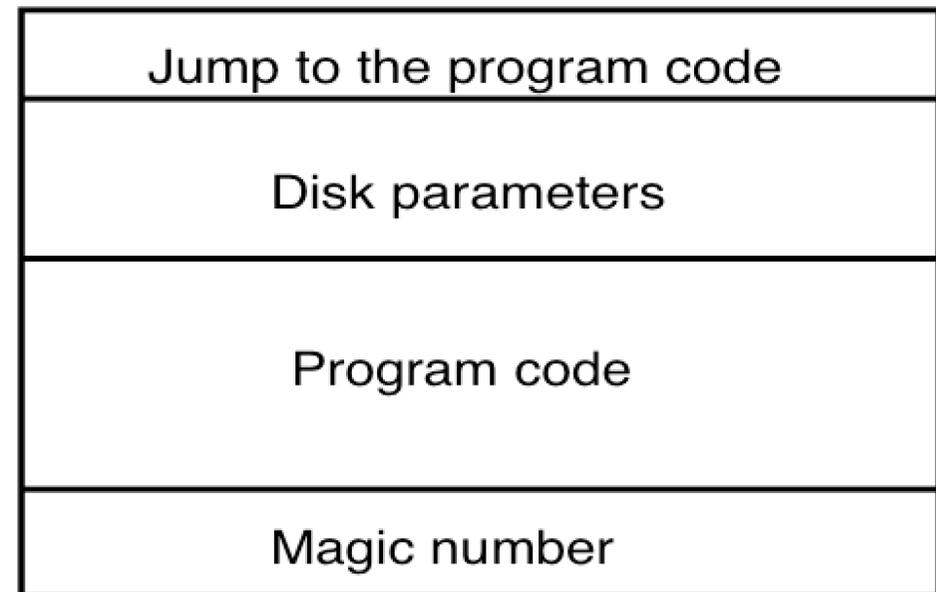
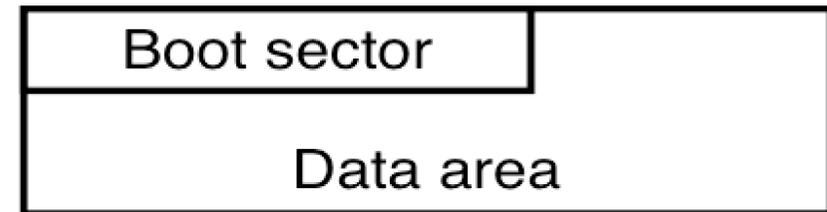
- Simple scheme
  - one linux partition, one swap partition
- Recommended scheme: more partitions for different file systems under linux
  - /: 100-150MB, root partition for /bin, /dev, /etc, /lib, /sbin,...
  - /boot: 20-50MB, boot partition, including linux kernels
  - /swap: double the memory space, but could be smaller, partition for virtual memory
  - /usr: depends on software packages, 1GB-5GB, including X windows
  - /var: 500MB - 1GB
  - /home: as large as possible
- Logical Volume Management (LVM)
  - For multiple hard drives with flexible FS space management.

# Boot procedure (hard disk)

- Boot from hard disk
  - OS dependent load program is installed on MBR
- Boot for multiple OSs
  - OS independent load programs such as LILO (Linux Loader) or GRUB (GRand Unified Bootloader) is installed on MBR
  - OS dependent load program is installed on the boot sector of the partition of the OS
  - MAC bootloader is at `/System/Library/CoreServices/boot.efi` and loaded directly by EFI

# Boot procedure (floppy or CD)

- Floppy organization
  - Consists of boot sector and data area
  - The bottom right picture is the layout of the boot section
- Boot from floppy
  - The BIOS loads the program code and executes it
  - That program loads OS



# Boot up phases

- BIOS
  - Boot sequence
- Bootloader
  - GRUB or LILO
- Kernel
  - initrd (initial ram disk) for initial root directory
- Upstart scripts under /etc/init.d
  - Re-mount root directory
- Refer to
  - <https://wiki.ubuntu.com/Booting>
  - <http://manpages.ubuntu.com/manpages/bionic/man7/boot.7.html>

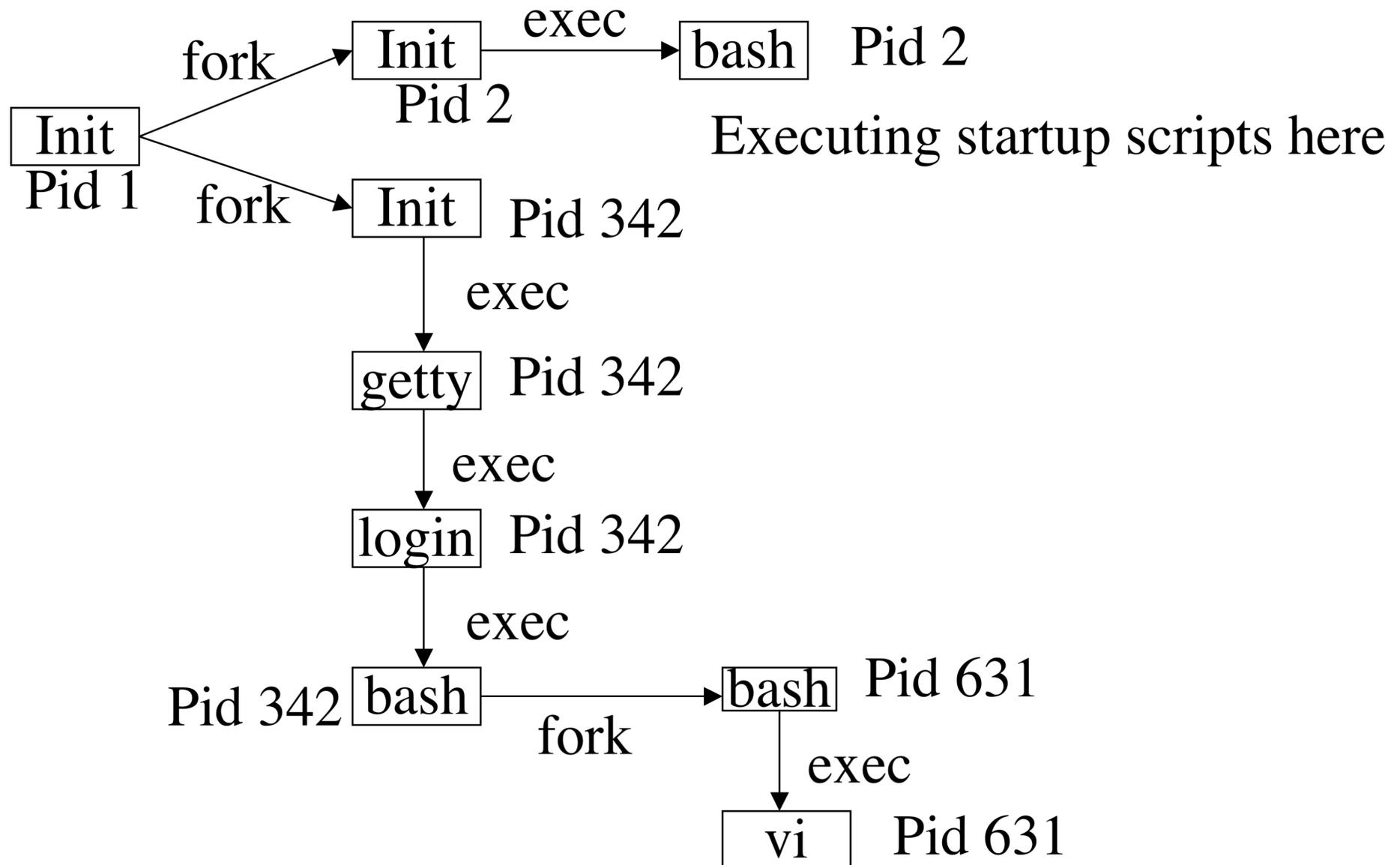
# How Linux started?

- When Linux kernel is loaded into memory and initialised, the *init* program is executed
- *init* is the first process in Linux
  - Runs some start-up scripts to start up services according to the run-level
  - Runs *getty* program that prompts login waiting for users to login

# Run levels for boot or shutdown

- Linux run levels
  - 0 - halt
  - 1 - single user mode
  - 2 - unused
  - 3 - multiuser mode (default mode)
  - 4 - X11 with GUI
  - 5 - unused
  - 6 - reboot
- Each level has a set of scripts for configuration

# Processes for non-GUI boot-up



# More details for boot-up

- When Linux boots,
  - The kernel mounts the root file system
  - The first program that the system executes is **init** (the first process)
    - It is responsible for starting the system startup scripts under **/etc/init.d**. The configurations for bootup are in **/etc/init**.
    - Modifies system run-level from initial bootup state to single user mode, then to its standard multiuser state.
    - Spawns `getty` ( which will invoke a login program)
  - Mount the file systems
    - File systems listed in **/etc/fstab** are mounted by command **mount -a** at boot
    - To mount more file systems, you should either modify **/etc/fstab**, or manually mount them.

# What if the system can't boot?

- If you made a mistake, e.g. the kernel image is corrupted
- If you accidentally removed the `/etc/passwd` but rebooted, what will happen? How to fix?
  - Use a live CD to boot the system
  - Mount the filesystem to be fixed
  - Recover the file
  - Reboot without the live CD.

# How to secure your machine?

- BIOS/EFI password
- Boot sequence
- Protect the single user mode with a root password
- Protect the machine in a secure room!

# Post installation

- Start a logbook
- Backup your system: a boot and rescue CD
- Edit login messages: `/etc/issue` and `/etc/motd`
- User privilege and access control
- Software installation
- Harden security

# Summary

- Linux boot process
  - BIOS/EFI starts first after power on
  - Bootloader (e.g. grub) is loaded into RAM from the boot sector of the booting device
  - Linux kernel and ram disk are loaded into RAM
  - After kernel initialization, the first process init is created (It is systemd in Ubuntu).
  - Init process read startup scripts via command shell like bash
  - Filesystems are mounted, software servers/daemons starts, and network interfaces are brought up.
  - Finally GUI (e.g. startx) is started or login shell (e.g. getty) is prompted for users to login.