Programming Lab 5E

Disk Drive Geometry

Topics: Integer arithmetic.

Prerequisite Reading: Chapters 1-5
Revised: June 22, 2021

Background[1]: A disk drive stores and retrieves data on one or more rotating platters coated with magnetic material. There are two read/write heads per platter (one per surface), attached to an arm controlled by an actuator. The actuator is used to position all the heads at a specific radial distance from the spindle, thus selecting one of several concentric tracks on the surface of each platter.

Together, the selected tracks combine vertically to form what is referred to as a cylinder; specifying a cylinder number thus determines the radial position of the arm. The tracks are further divided into sectors of 512 bytes each, representing the smallest amount of data that can be read or written.

Locating a particular sector on the disk requires specifying a cylinder, a head (or surface), and a sector. Cylinders and heads are numbered starting from zero, while sectors are numbered starting from one. Modern disk drives combine the physical cylinder, head and sector numbers into a single number called the Logical Block Address (LBA) given by:

read/write head — spindle — platter — file actuator — arm — power connector — data connector — circuit board
© 2010 Encyclopædia Britannica, Inc.

$$logical\ block\ address\ (lba) =$$
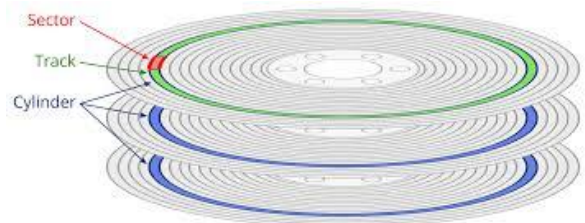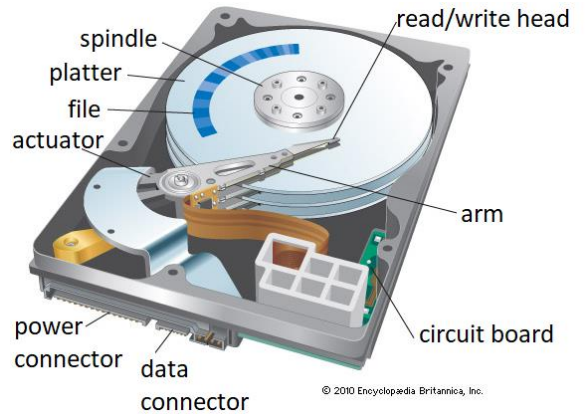$$(cylinder \times \#HEADS + head) \times \#SECTORS + sector - 1$$

Assignment: The main program will compile and run without writing any assembly. However, your task is to create an equivalent replacement in assembly language for the following function found in the C main program that converts a logical block address into the separate components of cylinder, head and sector stored as components of a structure. The original C version has been defined as "weak" so that the linker will automatically replace it in the executable image by the one you create in assembly; you do not need to remove the C version.

Sector — Track — Cylinder

```
typedef struct {uint16_t cylinder; uint8_t head; uint8_t sector;} CHS ;
void Log2Phys(uint32_t lba, uint32_t heads, uint32_t sectors, CHS *phy);
```

The equations for the physical cylinder, head and sector are given by:

$$cylinder = lba \div (heads \times sectors)$$
$$head = (lba \div sectors) \% heads$$
$$sector = (lba \% sectors) + 1$$

The main program allows you to specify the total capacity of the disk by entering values for the total number of cylinders, heads, and sectors. The slider is then used to select a particular logical block address, which the program converts into the corresponding physical coordinates of cylinder, head and sector, and displays them on the screen.

ARM Assembly for Embedded Applications

#Cylinders: − 768 +
#Heads: − 8 +
#Sectors: − 128 +
Capacity: 384.0 MB
LBA = 0200000
0   0786431
Physical Cyl: 195
Physical Head: 2
Physical Sector: 65
Lab 5E: Disk Drive Geometry

---

[1] https://en.wikipedia.org/wiki/Disk_storage