

22C:050 Solutions for HW#9

2.

Assuming that a sector holds 256 bytes and that each sector can hold 64 disk addresses, how many bytes could be stored in a file such as is shown in Figure 12.2, assuming

a) no index sectors?

256 bytes

b) one index sector?

$256 \times 64 = 16,384$ bytes

c) two levels of index sectors?

$256 \times 64 \times 64 = 1,048,576$ bytes

d) three levels of index sectors?

$256 \times 64 \times 64 \times 64 = 67,108,864$ bytes

3.

The classic Unix file system allows only 32 bits (4 bytes) to specify the address of any sector on disk, and each sector holds 512 bytes. What is the maximum storage capacity of a single disk device?

$2^{32} \times 512 = 2,199,023,255,552$ bytes

5.

Directories under both the Xerox Pilot system and Unix map textual file names to numerical names (in Unix, the numerical name is the i-number specifying the i-node that describes the file). Both systems use a tree structure for efficient access to a particular sector of a particular open file. How do these systems differ?

Pilot uses one big search B-tree as an efficient way to access the sectors; while UNIX uses a separate tree for each file. However, the Pilot system uses additional fields to store, attached to each sector of the disk, the file number of the file to which that sector is allocated, and the sector number within that file that the sector represents. Therefore, the Pilot system has a linear search of the disk that can be used to rebuild the system once there is corruption, although it is not efficient.

10.

Write code for the "mark" and "seek" services defined in Figure 12.11, using the data structures shown in Figure 12.9, and taking into account the needs of the code shown in Figure 12.10.

```
void mark(struct filevariable * f )
{
    struct diskfilevariable * df = (struct diskfilevariable *) f;
    return df->pos;
}
```

```
void seek( struct filevariable * f, int i)
{
    struct diskfilevariable * df = (struct diskfilevariable *) f;

    if(i/ df->disk->sectorsize != df->pos/ df->disk->sectorsize)
    {
        free( df->buf );
        df->buf = NULL;
    }

    df->pos = i;
}
```