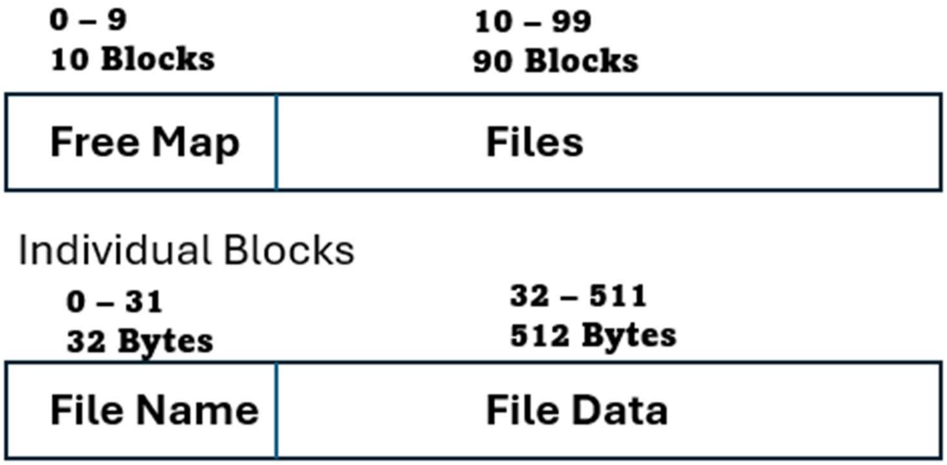
Programming Assignment 6 – File System

In this assignment, you'll be building a simplified, command-line file system. This project will demonstrate how an operating system manages data on a disk, keeping track of what's free and where files are stored.

Your task is to simulate a small disk and implement the fundamental operations of a file system. You will create a program that allows a user to interact with this simulated file system using a set of commands, or "system calls."

Your simulated disk will have 100 blocks, each capable of holding 512 bytes of data, shown below



To manage these blocks, you'll need to create two essential data structures:

- **Free Map:** This is a bit-level map that tells you which blocks on your disk are available and which are in use. The first 10 blocks of your disk are reserved for this map. In your code, you'll represent this map as a simple array or data structure. A 1 will signify an **allocated** (in-use) block, and a 0 will signify a **free** block. Every time a file is created or deleted, you'll update this map.
- **File Table:** This acts as a lookup table for all the files on your disk. Unlike the Free Map, this table is not stored on the simulated disk itself but is loaded into memory when your program starts. You will implement this as an array where each index corresponds to a block number. If a file is stored at block 5, you'll store its filename in fileTable[5].

This filesystem will allow users to read, write, create, and delete files on disk. Because the disk is so small (and to keep it simple), each file is just one block long! There is only one directory(root) defined by this file system and permanently available for users to store their files. No other directories are provided by the system. Users cannot create any directory either. Therefore, all files are maintained in the root directory. On the startup, the cwd is the root directory.

Required Functionality (System Calls):

Your program must present a menu or a command-line interface where users can choose from the following six commands:

1. format: Initialize the entire disk, setting all blocks to a "free" state and clearing the file table.
2. create: Find the first available free block, allocate it, and associate a new filename with it in your file table.
3. read: Given a filename, find its block, and print its contents to the screen.
4. write: Given a filename and some data, find the file's block and write the new data to it.
5. delete: Given a filename, find its block, free the block in your Free Map, and remove the file from the file table.
6. dir: List all the files currently stored in your file system.

You must handle errors gracefully (e.g., what happens if a user tries to delete a file that doesn't exist?).