

File System Implementation

COMP755

C review

- A star before a variable means an address
`void writeDisk(int location, void *bytes, int length)`
- `*bytes` is a pointer
- You can specify the address of a variable by putting a "&" before the variable

```
int buf, len;  
len = readDisk(0, &buf, 4)
```
- The second parameter is the address of `buf`

C Data Size

In most systems:

- long 64 bits
- int 32 bits
- short 16 bits
- char 8 bits -128 to 127
- unsigned char 8 bits 0 to 255

void

The keyword void has two meanings in C

- If a method returns void, then it does not return a value
- A *void pointer is a pointer to anything
 - C has weak typing

Struct

- C has the concept of a structure or struct
- A struct is like a class with no methods and all fields are public

```
struct inode { // struct definition
```

```
    int fileSize;
```

```
    char name[17];
```

```
};
```

```
struct inode rootnode; // struct use
```

Struct Access

- A data field of a struct can be accessed like a class field

```
struct inode myNode;
```

```
myNode.size = 47;
```

- If you have a pointer to struct, then you access the field relative to the pointer

```
struct inode *nextNode;
```

```
nextNode->size = 42;
```

How Free Space is Managed

- Popular options
 - bit map – An area in the beginning of the disk is reserved to hold a set of bits, one bit per cluster.
 - linked list of clusters – A list of the available clusters
 - extents – A linked list of cluster groups including the number of clusters in the group and a link to the next group.

The Bitmap Release Function

- Assume a global array containing the bit map

```
char freeSpace[1024];
```

- A 1 bit means the cluster is available and a zero means the cluster is in use.
- Consider:

```
void relBlock( int clusterNum )
```


Possible Implementations

```
void relBlock( int cn ) {  
    freeSpace[cn/8] |= 1 << (cn%8);  
}
```

/ efficient but less clear solution */*

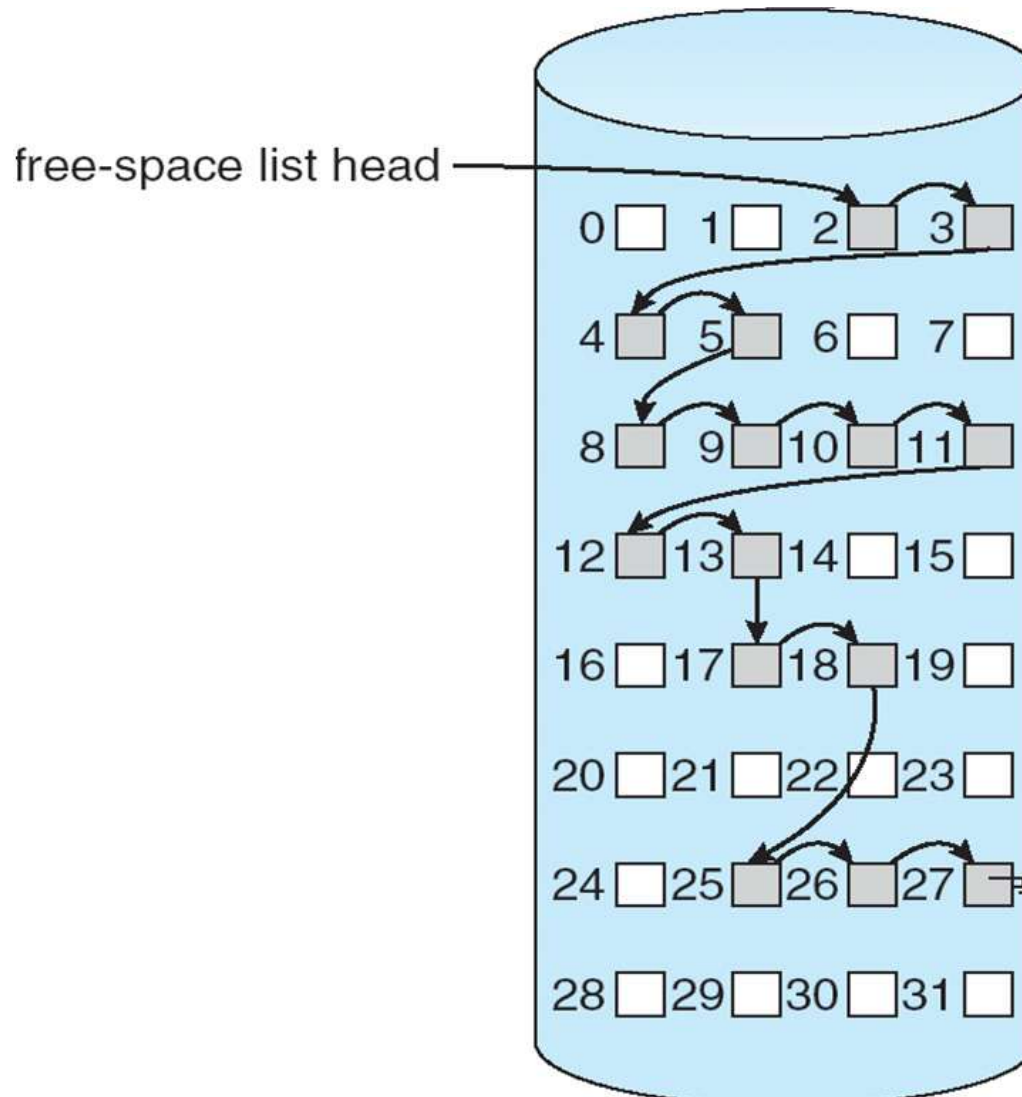
```
void relBlock( int cn ) {  
    freeSpace[cn>>3] |= 1 << (cn&7);  
}
```

Simulation Hint

- A bitmap can be implemented as an array of 256 boolean variables
- It uses more space, but that is probably not an issue



Linked Free Space List on Disk



Linked List Free Space

- During system initialization, all of the available blocks are linked together

```
int next;           // address of next block
for (int blk = 0; blk < 255; blk++) {
    next = blk + 1;
    writeDisk(blk, &next, sizeof next);
}
next = 0;          // zero means end of list
writeDisk( 255, &next, sizeof next); // end of list
```

Information in an Inode

- The Inode keeps the metadata about a file.
 - filename
 - size
 - access rights, owner, time stamps, use count, etc
- Directories are files and have Inodes
 - Flag that the Inode is for a directory
- Additional information may also be stored in the Inode

Inode in RAM

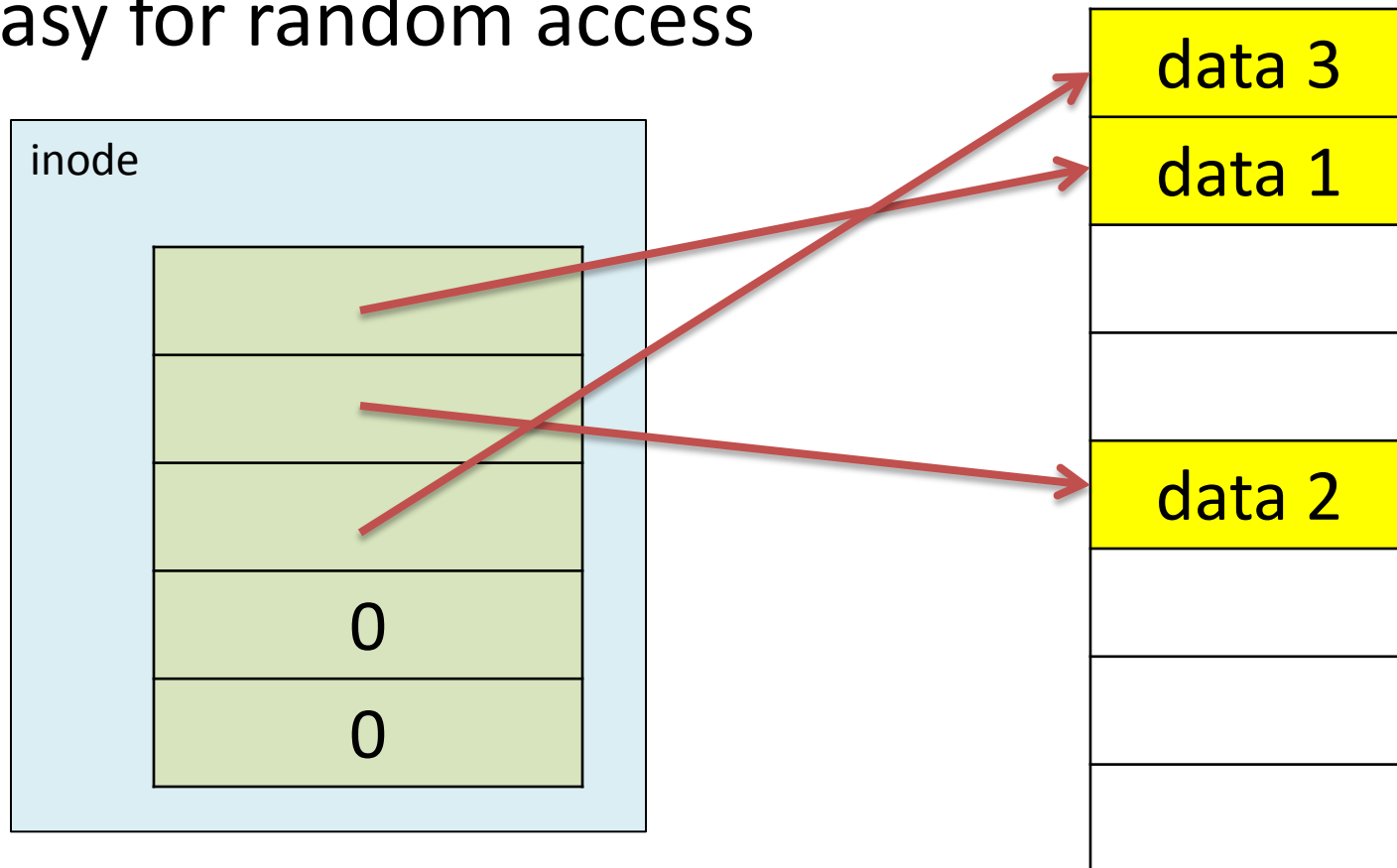
- There may be fields in the inode that only have meaning when the inode is in RAM
- Examples of such data might be:
 - Location of current data block in RAM
 - Number of current data block

Where is the Data?

- The data or text of the file is located in (possibly many) data blocks on the disk.
- Generally the Inode will contain information about location of the data blocks.
- Popular options
 - array of data disk addresses
 - linked list of disk blocks

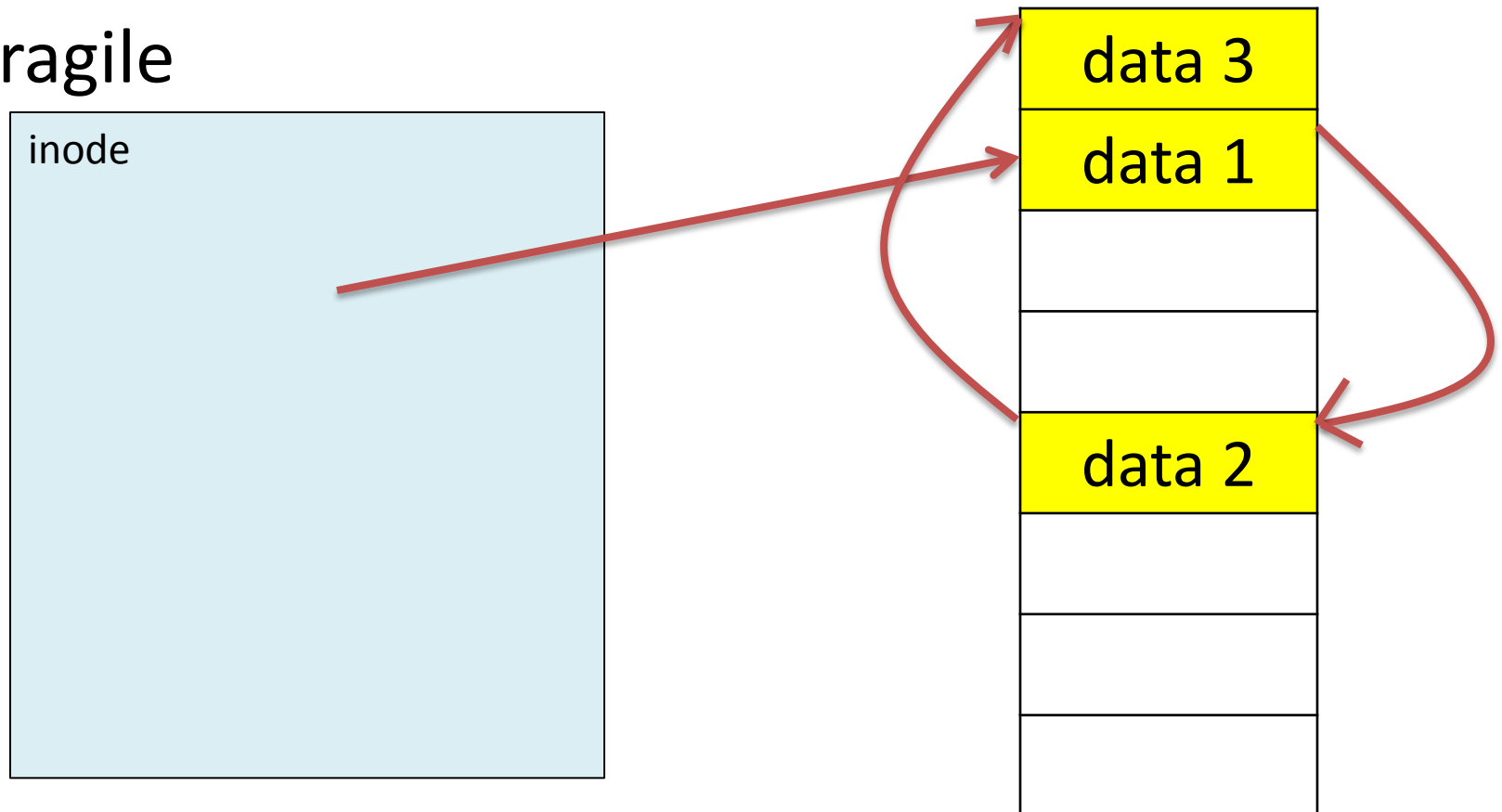
Array of Disk Addresses

- Limits the size of a file
- Easy for random access



Linked List of Data Blocks

- Uses space in data block
- Easy for sequential access
- Fragile



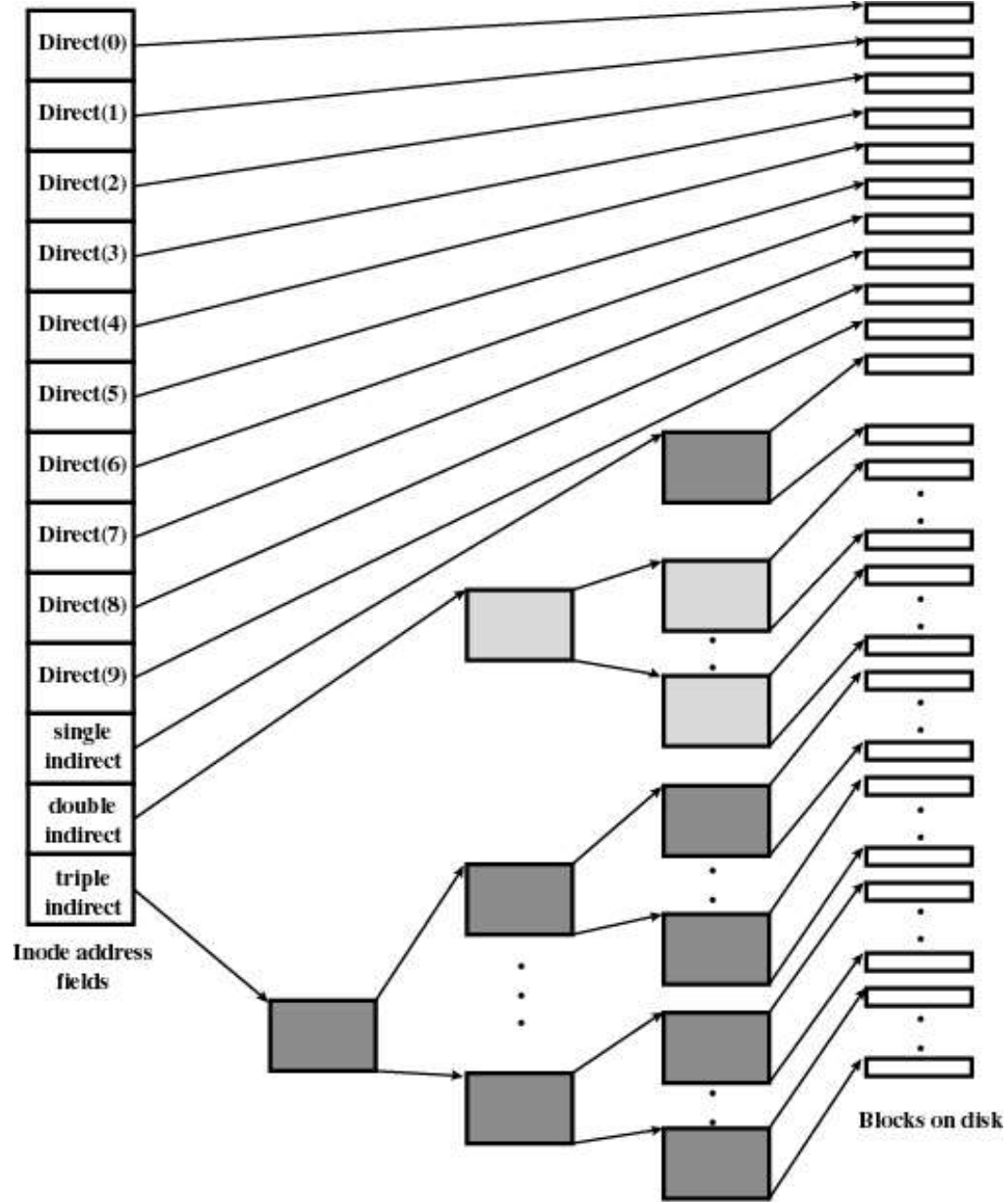


Figure 12.13 UNIX Block Addressing Scheme

What is in a Directory?

- A directory contains the list of files and other directories that are contained in it.
- If you have `/dog/cat` and `/dog/aardvark`, somewhere you need a list of `cat` and `aardvark`
- A hash table can be used instead of a list

name	inode
cow.txt	25
aardvark	31
HW1.doc	17
cat	122

Where to Keep Directory Lists

- Popular options for the directory list
 - A directory is a file where the “text” of a directory contains the list of names and pointers (*Unix method*)
 - The list is kept in the Inode
 - Linked list of Inodes
 - Hash table

Directory Files

- In Unix a directory is very much like a regular file except that the data of the file is a list of names and pointers to inodes
- Because each directory has an inode, each directory can have individual access rights
- Multiple access are required to find a file

Looking for a File

directory files

Consider reading the text of [/dog/cat/goat.txt](#)

- Get the root inode
- Check the inode access rights
- Get the root directory file
- Search the directory for dog
- Get the dog inode
- Check the dog inode access rights
- Get the dog directory file

Still Looking

directory files

- Search the dog directory for cat
- get the cat inode
- Check the cat inode access rights
- Get the cat directory file
- Search the directory for goat
- Get the goat inode
- Check the goat inode access rights
- Get the goat data

Directories in the Inode

- Convenient because the Inode is probably already loaded
- Limits the number of files and sub directories that can be in a directory

Looking for a File

directory in Inode

Consider reading the text of `/dog/cat/goat.txt`

- Get the root inode
- Check the inode access rights
- Search the directory list for dog
- Get the dog inode
- Check the dog inode access rights

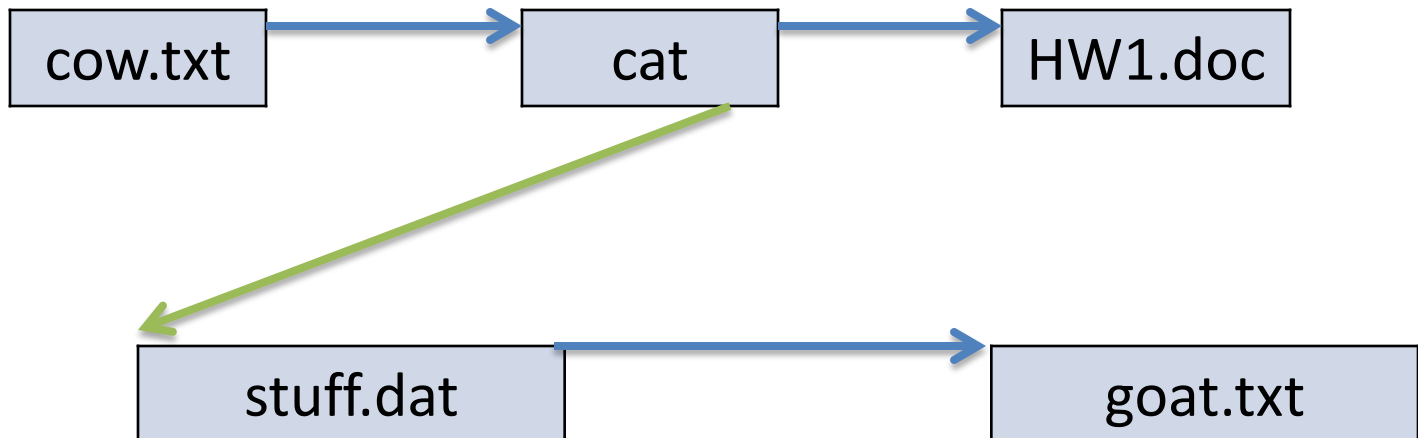
Still Looking

directory in Inode

- Search the dog directory list for cat
- get the cat inode
- Check the cat inode access rights
- Search the directory list for goat
- Get the goat inode
- Check the goat inode access rights
- Get the goat data

Linked List of Inodes

- Easy to add or delete items
- Fragile



Looking for a File

linked Inodes

Consider reading the text of [/dog/cat/goat.txt](#)

- Get the root inode
- Check the inode access rights
- do {
- Get the next inode in the list
- } while not dog inode or end of list
- Check the dog inode access rights

Still Looking

Linked Inodes

- do {
- Get the next inode in the list
- } while not cat inode or end of list
- Check the cat inode access rights
- do {
- Get the next inode in the list
- } while not goat inode or end of list
- Check the goat inode access rights
- Get the goat data

Hash Table Directories

- A possible implementation is to have all files in one hash table
- The full path and filename are hashed to find the location of the inode
- This restricts the ability to have access rights on directories
- It is difficult to list all files in a directory

Looking for a File

hashed Inodes

Consider reading the text of `/dog/cat/goat.txt`

- Hash the path and filename
- Get the inode of `goat.txt`
- Check the `goat.txt` inode access rights
- Get the goat data

File Handles

- The open function returns an integer file handle. This is used by other functions to identify the file.
- You will need to keep a list of open files
- The file handle is an index into the list of open files
- Often the file handle array contains a pointer to the inode for the file in RAM
- You might also want to open directories

Writing Data to a File

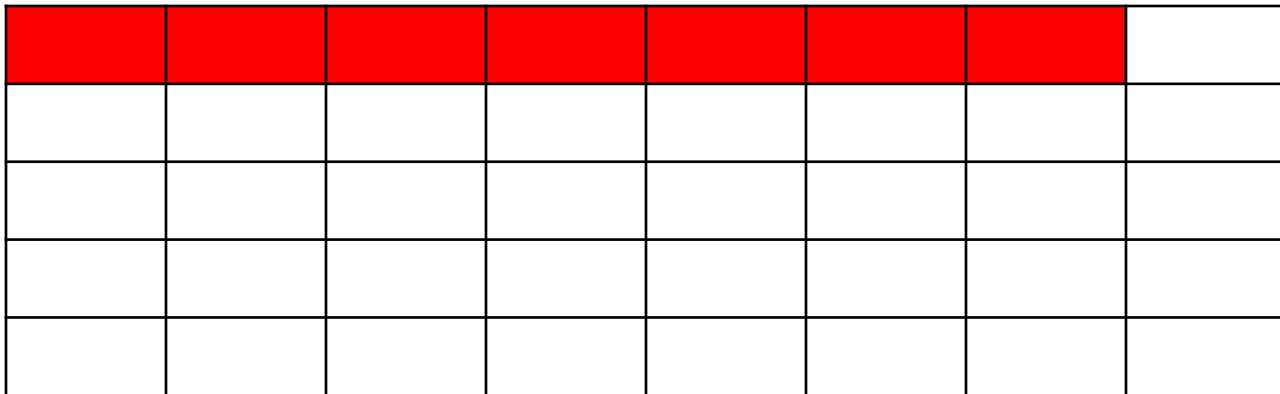
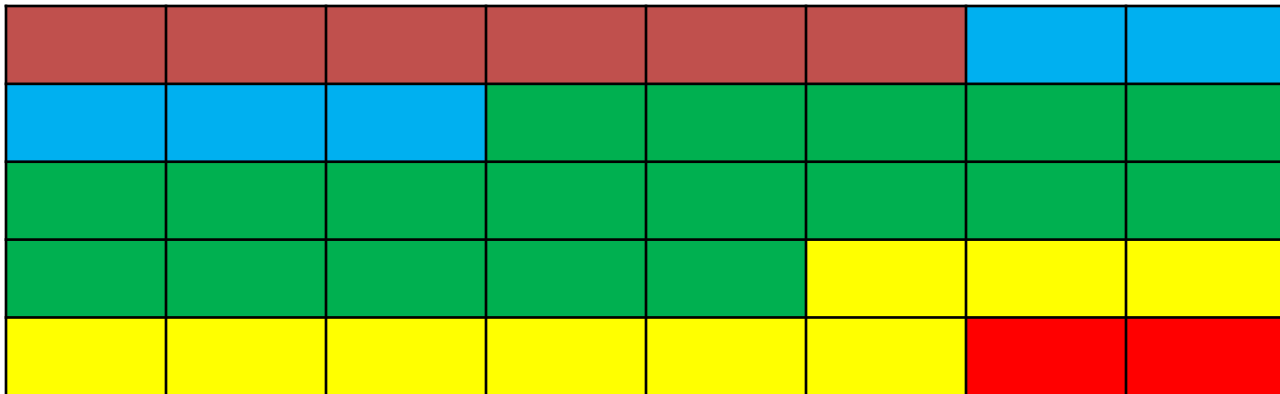
- The `simwrite` function writes one line of text to a file.
- It will probably take several calls to `simwrite` for the data to fill a disk block.
- The disk block will be written only when it is full.
- Lines of text should be separated by a newline or something.

Overlapping a Data Block

- It is likely that some line of text will start in one data block and flow into another
- The first part of the data should be copied to one block, the block written to disk and then the remainder of the text line copied to the beginning of the data block

Data in a Block

- When the buffer is full, the remaining bytes should be in a second block



Bookkeeping

As you write data:

- Keep track of the file size in the Inode
- Keep track of the disk location of each block
- Record which block you are writing
- Keep the address of the block in RAM
- Keep a pointer into the block so you know where the next line will start

Closing a File

- You may need to write the last block of data to the disk.
- Release the memory space for the
 - data block
 - Inode
- Remove file from list of open files

Disk Layout

- The beginning of the disk usually contains information about the file system
 - Location of root inode
 - Location of free block data structure
 - Size of file system
- When an existing file system is used, the OS reads this data into RAM
- This data is written to the disk when the file is initialized

Mixing Inodes and Data

- Some file systems (particularly those with more complex directory structures) attempt to keep the inodes together on the disk
- Other systems mix inodes and data throughout the disk

Register

- Make sure you register soon
- Classes with low enrollment may be cancelled



VOTE

