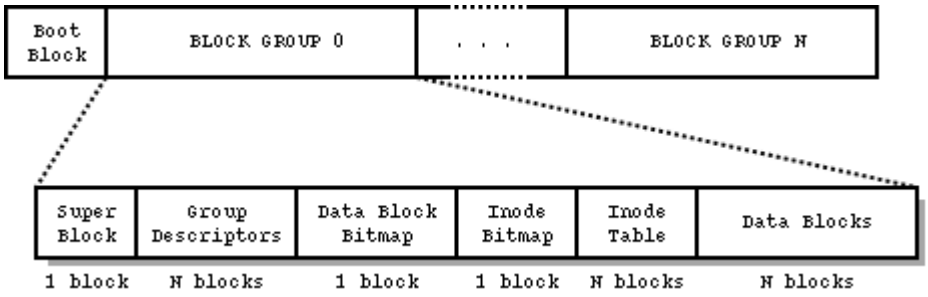
Ext2 FileSystem Structure



- Ext2 or extended files system is a file system for the kernel.
- Block Group: made up of logical blocks of continuous space
  - **Superblock**: contains information such as the total number of blocks on disk, the size of a block (usually 1024 bytes), the number of free blocks, etc.
    - s_inodes_count and s_blocks_count store the number of inodes and blocks on disk
  - **Group Descriptors**: tells us the location of the block/inode bitmaps and of the inode table through the bg_block_bitmap, bg_inode_bitmap and bg_inode_table fields. These values indicate the blocks where the bitmaps and the table are located.
  - **Data Block Bitmap**: bitmap is a sequence of bits. Each bit represents a specific block. A bit value of 0 indicates that the block is free, while a value of 1 indicates that the block is being used.
  - **Inode Bitmap**: bitmap is a sequence of bits. Each bit represents a specific inode in the block group. A bit value of 0 indicates that the inode is free, while a value of 1 indicates that the inode is being used.
  - **Inode Table**: consists of a series of consecutive blocks, each of which contains a predefined number of inodes. The block number of the first block of the inode table is stored in the bg_inode_table field of the group descriptor.
    - *Number of inodes per block*: `unsigned int inodes_per_block = block_size / sizeof(struct ext2_inode);`
    - *Size in blocks of the inode table*: `unsigned int itable_blocks = super.s_inodes_per_group / inodes_per_block;`
    - imode: the type and access rights of a file. S_ISREG is regular file. S_ISDIR is directory
    - i_blocks: counts the number of blocks used by the file
    - i_block[EXT2_N_BLOCKS]: pointers to the actual data blocks of the file
      - 0..11: point directly to the first 12 data blocks of the file.
      - 12: points to a single indirect block
      - 13: points to a double indirect block
      - 14: points to a triple indirect block
  - Data Blocks

Reference

- http://cs.smith.edu/~nhowe/262/oldlabs/ext2.html

```
#include <stdio.h>
#include <unistd.h>
```

```c
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <string.h>

#include "ext2.h"

unsigned char *disk;

int main(int argc, char **argv) {
  if(argc != 2) {
    fprintf(stderr, "Usage: readimg <image file name>\n");
    exit(1);
  }

  int fd = open(argv[1], O_RDWR);
  disk = mmap(NULL, 128 * 1024, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);

  if(disk == MAP_FAILED) {
    perror("mmap");
    exit(1);
  }

  // Task 1
  struct ext2_super_block *sb = (struct ext2_super_block *)(disk + 1024);
  struct ext_group_desc *gd = (struct ext2_group_desc *)(disk +
EXT2_BLOCK_SIZE * 2);
  unsigned char *block_bitmap = (char *)(disk + EXT2_BLOCK_SIZE * gd-
>bg_block_bitmap);
  unsigned char *inode_bitmap = (char *)(disk + EXT2_BLOCK_SIZE * gd-
>bg_inode_bitmap);

  printf("Inodes: %d\n", sb->s_inodes_count);
  printf("Blocks: %d\n", sb->s_blocks_count);
  printf("Block group:\n");
  printf("\tblock bitmap: %d\n", gd->bg_block_bitmap);
  printf("\tinode bitmap: %d\n", gd->bg_inode_bitmap);
  printf("\tinode table: %d\n", gd->bg_inode_table);
  printf("\tfree blocks: %d\n", gd->bg_free_blocks_count);
  printf("\tfree inodes: %d\n", gd->bg_bree_inodes_count);
  printf("\tused_dirs: %d\n", gd->bg_used_dirs_count);

  // Task 2
  int byte, bit;

  printf("Block bitmap: ");
  for (byte = 0; byte < sb->s_blocks_count / 8; byte++) {
    for (bit = 0; bit < 8; bit++) {
      printf("%d", (block_bitmap[byte]& (1 << bit)) & 1);
    }
    printf(" ");
  }
```

```
    printf("\n");

    printf("Inode bitmap: ");
    for (byte = 0; byte < sb->s_inodes_count / 8, byte++) {
      for (bit = 0; bit < 8; bit++) {
        printf("%d", (inode_bitmap[byte]& (1 << bit)) & 1);
      }
      printf(" ");
    }

    return 0;
  }
```