ECE 598 – Advanced Operating Systems
Lecture 15

Vince Weaver

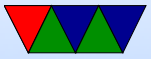
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

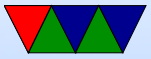
24 March 2015

Announcements

- Homework #4 was posted, due Wednesday
- Am going to try to catch up so we have all 10 HWs described in syllabus. Be sure you do them! Each worth 6% of final grade.
- Interesting talk by someone at Intel at 11am in Barrows 130.

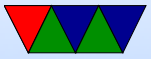


Filesystems

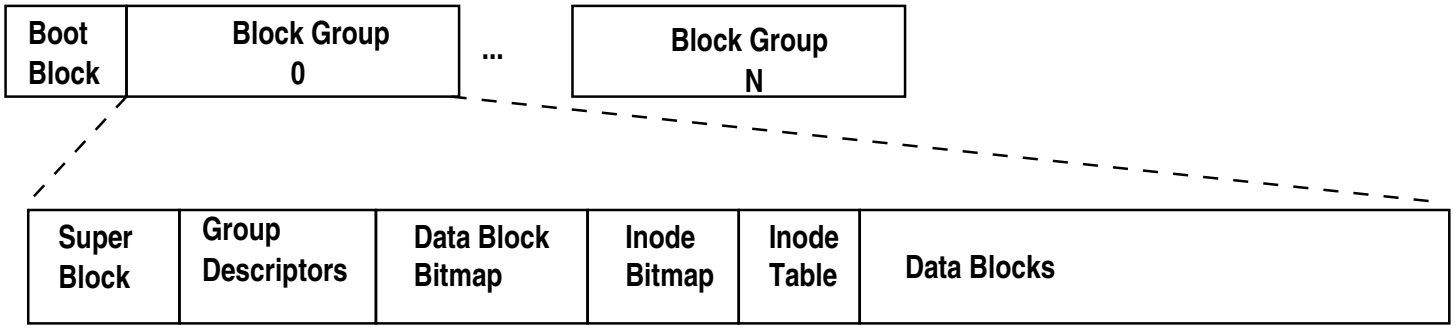


Ext2 FS

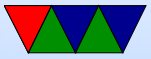
- All structures are little-endian (To aid in moving between machines)
- Block size 1024-4096 (for various reasons it's complicated on Linux to have a block size greater than the page size)
  (also, does blocksize have to be power of 2? Some CD-ROMs had blocksize of 2336 bytes)
- Format



– Boot sector, boot block 1, boot block 2, boot block 3



– Block group: superblock, fs descriptor, block bitmap, inode bitmap, inode table, data blocks
– Superblock – located at offset 1024 bytes. Copies scattered throughout (fewer in later versions)
  Info on all the inode groups, block groups, etc.
– Block descriptor table – description of how disk is split up into block groups



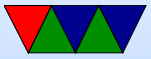
each group has own block descriptor

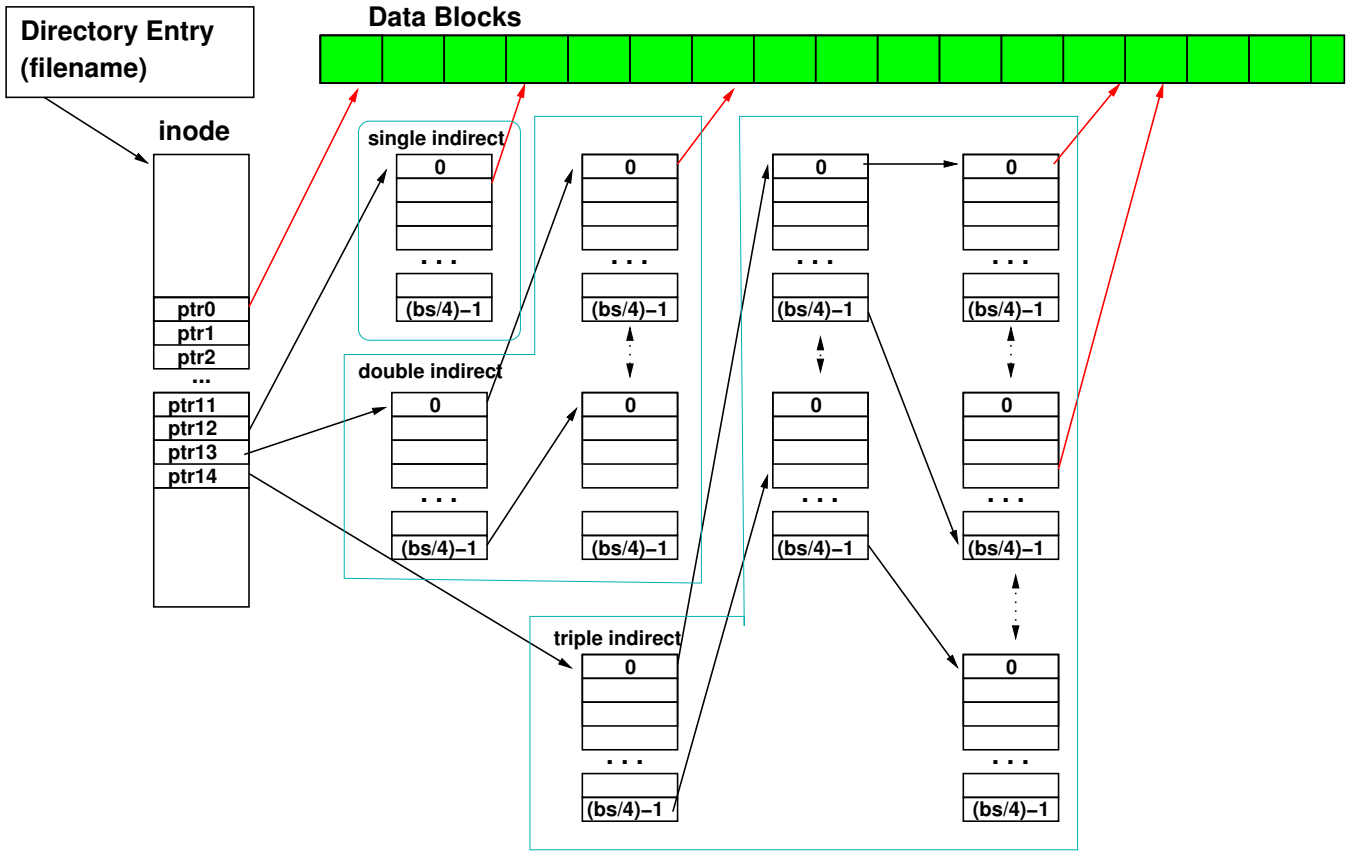
Says where the various following things are, and has file counts.

- Block bitmap – bitmap of blocks (1 used, 0 available) block group size based on bits in a bitmap. if 4kb, then 32k blocks = 128MB.
- Inode bitmap – bitmap of available inodes
- Inode table – all metadata (except filename) for file stored in inode

  Second entry in inode table points to root directory

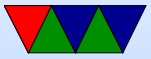
  inode entries are 128 bytes.



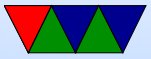


– Directory info –

Superblock links to root directory (usually inode 2)



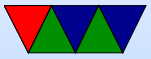
Directory inode has info/permissions/etc just like a file
The block pointers point to blocks with directory info.
Initial implementation was single linked list. Newer use hash or tree.
Holds inode, and name (up to 256 chars). inode 0 means unused.



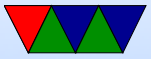
type	size
inode of file	4
size of entry	2
length of name	1
file type	1
file name	N

- Hard links – multiple directory entries can point to same inode

- . and .. entries, point to inode of directory entry



- Subdirectory entries have name, and inode of directory

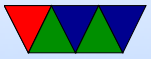


Ext3/Ext4

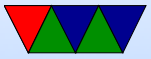
- Compatible with ext2

- ext3
  - Htree instead of linked list in directory search
  - online fs growth
  - journal
  - Journal

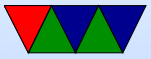
    metadata and data written to journal before commit. Can be replayed in case of system crash.



- ext4
  - Filesize up to 1Exabyte, filesize 16TB
  - Extents (Rather than blocks) , an extent can map up to 128MB of contiguous space in one entry
  - Pre-allocate space, without having to fill with zeros (which is slow)
  - Delayed allocation – only allocate space on flush, so data more likely to be contiguous
  - Unlimited subdirectories (32k on ext3 and earlier)
  - Checksums on journals
  - Improved timestamps, nanosecond resolution, push

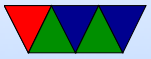


beyond 2038 limit

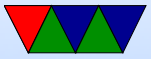


btrfs

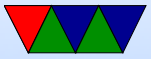
- B-tree fs (binary tree)
- Copy on write. When write to a file, old data not overwritten. Since old data not over-written, crash recovery better
  Eventually old data garbage collected
- Data in extents
- Copy-on-write



- On-line defragmentation
- On-line volume growth
- Built-in RAID
- Transparent compression
- Snapshots
- Checksums on data and meta-data
- De-duplication

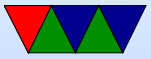


- Cloning – can make an exact snapshot of file, copy-on-write
  different than link, different inodles but same blocks



Embedded

- Designed to be small, simple, read-only?
- romfs
  - 32 byte header (magic, size, checksum,name)
  - Repeating files (pointer to next [0 if none]), info, size, checksum, file name, file data
- cramfs



ZFS

Advanced OS from Sun/Oracle. Similar in idea to btrfs