OS\_Project#4 File System

2013012148 Lee Jaeil

**Project4 File System**

**Goal of Project4**

Current xv6 fs inode has 12 direct address and one indirect address. By Adding double-indirect address and changing some variables, bmap and itrunc, we can extend max file size.

**1. setting**

1)in param.h  
Change FSSIZE 1000 to 2000(to test more huge file which size is more than current max file size).  
2)in fs.h  
Change NDIRECT to 11. Struct dinode's size is 64bytes, which means that one inode block can contain eight inode struct. to keep this situation, we subtract one direct address and add one double-indirect address to addrs array.  
Define NDINDIRECT (NINDIRECT \* NINDIRECT). This means that one double-indirect address can contain NINDIRECT\*NINDIRECT data blocks.  
Change MAXFILE to (NDIRECT+NINDIRECT+NDINDIRECT).  
Change addrs[NIRECT+1] to addrs[NDIRECT+2], which means adding one double-indirect address.  
3)in file.h  
Change addrs[NDIRECT+1] to addrs[NDIRECT+2]

**2. bmap**

Function bmap returns the disk block address of the nth block in inode ip. If there is no such block, bmap allocates one.  
Add case of using double-indirect address.  
1) if bn is not handled in NINDIRECT case, subtract NINDIRECT from bn.  
2) set if-statement to handle file of which size is more than NINDIRECT + NDIRECT blocks.  
3) load double-indirect block, allocating if necessary.  
if double-indirect block is empty, set block by balloc(ip->dev). Read the buffer by bp = bread(ip->dev, addr). read data of buffer by saving (uint\*)bp->data.  
4) read indirect block(allocating if necessary).  
As bp is handled by double-indirect block, select indirect block at first. By doing "bn/NINDIRECT", we can determine indirect block number.  
5) read data block(allocating if necessary).  
we can determine data block by doing bn%NINDIRECT.  
6) log\_write and brelse.  
To record the block number in memory to reserve a slot in the log on disk, do log\_write(dbp) and log\_write(bp)(dbp is double-indirect block and bp is indirect block). And when job is done, to release locked buffer, do brelse(dbp) and brelse(bp).

**3. itrunc**

Function itrunc is called when the inode has no links to it(no directory entries referring to it) and has no in-memory reference to it(is not an open file or current directory).  
Make if-statement to handle double-indirect block.  
1) load double-indirect block.  
read the buffer by doing bp = bread(ip->dev, ip->addrs[NDIRECT+1]). and read data of buffer by doing a = (uint\*)bp->data;  
2) run for-loop to free indirect blocks.  
run for loop(0~NINDIRECT). if indirect block is empty, continue. If not, read indirect block and free all data blocks(by doing bfree(ip->dev, a[j]) which indirect block points. and free the indirect block.  
3) brelse buffers.  
when brelse indirect block, do brelse after all data blocks which the indirect block points are freed. And when brelse double-indirect block, do brelse after all indirect blocks which the double-indirect block points are freed.