CS 111 week 7
Project 3B: File System Audit

Tianxiang Li

Project 3B: Overview

Write a program to identify inconsistencies/errors given a disk image

essentially fsck(file system consistency check) without fixing the inconsistencies/errors

- Input: *.csv files
- Output: Inconsistencies/errors on:
  - Data Block Number
  - I-node
  - Directory

- Note: the program can be written in any language you prefer.

Basic Logic

- Read csv file, parse its data
- Look for keywords: INODE, SUPERBLOCK, …
- Extract important fields: numBlocks in SB, …
- Check columns of each row, detect inconsistency and errors

Data Block Number Errors

```
struct ext2_inode {
        __u16  i_mode;          /* File mode */
        …
        __u32   i_block[EXT2_N_BLOCKS];  /* Pointers to blocks */
}
```

Possible Errors:

- **Invalid**: Block Number < 0, > max block number (got from SB)
- **Reserved**: Block Number is used (by boot block, SB, BGT, inode bitmap, block bitmap, inode table).
- **Unreferenced**: Not referenced by any file but marked as allocated on block bitmap
- **Allocated**: Allocated to a file but marked as free on block bitmap
- **Duplicate**: Used by more than two files.

Data Block Number Errors: Pseudo Code

```
max_block = SB. s_blocks_count; //max block number from SB
orig_block_bitmap = block_bit_map; //block bitmap (locate using group descriptor)
memset(my_block_bit_map, 0, sizeof(my_block_bitmap); //calculate bitmap based on i_block
reserved_bit_map = calc_reserved_bit_map(); //block numbers used by Boot Block, SB, ....
for every inode {
        if (inode not used) continue;
        for every data blocks in inode {
                if (block_number < 0 || block_number > max_block)  report INVALID
                if (block_number in reserved bit map)  report RESERVED
                if (block_number is free in orig_block_bitmap) report AllOCATED
                if (block_number is marked as used in my_block_bit_map)
                        report DUPLICATED
                Mark block number as used in my_block_bit_map;
        }
}
```

Data Block Number Errors: Pseudo Code (Cont)

```
for every block in my_block_bit_map {
        if (block number is marked as used in orig_block_bitmap &&
        block number is free in my_block_bit_map )
                report Unreferenced // Not referenced by any file but marked as allocated on block bitmap
}
```

Inode Errors

Scan through all of the I-nodes in .csv to determine which are allocated.

Compare your list of allocated/unallocated I-nodes with the free I-node bitmaps.

```
struct ext2_inode {
        __u16  i_mode;          /* File mode */
        …
        __u32   i_block[EXT2_N_BLOCKS];  /* Pointers to blocks */
}
```

if (i_mode == 0) inode is free; else inode is used.

Possible Errors:

- **Allocated**: i_mode != 0 but marked as free on inode bitmap
- **Unallocated**: i_mode == 0 but marked as used on inode bitmap

Inode Errors: Pseudo Code

```
orig_inode_bitmap = inode_bit_map; //inode bitmap (locate using group descriptor)
for every inode {
        if (inode.i_mode != 0 && inode marked as free in orig_inode_bitmap)
                report ALLOCATED;
        else if (inode.i_mode == 0 && inode marked as used in orig_inode_bitmap)
                report UNALLOCATED;
}
```

Directory Errors

```
struct ext2_dir_entry {
        __u32   inode;                          /* Inode number */
        …
        char    name[EXT2_NAME_LEN];  /* File name */
};
struct ext2_inode {
        …
        __u16   i_links_count;      /* how many times this particular inode is linked (referred to). */
        …
}
```

Possible Errors:

- **Incorrect link count**: Number of dir_entry pointing to the inode is not the same as i_links_count
- **Unallocated**: inode referenced in dir_entry is marked as free on inode bitmap
- **Invalid**: inode referenced in dir_entry is < 0 or > maximum inode number (got from SB)
- . is not pointing to the current dir.
- .. is not pointing to the parent dir.

Directory Errors: Pseudo Code

```
max_inode = SB. s_inodes_count;
memset(inode_ref_array, 0, sizeof(inode_ref_array); //calculate ref count based on directory
memset(inode_par_array, 0, sizeof(inode_par_array); //store the parent of each inode
for every inode {
        if (inode not directory) continue;
        par_ino = inode.i_ino;   //inode number of current dir (parent inode)
        for every directory entry in inode {
                child_ino = dir_entry.inode; //inode number of each child entry in the current dir
                child_name = dir_entry.name; //file or dir name of child entry
                if (child_ino < 0 || child_ino > max_inode ) report INVALID
                if (child_ino is free in inode_bitmap) report UNALLOCATED
                if (child_name is . && child_ino != par_ino) report CURRENT_MISMATCH;
                if (child_name is not . && child_name is not ..)
                        inode_ref_array[child_ino]++;
                inode_par_array[child_ino] = par_ino;
        }
}
```

	inode	rec_len	name_len	file_type	name							
offset: 0	13	12	1	2	.	\0	\0	\0				file: .
offset: 12	10	12	2	2	.	.	\0	\0				file: ..
offset: 24	18	16	5	2	m	u	s	i	c	\0	\0	file: music
offset: 40	15	16	8	1	t	e	s	t	.	t	x	t	file: test.txt
offset: 56	19	12	3	2	b	i	n	\0				file: bin

Directory Errors: Pseudo Code (Cont)

```
for every inode {
        par_ino = inode.i_ino; ;   //inode number of current dir (parent inode)
        if (inode_ref_array[par_inode] != inode.i_links_count) //check hard links count
                report INCORRECT_LINK_COUNT
        if (inode not directory) continue;
        for every directory entry in inode {
                child_ino = dir_entry.inode; //inode no. of each child entry in the current dir
                child_name = dir_entry.name;
                if (child_name is .. && child_ino != inode_par_array [par_ino])
                        report PARRENT_MISMATCH;
}
```

The links count we calculated

The links count field of this inode

	inode	rec_len	name_len	file_type	name								
offset: 0	13	12	1	2	.	\0	\0	\0					file: .
offset: 12	10	12	2	2	.	.	\0	\0					file: ..
offset: 24	18	16	5	2	m	u	s	i	c	\0	\0	\0	file: music
offset: 40	15	16	8	1	t	e	s	t	.	t	x	t	file: test.txt
offset: 56	19	12	3	2	b	i	n	\0					file: bin

End of Discussion

- Good luck on the Project