

# Assignment 6 – Inodes and links

## Question 1: FS inodes (6 points)

### 1.a toy-fs (2 points)

Here is a toy-fs:

- it is built on a disk with 4KB blocks
- toy-fs allocates 100 blocks for storing inodes
- each inode is 128B in size

**What's the max number of files and dirs the toy-fs can have?**

### 1.b toy-fs inode (2 points)

For the same toy-fs, it uses linked lists to track disk blocks for file mapping (file mapping maps from file offset to the block number). Below is the definition of a toy-fs inode:

```
struct toyfs_inode {
    ... /* metadata */
    uint32_t ptr; /* pointer to the first data block:
                  "ptr" contains the block number */
};

struct datablock {
    char data[4096 - 4]; /* the file contents */
    uint32_t ptr; /* pointer to the next data block
                  for the final block, "ptr" must be NULL */
};
```

For a 4MB file in toy-fs, **how many blocks it consumes?**

**What's the metadata overhead of the file?****Write the overhead down in percentage with 1 decimal place.**

[notes:

- "metadata overhead" means the size of metadata over the size of the actual file (excluding metadata).
- what is "metadata"? please google. (hint: the metadata here has two parts.)
- as mentioned in (1.a), the inode is of size 128B.

]

**1.c fs3650 inode (2 points)**

Below is the inode's definition in lab4.

Each block in fs3650 is 4KB in size (FS\_BLOCK\_SIZE).

```
#define FS_BLOCK_SIZE 4096

struct fs_inode {
    uint16_t uid;    /* metadata */
    uint16_t gid;    /* metadata */
    uint32_t mode;   /* metadata */
    uint32_t ctime;  /* metadata */
    uint32_t mtime;  /* metadata */
    int32_t  size;   /* metadata */
    uint32_t ptrs[FS_BLOCK_SIZE/4 - 5]; /* pointers to data blocks */
    /* inode = 4096 bytes */
};
```

**What's the maximum size (in KB) of fs3650 files?**

## Question 2: Hard links and soft links (4 points)

In class, we introduce hard/soft links:

\* hard link: `$ ln /tmp/a /tmp/hardlink`

\* soft link: `$ ln -s /tmp/a /tmp/softlink`

Can you create dir cycles with links?

By “dir cycles”, I mean  $a \rightarrow b \rightarrow c \rightarrow \dots \rightarrow a$ , where

–  $a/b/c/\dots$  are directories

– “ $\rightarrow$ ” means being a child in the fs hierarchical tree. For example, “ $a \rightarrow b$ ” means dir “a” has dir “b” as its child. In other words, one can “`cd /path_before_a/a/b`”.

### 1.a (2 points)

Can you create dir cycles by soft links only?

**If yes, write down shell cmds to create a cycle.**

**If no, explain why in one sentence.**

[hint: try (1) create two dirs (say A and B);

(2) create a soft link of A in B;

(3) create a soft link of B in A;

(4) checkout if this is a cycle by `cd` into A’s and B’s link]

1.b (2 points) Can you create dir cycles by hard links only?

**If yes, write down shell cmds to create a cycle.**

**If no, explain why in a sentence.**