CI583: Data Structures and Operating Systems
File systems: directories

Efficient directories

As a last word on file systems, we will briefly look at how directories might be implemented in a modern system.

In simple file systems such as `S5FS`, a directory is an ordinary file that maps some file names to metadata (file size, starting block, ownership, etc), listed in the order in which the files were added.

Efficient directories

This works OK when directories contain only a small number of files.

Even when directories are small, how should we maintain these directory files when files are added to and removed from the directory?

If the first 3 files are removed from the directory, is that space now lost due to fragmentation?

S5FS ignores these issues (and was right to do so – read about *worse is better*!).

`https://en.wikipedia.org/wiki/Worse_is_better`

Efficient directories

Times have changed, however.

Directories might contain thousands of files.

Organising them in this linear (O(n)) way means, in the worst case, searching to the end of the list every time we add, rename, or otherwise access a file.

Efficient directories

There are two main approaches to providing more efficient directories:

1. hashing: construct a function, f, such that $f(/home/me/myfile)$ returns the block that contains the directory entry for $/home/me/myfile$.

   The details of this approach get complicated when we consider adding and removing items from directories, which require us to modify the hashing function (and make sure it continues to work for the previous cases),

Efficient directories

2. indexes: represent the contents of a directory as an efficient data structure of the type used in an RDBMS.

Efficient directories

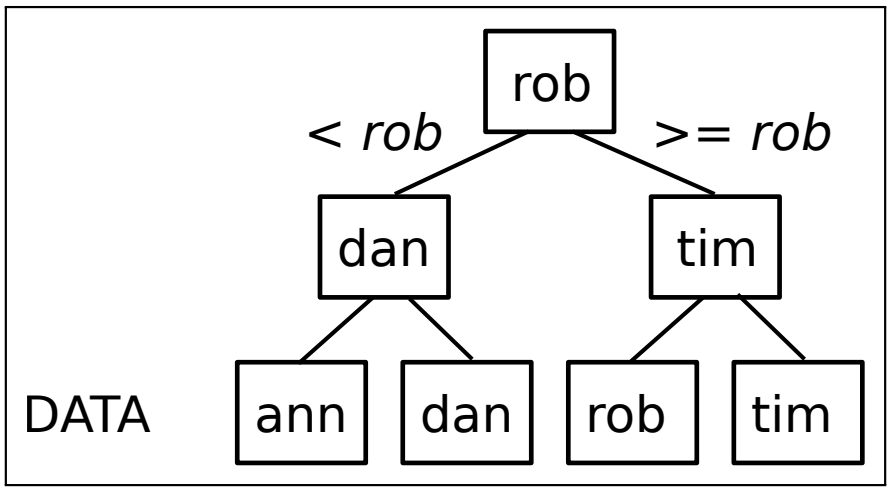
Indexing involves uses a balanced search tree to represent the contents of a directory.

The tree is *balanced* because each leaf node is the same distance from the root.

Searching such a tree is very efficient, but insertion and deletion are more expensive operations.

Efficient directories

The most common type of balanced tree used in file systems is the B+ tree. Only the leaves contain data, which are blocks containing directory entries. This figure is an over-simplification, but conveys the idea.



Demo

Next time

Memory management: virtual memory, page tables, etc.