

archive.today

webpage capture

Saved from

https://commentsarchive.softf1.com/index.php?title=C_0000007_mmmv_btreetlike_folder_structure_t1

search

19 Dec 2020 04:44:28 UTC

no other snapshots from this url

All snapshots

from host

commentsarchive.softf1.com

Webpage

Screenshot

share

download .zip

report bug or abuse

donate

/*=====

// comments

archive

=====*/

Main page

Recent changes

Random page

Help

Tools

What links here

Related changes

Special pages

Printable version

Permanent link

Page information

Log in

Page

Discussion

Read

View source

View history

Search commentsarchive

C 0000007 mmmv btreetlike folder structure t1

C 1..100

The idea is that the vertices of a [B-tree](#) are folders, folder names are keys, with the exception of the folder named "root", files have a key as their file name prefix, which is written as a base 10 positive whole number, including zero, followed by an underscore and the rest of the file name. The interpretation of the keys in non-leaf vertices is: "less than or equal to the key". The tree structure differs from a [B-tree](#) by the following differences:

• there is no requirement that the paths from the root to all leaves are equal;

• the minimum number of keys at a non-leaf vertex is 1 independent of the maximum number of allowed keys at a non-leaf vertex;

• the tree does not need to be perfectly balanced;

• the theoretical maximum number of records in the tree is determined by the maximum folder name length and maximum file name length, which are determined by the filesystem.

```
#!/usr/bin/env bash
#=====
# Initial author of this script: Martin.Vahi@softf1.com
# This file is in public domain.
# The following line is a spdx.org license label line:
# SPDX-License-Identifier: 0BSD
#
# This Bash script generates a folder that conforms to the
# mmmv_btreetlike_folder_structure_t1 specification.
# https://commentsarchive.softf1.com/index.php?
title=C_0000007_mmmv_btreetlike_folder_structure_t1
#-----
S_FP_DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
S_FP_ORIG="$( pwd )"

ffile(){
    local S_KEY="$1"
    local S_FILENAME_SUFFIX="_data.txt"
    local S_FN="${S_KEY}${S_FILENAME_SUFFIX}"
    echo "x" > ".$S_FN"
} # ffile

dddir(){
    mkdir ./${1}
    cd "${1}"
} # dddir

enddd(){
    cd ../
} # enddd

#-----
cd "$S_FP_DIR"
rm -fr ./tree_demo
#-----
dddir tree_demo
    dddir root
        ffile 99999
        ffile 501
        dddir 500
            ffile 500
            ffile 499
            ffile 301
            dddir 300
                ffile 300
                ffile 251
            enddd
        enddd
    enddd
```

https://archive.fo/ZkxTw

1/3

```
dddir 250
  ffile 250
  enddd
dddir 210
  ffile 201
  enddd
enddd
dddir 200
  dddir 150
    ffile 150
    ffile 101
    enddd
  enddd
dddir 100
  dddir 80
    ffile 80
    enddd
  dddir 30
    ffile 0
    enddd
  enddd
enddd
enddd
enddd

#-----
# The console output of this script:
# -----citation---start-----
#
# | demo.bash
# | tree_demo
# | root
# |
# | 100
# | | 30
# | | | 0_data.txt
# | | | 80
# | | | | 80_data.txt
# | | 200
# | | | 150
# | | | | 101_data.txt
# | | | | 150_data.txt
# | | 500
# | | | 210
# | | | | 201_data.txt
# | | | 250
# | | | | 250_data.txt
# | | | 300
# | | | | 251_data.txt
# | | | | 300_data.txt
# | | | 301_data.txt
# | | | 499_data.txt
# | | | 500_data.txt
# | | 501_data.txt
# | | 99999_data.txt
#
# 11 directories, 14 files
# -----citation---end-----
#-----

tree -L 9
cd "$S_FP_ORIG"
#=====
```

Implementation Related Aspects

As with all filesystem access cases, there's the issue with filesystem access speed, race conditions, power failures during filesystem operations, etc., but those are all outside of the scope of this specification and should be handled by the implementations the way the developers of the implementations see fit. The mmmv_btreetlike_folder_structure_t1 describes only the end result, which might be a folder structure at some documentation CD/DVD.



