

# Proposal for RTFS - Real-Time FileSystem

Authors: Domenik Irrgang, Tobias Derksen

Date: Venlo, 20th September 2017

## Introduction

This project will be carried out in context of the module IOT-CPP of the Informatics study at Fontys School of Technology Venlo (NL).

The purpose of this project is to meet the learning goals of the module which includes low-level C programming, memory management and real-time computations. All of these topics are included in this project.

## File System

The goal of this project is to write a file system using a balanced-tree structure to quickly find inodes. The filesystem should be usable on the linux operating system. We will use FUSE to register the filesystem to the kernel and to mount a folder into the existing directory structure. The filesystem itself will be written to a single file, containing all file, folders and meta information of this filesystem.

## Structure

The filesystem will be based in inodes and blocks. The block size will be variable and can be determined on creation. Each block gets an inode which identifies the content. To quickly find inodes inside the filesystem, a balanced key structure will be used. The key will be composed out of hashes of the filename and the parent directory structure of the file. The value, associated with the key, will be the starting address of the inode inside the file system. Therefore, single files, as well as all files of one folder can be found very quickly by looking up the inodes addresses in the tree structure.

## Real-Time Component

To meet the real-time computation learning goal, the file system will support a special mount option which enables a memory write cache. This ensures a fast write on the file system while the real write will be done asynchronously at a later point in time. Application which needs a real time performance on this file system can instantly continue working while the file system writes down the data in background.

This is of-course a very dangerous option. If the power supply fails before all data has been written to disks, data will be lost and possibly the file system will be in an inconsistent state.