

Betriebssysteme Übung

Tutorium

Intro

Tino Kallinich -Friedrich

Dennis Schneider

Agenda

Termine & Kontakt

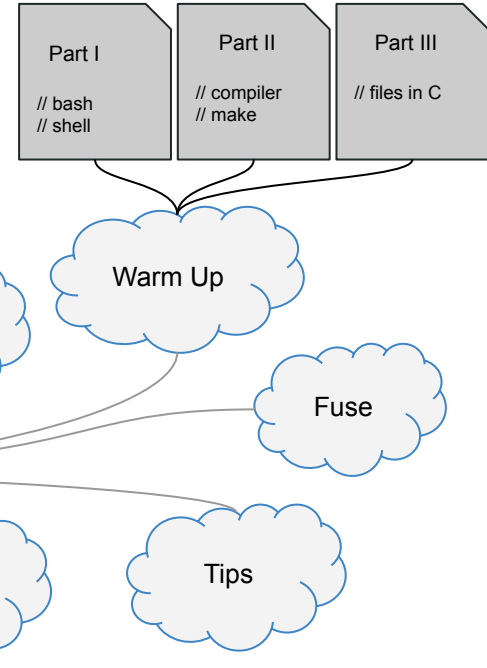
Organisation

Aufgabe

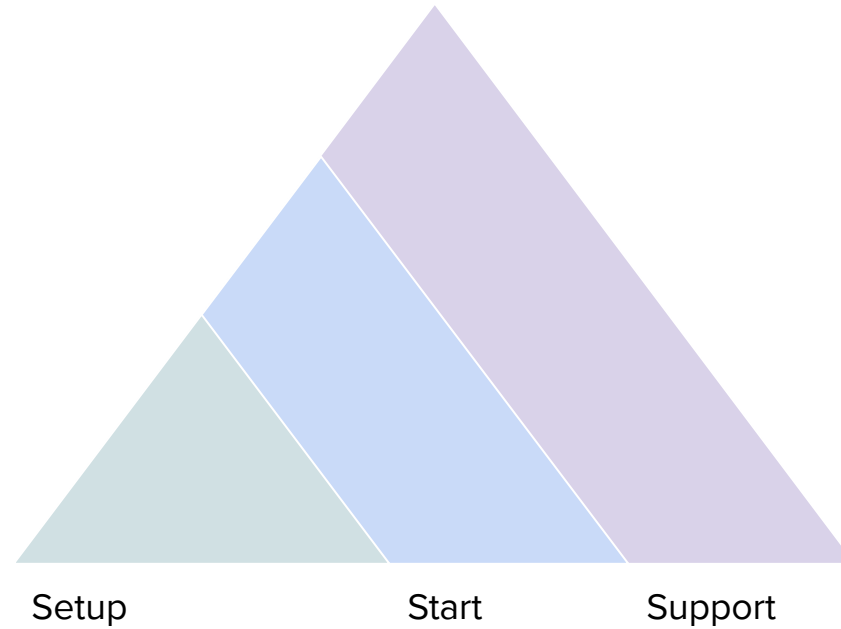
Ausblick | Tools

Get Started

Fragen



Verlauf



Termin

erste 9.10.2019 Mittwoch 17:10 -18:30

folgende Mittwoch 17:10 -18:30

Kontakt

Tino Kallinich-Friedrich

email: kati1017@hs-karlsruhe.de

Dennis Scheider

email: scde1037@hs-karlsruhe.de

Organisation

Wir bieten an ...

Wöchentliches Tutorium (ca. 15 Termine)

Hilfestellung | Tipps | Tricks

Q & A

Feedback?

Aufgabe

In-Memory File System (RAM)

- Beim Starten des Programms mount.myfs wird ein leeres Dateisystem erstellt
- Dateien die in dem Dateisystem abgelegt werden, werden ausschließlich im RAM des Rechners gehalten
- Nach Programmende, - absturz, Neustart des Rechners etc. sind die Daten nicht mehr vorhanden

On-Disk File System

- Das von mount-.myfs verwaltete Dateisystem wird auf einen Datenträger geschrieben
- Nach Programmende und Neustart sind alle Dateien noch vorhanden

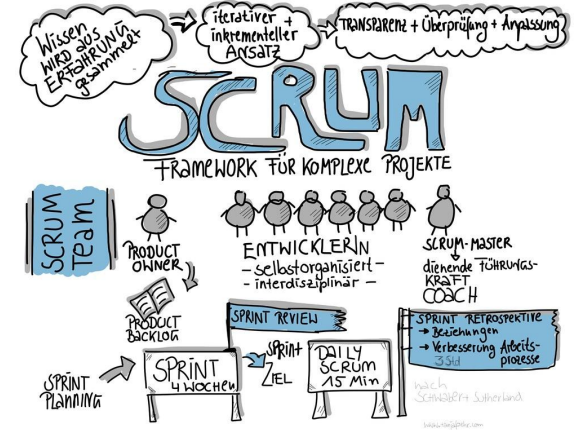
Ausblick

Was euch erwartet.

//Tafel: Projektverlauf, Meilensteine

Wie Ihr Probleme vermeidet.

Konstruktives Zusammenarbeiten



Tools

Team Organisation

<http://opensource.com>

Team Kommunikation

Versionierung

Testing & Debugging

<http://www.valgrind.org>

Betriebssystem

IDE

Get Started

Warm Up

Part I

// bash and shell

Tutorial Link: [bash tutorial](#)

// learn bash commands

Source Link: [script](#)

Part II

// compiler and make files

Source Link: [script](#)

// create, compile and link C++ files from command line

// create and run make files

Part III

// files in C

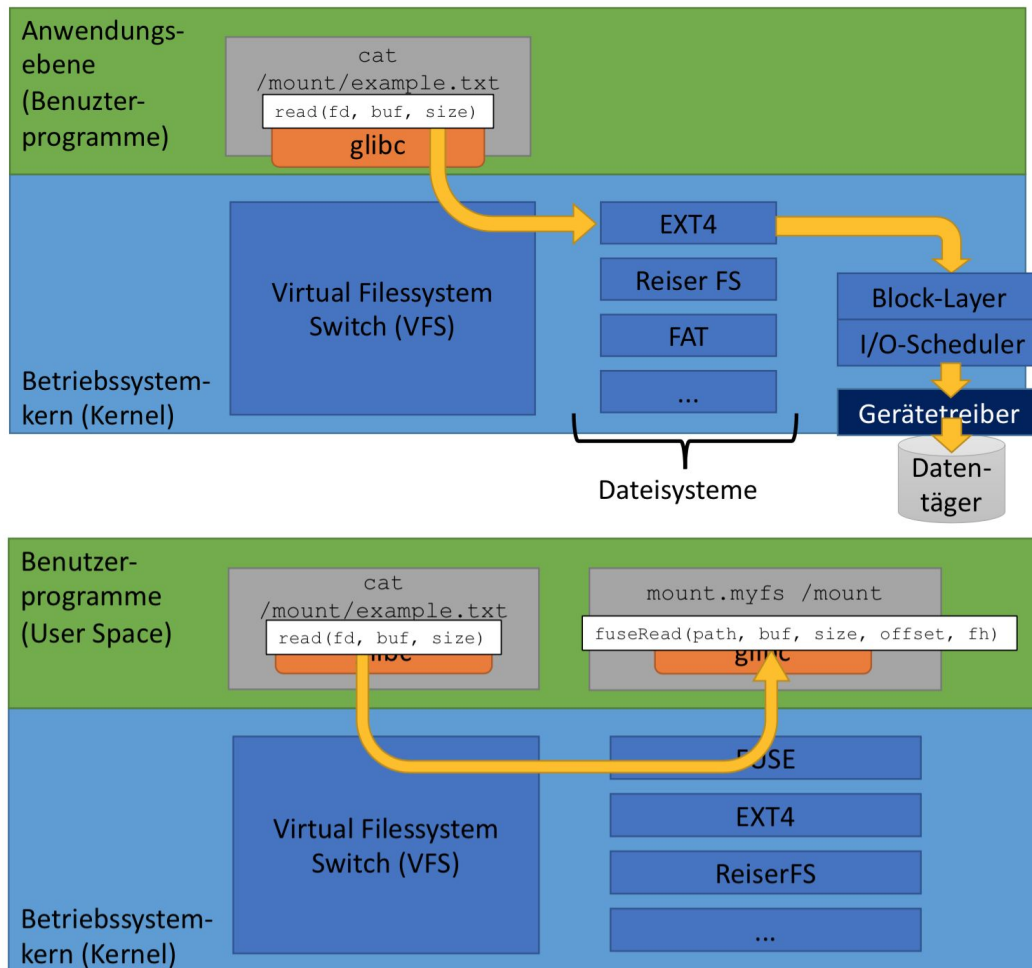
Source Link: [script](#)

// read and write to file from C++

Homework

Get Started Fuse

Link zur Dokumentation: [fuse](#)



Get Started

Projekt Struktur

Homework

MkFs

// Erstellt die Struktur des Dateisystems

MyFs

// Ermöglicht den Zugriff auf das Dateisystem mittels Fuse Methoden.

Unittest

// Deine Unittests.

Make

// Kompiliert und Linked alle Projekt relevanten Klassen. (inkl. unittest)

mount

// mounting point für das Dateisystem [../path]

log.txt

// Echtzeit log für das Dateisystem [../path]

Get Started Links

Hilfreiche Links

- SSFS :
<https://engineering.facile.it/blog/eng/write-filesystem-fuse/>
- C++ Tutorial:
<https://www.youtube.com/watch?v=tvC1WCdV1XU&list=PLAE85DE8440AA6B83>
<http://www.cplusplus.com/doc/tutorial/>
- Test C++ Code with Catch Framework
<https://github.com/catchorg/Catch2/blob/master/docs/tutorial.md#top>
- Betriebssysteme File Allocation Table
<https://www.youtube.com/watch?v=V2Gxqv3bJCk>

Homework

Get Started

next Steps

1. Team Up
2. Get organised
3. Understand the project
4. Commit to the project // what if ...
5. Setup / commit to a timetable // realistic last chance 8 Jan. 2020
6. Setup IDE
7. run / evaluate SSFS
8. Do your Homework

Next Week: // what else?

- open questions (IDE, Project, Organisation)
- Homework Recap
- Fuse
- C++

Fragen

