

Ausgewählte Kapitel der Systemsoftwaretechnik: Rekonfigurierbare Systemsoftware

Christoph Erhardt, Peter Ulbrich

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Sommersemester 2014

https://www4.cs.fau.de/Lehre/SS14/MS_AKSS/



Einführung: Rekonfigurierbare Systemsoftware

Masterseminar AKSS: Rekonfigurierbare Systemsoftware

Organisatorisches

Themen und Einteilung

Seminarmodus

Fachliteratur lesen und verstehen

Vortrag strukturieren, gestalten und vorbereiten





Brücke von Alcántara, Fertigstellung 106 n. Chr.
Bild: https://commons.wikimedia.org/wiki/File:Bridge_Alcantara.JPG

- Einmal bauen, kann 2000 Jahre unverändert stehen bleiben.





Brücke von Alcántara, Fertigstellung 106 n. Chr.
Bild: https://commons.wikimedia.org/wiki/File:Bridge_Alcantara.JPG

- Einmal bauen, kann 2000 Jahre unverändert stehen bleiben.

Klappt sowas auch mit Software?



Alltag im Software-Bereich

- Sicherheitslücken
- Neue Anforderungen, neue Features
- Geänderte Rahmenbedingungen
- ...

⇒ **Software ist nie wirklich „fertig“!**



Alltag im Software-Bereich

- Sicherheitslücken
- Neue Anforderungen, neue Features
- Geänderte Rahmenbedingungen
- ...

⇒ **Software ist nie wirklich „fertig“!**



Klassischer Ansatz

1. Änderungen einspielen
2. Software neu starten

⇒ **Zustandsverlust, Ausfallzeiten, ...**



Alltag im Software-Bereich

- Sicherheitslücken
- Neue Anforderungen, neue Features
- Geänderte Rahmenbedingungen
- ...

⇒ **Software ist nie wirklich „fertig“!**



Klassischer Ansatz

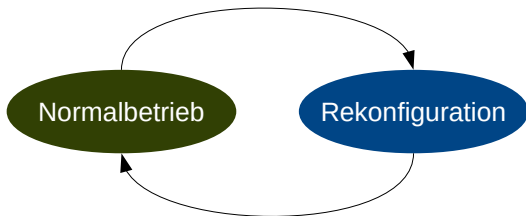
1. Änderungen einspielen
2. Software neu starten

⇒ **Zustandsverlust, Ausfallzeiten, ...**



Geht das nicht besser?





- Rekonfiguration im laufenden Betrieb
- Kein Zustandsverlust
- Deutlich reduzierte Verzögerung



Was wird rekonfiguriert und warum?

Betriebssystem-/Anwendungscode

- **Ziel:** Stopfen von Sicherheitslecks, Einspielen neuer Versionen



Was wird rekonfiguriert und warum?

Betriebssystem-/Anwendungscode

- **Ziel:** Stopfen von Sicherheitslecks, Einspielen neuer Versionen

Systemkomponenten in eingebetteten Systemen

- **Ziel:** Erweiterbarkeit, Flexibilität



Was wird rekonfiguriert und warum?

Betriebssystem-/Anwendungscode

- **Ziel:** Stopfen von Sicherheitslecks, Einspielen neuer Versionen

Systemkomponenten in eingebetteten Systemen

- **Ziel:** Erweiterbarkeit, Flexibilität

Ablaufplanung in Echtzeitsystemen

- **Ziel:** garantierte Dienstgüte von Regelanwendungen, optimierte Ressourcenausnutzung



Plugins, Erweiterungen

- Kernel-Module
- Dynamische Bibliotheken
- Nachladbare Klassen (z. B. Java)



~~Plugins, Erweiterungen~~

- ~~■ Kernel-Module~~
- ~~■ Dynamische Bibliotheken~~
- ~~■ Nachladbare Klassen (z. B. Java)~~

Dynamic Software Update

1. Konstruktiv durch geeignete Programmiersprache (OOP, AOP)
2. Instrumentierung von vorhandenem Code
3. Externe Werkzeuge – ohne Änderungen an der Codebasis



~~Plugins, Erweiterungen~~

- ~~■ Kernel-Module~~
- ~~■ Dynamische Bibliotheken~~
- ~~■ Nachladbare Klassen (z. B. Java)~~

Dynamic Software Update

1. Konstruktiv durch geeignete Programmiersprache (OOP, AOP)
2. Instrumentierung von vorhandenem Code
3. Externe Werkzeuge – ohne Änderungen an der Codebasis

Zustandstransfer?!



- Regelfall: System komplett statisch konfiguriert
 - Ausnutzen von statischem Anwendungswissen
 - Optimierung bezüglich Geschwindigkeit und Größe
- Fragestellung: Wie Software-Änderungen „im Feld“ einspielen?
 - ... ohne den laufenden Betrieb zu stören
 - ... ohne das System ineffizient zu machen



Echtzeitsysteme

- Ablaufplan meist statisch vorgegeben (*Offline-Scheduling*)
- Planungskriterium: Einhaltung aller Fristen
- Ressourcenauslastung kein primäres Entwurfsziel
- Garantierte Dienstgüte

Herausforderung: Ablaufplanung dynamisch anpassbar machen



Einführung: Rekonfigurierbare Systemsoftware

Masterseminar AKSS: Rekonfigurierbare Systemsoftware

Organisatorisches

Themen und Einteilung

Seminarmodus

Fachliteratur lesen und verstehen

Vortrag strukturieren, gestalten und vorbereiten



■ Verantwortliche

- | | | |
|---------------------|----------------|-------------------|
| ■ Christoph Erhardt | Raum 0.042-113 | erhardt@cs.fau.de |
| ■ Peter Ulbrich | Raum 0.037-113 | ulbrich@cs.fau.de |

■ Termin

- Freitag, 12:15–13:45
- Raum 01.151-128

■ Web-Seiten

- Lehrveranstaltung:
https://www4.cs.fau.de/Lehre/SS14/MS_AKSS/
- Anmeldung (Waffel):
<https://waffel.cs.fau.de/signup/?course=205>

■ Rückmeldungen und Fragen

- Bitte Fragen stellen!
- Auf Fehler aufmerksam machen



- Thema 1: Dynamische Updates für objektorientierte Betriebssystemkerne
- Thema 2: Dynamische Software-Aktualisierung für C-Anwendungen
- Thema 3: Sicherheits-Patches für den Linux-Kern ohne Neustart
- Thema 4: Zustandstransfer bei dynamischer Aktualisierung
- Thema 5: Dynamische aspektorientierte Programmierung
- Thema 6: Rekonfiguration für eingebettete Geräte
- Thema 7: Anwendungsgewahre Ressourcenverwaltung in Echtzeitregelungssystemen
- Thema 8: Rekonfiguration in Echtzeitsystemen

Hinweis:

Weiterführende Literatur zu den einzelnen Themen siehe Web-Seite



- Schriftliche Ausarbeitung
 - Umfang mindestens 6 Seiten
 - ACM-Stil (zweispaltig, 9-Punkt), siehe:
<http://acm.org/sigs/publications/proceedings-templates>
- **Abgabe der Ausarbeitung:**
 - Erste Fassung: spätestens zwei Wochen vor dem Vortrag
 - **Vortragsfassung: spätestens eine Woche vor dem Vortrag**
 - Finale Fassung: spätestens eine Woche nach dem Vortrag

Hintergrund: Einarbeitung der Resonanz aus dem Vortrag
- Weitere Hinweise
 - Ausarbeitung unter Zuhilfenahme der Wissensbasis erstellen:
https://www4.cs.fau.de/Lehre/SS14/MS_AKSS/wissensbasis.pdf
 - Ausarbeitung entweder auf Deutsch oder Englisch



- Vortrag
 - 40-Minuten-Vortrag plus anschließende Diskussion
 - Zur Vorbereitung *mindestens* einmal zur Probe halten
 - **Abgabe der Vortragsfolien:**
 - Erste Fassung: spätestens eine Woche vor dem Vortrag
 - Finale Fassung: spätestens einen Tag vor dem Vortrag
 - Grundlagen der Wissensbasis auch für den Vortrag anwenden

- Aktive Teilnahme
 - Vorbereitung anhand der Vortragsfassung des jeweiligen Vortrags
 - Anwesenheit
 - Beteiligung an den Diskussionen

- Vortragsevaluation
 - Die Seminarteilnehmer bewerten gegenseitig ihre Vorträge
 - Evaluationsbögen werden zu den Seminarterminen bereitgestellt



Einführung: Rekonfigurierbare Systemsoftware

Masterseminar AKSS: Rekonfigurierbare Systemsoftware

Organisatorisches


Themen und Einteilung

Seminarmodus

Fachliteratur lesen und verstehen

Vortrag strukturieren, gestalten und vorbereiten



- Gründe, ein Papier zu lesen
 - Literaturanalyse relevanter verwandter Arbeiten
 - Begutachtung von zur Veröffentlichung eingereichten Beiträgen
 - [Weil es für das Masterseminar notwendig ist]
 - ...
- Mögliche Herangehensweise: Mindestens drei Lesedurchgänge mit jeweils unterschiedlichem Fokus
 - 1. Durchgang: Erster allgemeiner Eindruck
 - 2. Durchgang: Überblick über den Inhalt
 - 3. Durchgang: Detailliertes Verständnis
- Literatur
 -  Srinivasan Keshav
How to Read a Paper
ACM SIGCOMM Computer Communication Review, 37(3):83–84, 2007.



1. Lesedurchgang

- Ziel: Verschaffen eines ersten allgemeinen Eindrucks
- Interessante Fragestellungen
 - In welche Kategorie (z. B. Analyse eines bereits existierenden Systems, Beschreibung eines Prototyps, etc.) fällt das Papier?
 - Was ist der wissenschaftliche Beitrag des Papiers?
 - Sind die getroffenen Annahmen dem ersten Anschein nach berechtigt?
 - Mit welchen anderen Papieren ist das Papier thematisch verwandt?
- Vorgehensweise
 - Detailliertes Lesen
 - Titel
 - Abstract
 - Einleitung
 - Schluss
 - Kurzer Blick auf
 - Überschriften
 - Referenzen



2. Lesedurchgang

- Ziel: Verschaffen eines Überblicks über den Inhalt
- Interessante Fragestellungen
 - Was ist der (komplette) Inhalt des Papiers?
 - Wie würde ich einem Anderen den Inhalt des Papiers erklären?
 - Enthält das Papier offensichtliche Fehler?
- Vorgehensweise
 - Detailliertes Lesen bzw. Betrachten
 - Abschnitte aus 1. Lesedurchgang
 - Restliche Abschnitte
 - Abbildungen, Graphen, etc.
 - Aussparen von Details (z. B. Beweisen)
 - Notizen
 - Zentrale Punkte
 - Relevante Referenzen
 - Unklare Stellen



3. Lesedurchgang, Anfertigung der Ausarbeitung

- Ziel: Detailliertes Verständnis des Papiers
- Interessante Fragestellungen
 - Was sind die wesentliche Beiträge des Papiers?
 - Sind die auf Basis der Annahmen gezogenen Schlüsse korrekt?
 - Werden Annahmen getroffen, die nicht explizit erwähnt sind?
- Vorgehensweise
 - Besonderes Augenmerk auf Details
 - (Gedankliches) Nachvollziehen der präsentierten Experimente
 - Heranziehen von referenzierten verwandten Arbeiten
- Vertiefung, Anfertigung der Ausarbeitung
 - Die wichtigsten verwandten Arbeiten im gleichen Modus bearbeiten
 - Ausarbeitung unter Zuhilfenahme der Wissensbasis erstellen:
https://www4.cs.fau.de/Lehre/SS14/MS_AKSS/wissensbasis.pdf
 - Abgabetermine beachten



Einführung: Rekonfigurierbare Systemsoftware

Masterseminar AKSS: Rekonfigurierbare Systemsoftware

Organisatorisches

Themen und Einteilung

Seminarmodus

Fachliteratur lesen und verstehen

Vortrag strukturieren, gestalten und vorbereiten

