

Verlässliche Echtzeitsysteme

Übungen zur Vorlesung

Abstrakte Interpretation & Verifikation: Übungsaufgabe

Phillip Raffeck, Florian Schmaus, Simon Schuster

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)
<https://www4.cs.fau.de>

Sommersemester 2020



Gesamtsystem

- Idee: Vorverarbeitung (Filterung) von optischen Sensordaten
 - Bilder werden von Sensor in Ringpuffer abgelegt
 - Filterung der rohen Bilddaten
- Keine Implementierung notwendig

Aufgabenstellung

- Sensorstubs beschreiben
- Abwesenheit von Laufzeitfehlern im Filter nachweisen
- Bestimmung des Optimalwertes für Filterparameter
 - ↳ Astrée
- Funktionale Korrektheit des Ringpuffers nachweisen
 - ↳ Frama-C



Gaußsche Weichzeichnung

- Hinzufügen von Unschärfe
- Verminderung von Bildrauschen
- ~> Verbesserte Kantenerkennung



Original



Weichzeichnung

- Mathematisch: Faltung mittels Gauß-Funktion
- Matrix befüllt nach Gaußscher Verteilung ausgehend vom Zentrum
- Übungsziel: Normalisierungsfaktor bestimmen





Original



intervallbasierte
Normalisierung



korrekte
Normalisierung



Original



intervallbasierte
Normalisierung



korrekte
Normalisierung

Original Photo by pedrojperéz at Mouragefile.com



■ Beispiel für dynamisch angelegten Ringpuffer

```
1 struct BndBuf{
2     int* buf; // array of <size>
3     int size;
4     int read_pos;
5     int write_pos;
6 };
```

■ „Füllstandsanzeige“: Speichern von

- Belegten Plätzen
- Verfügbaren Plätzen

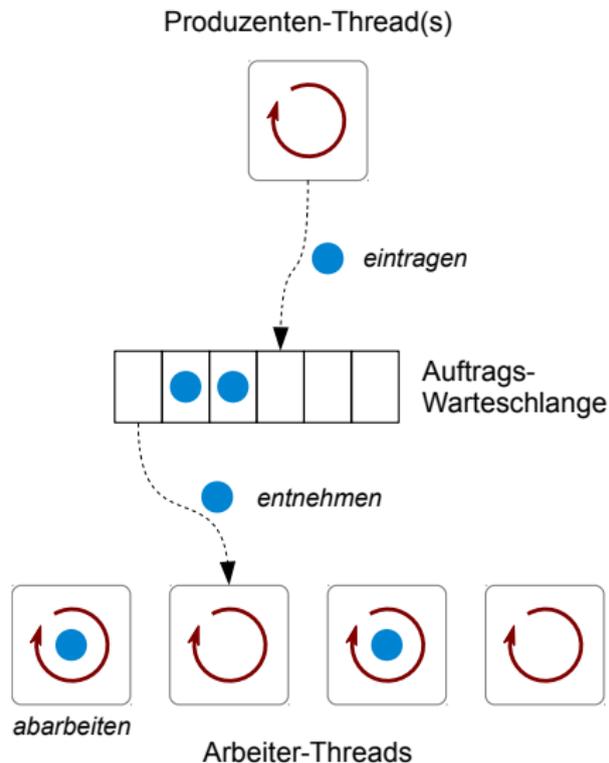
■ Mögliche Signalisierung falls

- Plätze frei wurden
- Plätze belegt wurde



Ringpuffer – Anwendungsszenario

- Verwaltung einer begrenzten Anzahl an Ressourcen
- Beispiel: Thread-Pool
- Feste Menge von Arbeiter-Threads:
 - laufen endlos
 - erhalten Aufträge zur Abarbeitung
- Verteilen der Aufträge mittels zentraler, synchronisierter Warteschlange (z. B. Ringpuffer)
- Vorteil: kein ständiges Erzeugen + Zerstören von Threads für Aufträge



Grundlegende Funktionalität

- Erfüllt jede einzelne Funktion ihre Aufgabe?

↪ Verhaltensbeschreibung jeder Funktion notwendig

Interessante Eigenschaften

- Verhalten sich die Read/Write-Positionszeiger wie erwartet?
- Werden Elemente auch wirklich eingefügt?
- Entspricht die Entnahmereihenfolge der Erwartung?

↪ Axiomatisierung mittels zusätzlicher Prädikate

↪ Anwendungsfunktionen als Beweishilfe

