



D&A - Übungsstunde 14

Diese Folien basieren auf denjenigen der Vorlesung, wurden aber durch den Assistenten Adel Gavranović adaptiert und erweitert

Heutiges Programm

Intro

Follow-up

Parallel Mergesort

Dining Philosophers

Informationen zur Prüfung

Prüfung

Outro



`n.ethz.ch/~agavranovic`

▶ [Link zum Material für die Übungsstunden](#)

▶ [Webseite des Assistenten](#)

▶ [Mail an Assistenten](#)

1. Intro

Intro

- Letzte Übungsstunde!

2. Follow-up

(Einige!) Fehler aus letzter Übungsstunde

Slide "nichtdeterministische Ausführung"

- Die Indizes sollten eigentlich in $[1, 3]$ sein

Slide "Counter Lösung 1"

- `int counter {0}` hätte auch einfach = 0 sein können.

int counter {0}

int cnt = 0

Slide "Quiz"

- AA war **falsch!**

3. Parallel Mergesort

Aufgabe: Mergesort (2-threads)

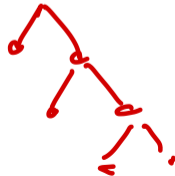
```
void mergesort_par( std::vector<int> & v ) {  
    int n = v.size();  
    int partSize = n / 2;  
  
    std::thread t1( mergesort, std::ref(v), 0, partSize-1 );  
    std::thread t2( mergesort, std::ref(v), partSize, n-1 );  
    t1.join();  
    t2.join();  
    merge( v, 0, partSize-1, n-1 );  
}
```



Analog mit n threads

Aufgabe: Mergesort Rekursiv

```
void mergesort_par(std::vector<int> & v, int cutoff, int l, int r) {  
    if (r-l < cutoff){ // sequential base case  
        mergesort( v, l, r );  
    } else {  
        int m = ( l+r )/2 ;  
        std::thread t (mergesort_par,std::ref(v),cutoff,l,m);  
        mergesort_par(v,cutoff,m+1,r); // avoid forking another thread  
        t.join();  
        merge(v,l,m,r);  
    }  
}
```

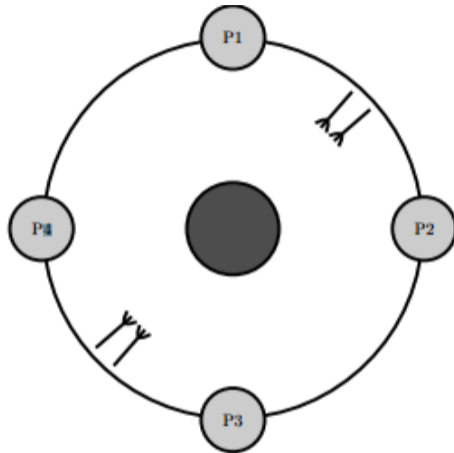


4. Dining Philosophers

Dining Philosophers

- Um Deadlock zu verhindern, zirkuläre Abhängigkeit brechen. Wie letztes Mal diskutiert.
- Max/Min Anzahl Philosophen die gleichzeitig essen?
- Es ist möglich dass nur ein Philosoph isst.

Gabeln bündeln! Dann können immer zwei essen.



5. Informationen zur Prüfung

Relevantes für die Prüfung

Prüfungsstoff für die Endprüfung schliesst ein

- Vorlesungsinhalt (Vorlesung, Handout) und
- Übungsinhalte (Übungsstunden, Übungsblätter).

Relevantes für die Prüfung

Prüfung (150 min) ist schriftlich. **Hilfsmittel: vier A4-Blätter ohne** inhaltliche und formale Anforderungen (Text, Bilder, ein-/doppelseitig, Ränder, Schriftgrößen)

Die Prüfung findet am Computer statt (Moodle und CodeExpert).

Alte Prüfungen

Prüfungssammlung

zuerst lösen, dann Lösungen anschauen!

Struktur

- etwa 4 Theorie-Aufgaben (ca. 70 Punkte):
 - Kurzaufgaben
 - Asymptotik und Rekursionsgleichungen
 - 2 grössere Aufgaben
- etwa 3 CodeExpert-Aufgaben (ca. 50 Punkte)



6. Prüfung

Let's have a look!

7. Outro

Allgemeine Fragen?

Bis zum nächsten Mal

Schönes Wochenende!