



# Basispraktikum: Systemarchitektur 1

---

Einführung  
SS 2009

23. April 2009  
Frank Bellosa, Gerd Liefländer,  
Andreas Merkel



# Übersicht

- Organisation
- Literatur
- Versuchsübersicht
- Teambildung
- Arbeitsumgebung
- Einführung in BS + Linux



# Basispraktikum

## Forum für:

- Erfahrung mit systemnaher Programmierung
- Programmieren von parallelen Programmen
  - Concurrency & Coordination
- Parallelitätsprobleme
  - Race Conditions (Wettlaufsituationen)
  - Exklusive Ressourcennutzung
  - Umgang mit gemeinsamen Ressourcen
  - Ausnahmen (exceptions)



## Unsere Erwartungshaltung:

- Interesse am Stoff
  - Hohe Eigeninitiative
  - Anwesenheit bei **jedem** Besprechungstermin
- ⇒ Anwesenheitsliste

⇒ Donnerstag: **17:30 – 19:00 Uhr**



# Besprechungstermine WS 2008/09

Do. 23. 4.

Do. 30. 4.

Do. 7. 5.

Do. 14. 5.

Do. 21. 5.

Do. 28. 5.

Do. 4. 6.

Do. 11. 6.

Do. 18. 6.

Do. 25. 6.

Do. 2. 7.

Do. 9. 7.

Do. 16. 7.

Do. 23. 7.



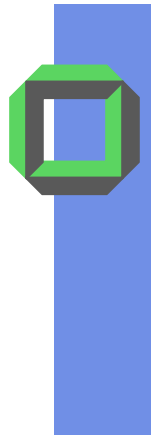
# Versuchsdurchführung

- Rechtzeitig anfangen
- Rechtzeitig anfangen
- **Rechtzeitig anfangen**
  - Aufgabenstellung im Team verstehen
  - Offene Fragen mit eurem Tutor abklären
  - Gründliches Konzept erarbeiten
  - Wechselseitiges Implementieren & Testen



# Versuchsdurchführung

- **Versuchsprotokoll** anfertigen!!!
  - Fragen beantworten und Prognosen abgeben
  - Design- und Implementierungsfehler festhalten
  - In der 2. Woche implementieren + testen
- **Fristgerechtes** Vorführen der Experimente nach Terminvereinbarung mit Tutor(in)
- **Fristgerechtes** Vorführen
- **Fristgerechtes** Vorführen



# Organisatoren

## Tutoren:

Dominik Bruhn  
[dominik@dbruhn.de](mailto:dominik@dbruhn.de)

Jonathan Dimond  
[befressor@gmx.de](mailto:befressor@gmx.de)

Johannes Weiß  
[weiss@tux4u.de](mailto:weiss@tux4u.de)

## Veranstalter:

Gerd Liefländer  
[lief@ira.uka.de](mailto:lief@ira.uka.de)  
Tel.: 608-3837

Frank Bellosa  
[bellosa@ira.uka.de](mailto:bellosa@ira.uka.de)

Andreas Merkel  
[amerkel@ira.uka.de](mailto:amerkel@ira.uka.de)





# Literatur

- Bacon, J.: Operating Systems, Addison-Wesley, 2004
- Silberschatz, A.: Operating Systems Concepts, 2005
- Tanenbaum, A.: Modern Operating Systems, Prentice Hall, 2002
- Bovet: Understanding the Linux Kernel
- Beck: Linux: Kernel Programming
- [www](#): Some "Online Java Tutorials"



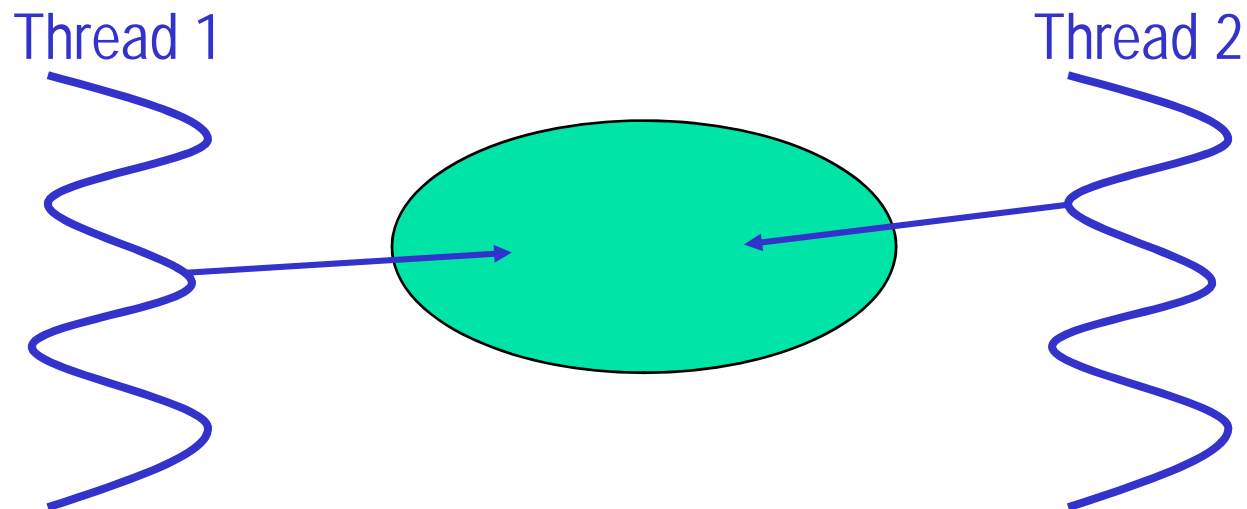
# Versuche

- Versuch 0: Thread-Erzeugung
- Versuch 1a: Kooperation
- Versuch 1b: Semaphore
- Versuch 1c: Erzeuger/Verbraucher
- Versuch 1d: Leser-/Schreiber
- Versuch 1e: Kommunikation



# Kooperation

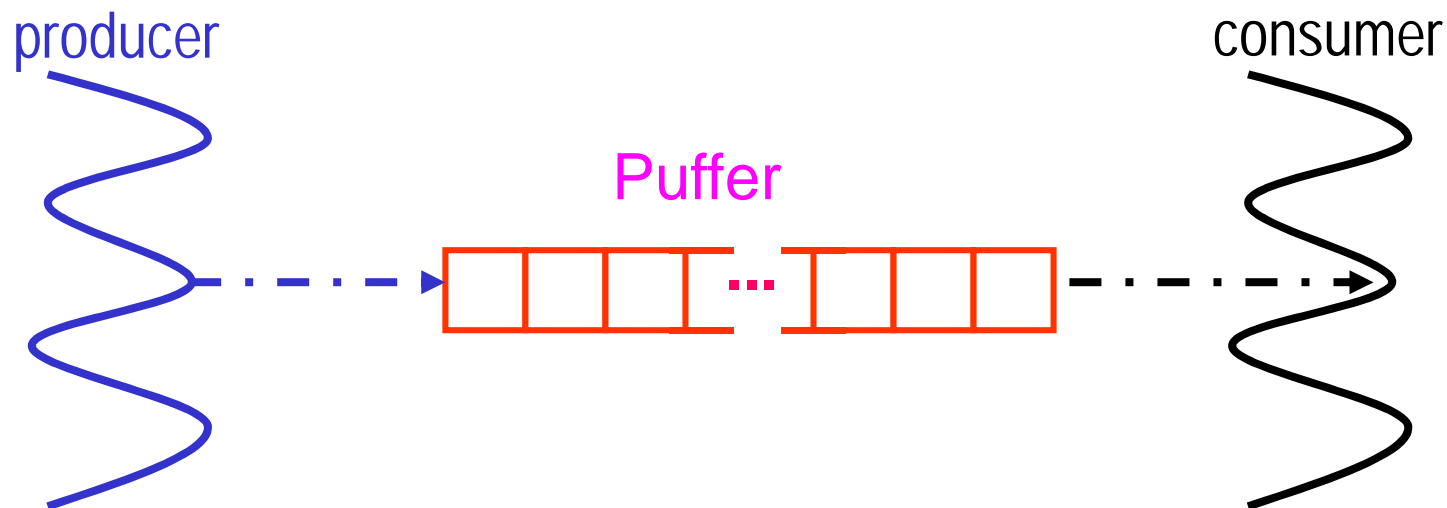
Lösung des Inkonsistenzproblems beim Zugriff auf gemeinsame Daten durch 2 Threads mittels des Monitorkonzepts von Java.



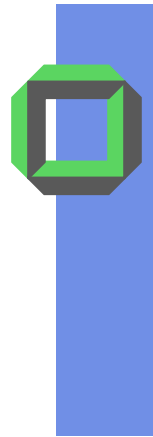


# Erzeuger-Verbraucherproblem

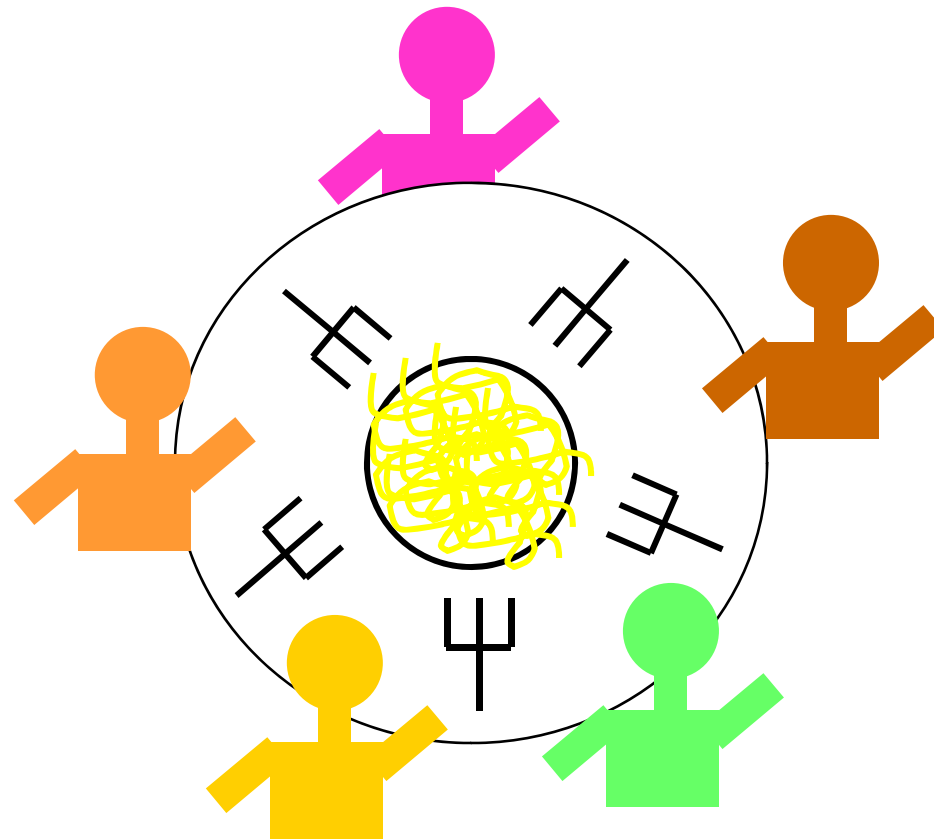
Erzeuger legt ein Produkt im Puffer ab, das vom Verbraucher abgeholt werden kann.



*Kann es da jemals Parallelitätsprobleme geben?*



# Philosophenproblem





# Betriebsmittelprobleme

- Schottenrestaurant
- Santa Claus
- Rohrpost
- Bibliothek
- Affenfelsen

# „Monkeys on the Rocks“

The **northern** and **southern** monkeys have to eat and drink, but most of the time they like to play.

∃ a small rope between the two rocks, but ∃ also hungry crocodiles below the rope.

Rope can carry up to  $m \geq 1$  monkeys, concurrent crossing in both direction is **not possible**.





# Simulationen

- Fahrstuhl
- Durlacher Tor
- Kreuzungsbahnhof





## *Wer arbeitet mit wem im Team?*

- Size of teams:  $n = 2$
- Teams ssgr01 ...
- Login: ssgr...
- see www-page

<http://i30www1.ibds.uka.de/teaching/courses/practicalwork.php?courseid=181&lid=en>



# Versuchsrechner

- Implementierungssprache: **Java**
- Entwicklungstool: **Eclipse**
- Vorführung der Versuchsergebnisse für Versuch 0 auf Linux-Rechnern, z.Z. 12 im Rechnerlabor 149 **oder auf eigenem Rechner**
- Spätere Versuche dann im Rechnerlabor



# Besprechungstermine im HS-102

- Vorbereitung der nächsten Versuche
  - Siehe Systemstruktur, Tasks, Prozesse, Threads (im Anschluss)
  - Scheduling ...
- Nachbereitung der vergangenen Versuche
  - Ausgewählte Teams tragen ihre Resultate vor
- Diskussion sonstiger versuchsnaher Themen
- *Fragen?*



# Versuch 0: Thread Erzeugung

- Erzeugen
- Abwickeln
- Identifizieren
- Beenden
- Parametrisierung der Java-Applikation
- Einfache Visualisierung von Threadabwicklungen

ABGABE Versuch 0:

**8. Mai 2009, 13:00 Uhr**  
(inklusive Versuchsprotokoll)