

# Vorlesung Betriebssysteme

Prof. Dr. Jan Dünneberger

Verteilte Systeme und Betriebssysteme  
Fakultät für Informatik und Mathematik  
Ostbayerische Technische Hochschule Regensburg

Wintersemester 2013/14

# Semaphore und Deadlocks

- Wie in der Vorlesung besprochen, stellt ein *Semaphor* die Operationen **P** ("proberen" nl. test/anfordern) und **V** ("verhogen" nl. increment/freigeben) für die Verwaltung abzählbar beschränkter Ressourcen unter mehreren nebenläufigen Prozessen bereit
- Im Folgenden seien die "Ressourcen" Wasserkocher, Kaffeefilter und Teefilter zu jedem Zeitpunkt nur für *einen* Prozess exklusiv verfügbar:

## Sperranforderungen beim "Kochen" und "Reinigen"

Kaffeekochen:

```
Wasserkocher.P();  
Kaffeefilter.P();  
  Kaffee kochen  
Kaffeefilter.V();  
Wasserkocher.V()
```

Reinigen:

```
Kaffeefilter.P();  
Teefilter.P();  
  Filter reinigen  
Teefilter.V();  
Kaffeefilter.V();
```

Teekochen:

```
Teefilter.P();  
Wasserkocher.P();  
  Tee kochen  
Wasserkocher.V()  
Teefilter.V();
```

- Wann wird eine Verklemmung auftreten?
- Wie kann man Verklemmungen im obigen Ablauf verhindern?

# Das Philosophenproblem

- Sehen Sie sich den Quellcode der Java-Implementierung des bekannten "Philosophenproblems" aus dem Buch "Concurrency" an

## Ausführen des "PhilCanvas" im *AppletRunner*

[http://www.doc.ic.ac.uk/~jnm/book/book\\_applets/concurrency/diners/Philosopher.java](http://www.doc.ic.ac.uk/~jnm/book/book_applets/concurrency/diners/Philosopher.java)

[http://www.doc.ic.ac.uk/~jnm/book/book\\_applets/concurrency/diners/Fork.java](http://www.doc.ic.ac.uk/~jnm/book/book_applets/concurrency/diners/Fork.java)

[http://www.doc.ic.ac.uk/~jnm/book/book\\_applets/concurrency/diners/PhilCanvas](http://www.doc.ic.ac.uk/~jnm/book/book_applets/concurrency/diners/PhilCanvas)

[http://www.doc.ic.ac.uk/~jnm/book/book\\_applets/concurrency/diners/Diners.java](http://www.doc.ic.ac.uk/~jnm/book/book_applets/concurrency/diners/Diners.java)

- Die zugehörigen GIFs finden Sie hier:  
[http://www.doc.ic.ac.uk/~jnm/book/book\\_applets/image](http://www.doc.ic.ac.uk/~jnm/book/book_applets/image)
- Experimentieren Sie mit den in der letzten Vorlesung vorgestellten Lösungsmöglichkeiten (asymmetrisch, randomisiert usw.)
- Wie könnte eine Lösung des *Readers & Writers*-Problems in UNIX (aus der letzten Übung) mit Semaphoren aussehen?