Network Resource Allocation

Vortrag von

- Helge Rabsch

Aus dem Buch

- N. Lynch „Distributed Algorithms“, 1996

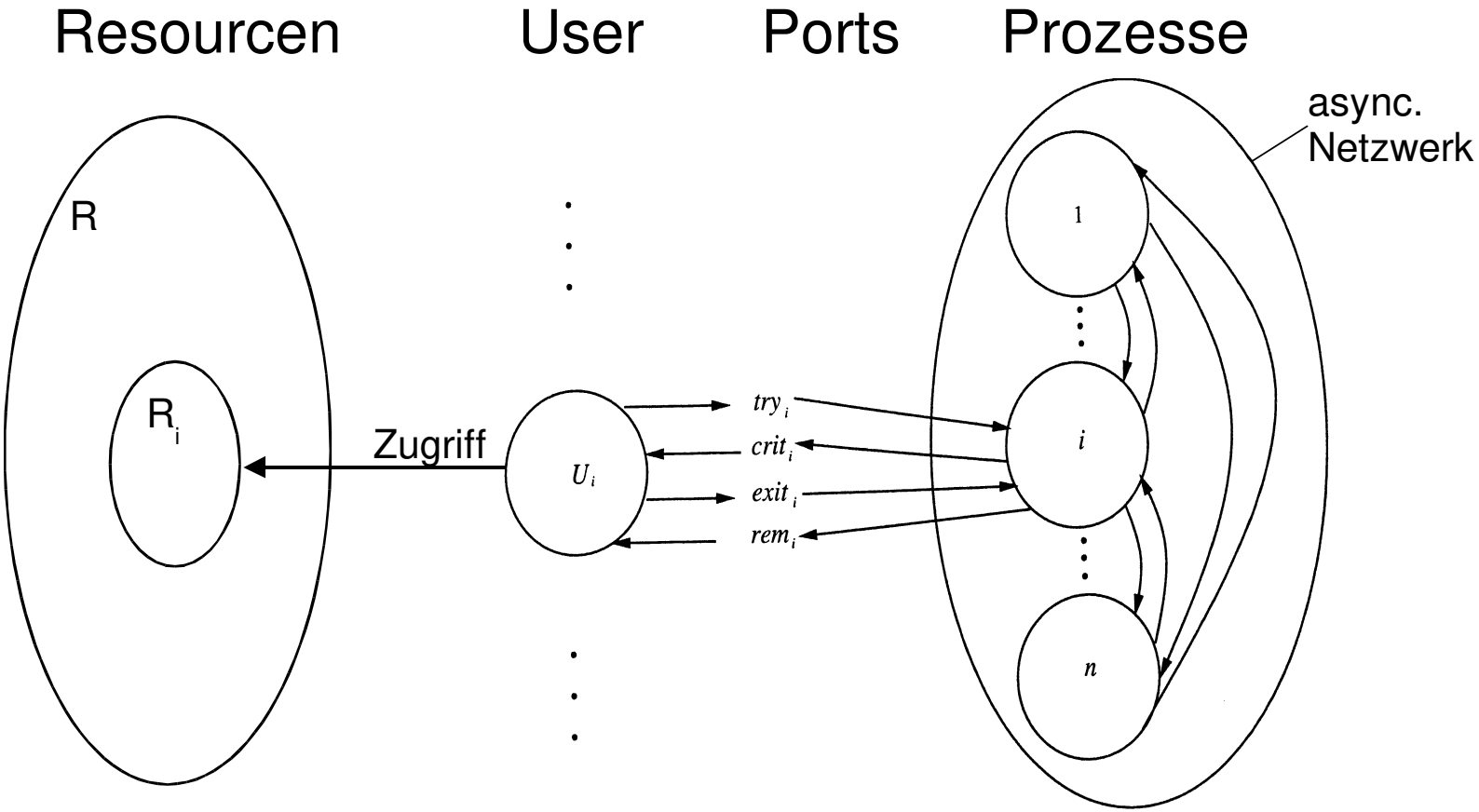
Veranstaltung

- Seminar „Verteilte Algorithmen“

Übersicht

- Einleitung
- Mutual Exclusion
- General Resource Allocation
- Zusammenfassung

Einleitung (I)
Interaktion zwischen Komponenten



Einleitung (II)

Resource Specification

Resource Specification $\mathfrak{R}$ für n User:

- endliche Menge R von Resourcen
- $\forall$ i, $1 \leq i \leq n$: $R_i \subseteq R$: Resourcen, die User i benötigt
- Beispiel: $R=\{r_1,r_2,r_3\}$, $R_1=\{r_1,r_2\}$, $R_2=\{r_2,r_3\}$
- U_i und U_j *konkurrieren*, wenn $R_i \cap R_j \neq \emptyset$

Einleitung (III)

Exclusion Specification

- Menge E: User, die miteinander konkurrieren
- Exclusion Specification ε: alle Mengen E
- Beispiele
  - $\varepsilon = \{ E \subseteq \{ 1,..,n \} : |E| > 1 \}$ Mutual Exclusion
  - $\varepsilon = \{ E \subseteq \{ i,i+1 \} : 1 \leq i \leq n \}$ Dining Philosophers

Einleitung (IV)
Mutual Exclusion

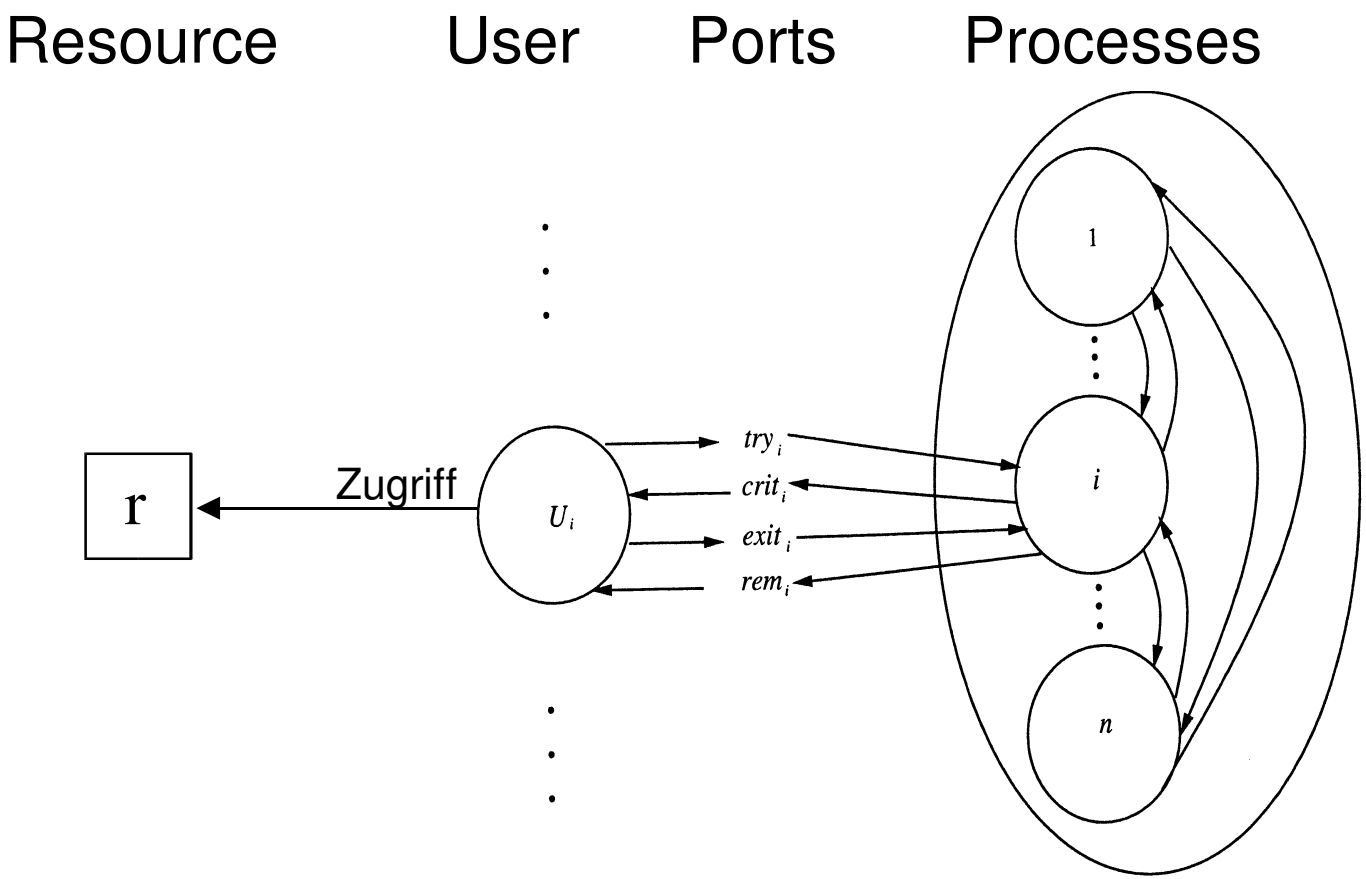
- mehrere User, *eine* Resource
- ein User zur Zeit im *kritischen Bereich* $\mathbf{C}$
- 3 Algorithmen:
  - CirculatingToken
  - LogicalTimeME
  - RicartAgrawalaME

Einleitung (V)

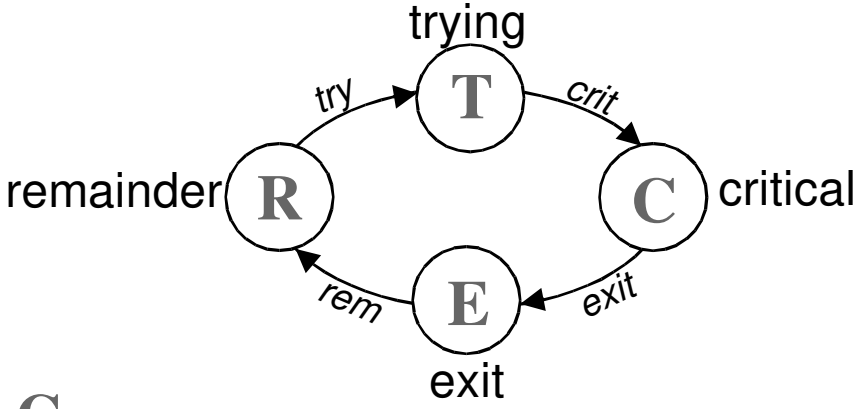
General Resource Allocation

- mehrere User, *mehrere* Resourcen
- nicht-konkurrierende User gleichzeitig in C
- Beispiel: *Dining Philosophers*
- 3 Algorithmen:
  - Coloring Algorithmus
  - RicartAgrawalaRA
  - AcyclicDigraphRA
- *Drinking Philosophers*

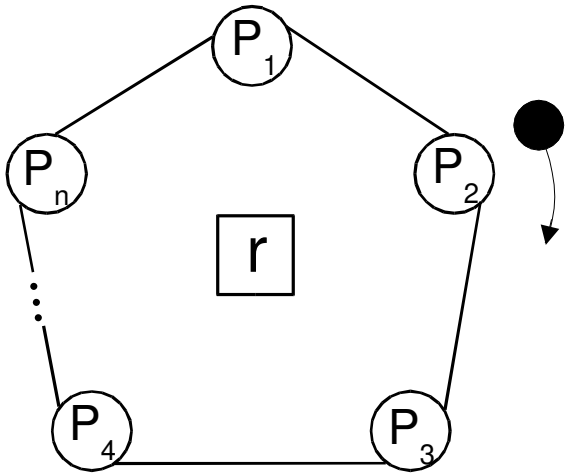
Mutual Exclusion



Mutual Exclusion
Korrektheitsbedingungen



- Well-Formedness
- Mutual Exclusion
  - nie mehr als ein User in **C**
- Progress
  - für **T** und für **E**
- Lockout-Freedom (⇒Progress)
  - für **T** und für **E**

Mutual Exclusion
CirculatingToken (I)

- Token umkreist Ring
- empfängt ein Prozess Token:
  - ggF. Zugriff auf Resource gewähren
  - Token weiterschicken