

Errata

- S. IX: 3. Zeile von unten: Für etliche ...
- S. 7: letzte Zeile: $a(0, n_2, \dots, n_p = \dots$
- S. 16: 22. Zeile: „n++“ statt „Counter++“
- S. 24: 13. Zeile von unten: ... Funktion „pthread-“
- S. 38: 13. Zeile: <https://maurer-berlin.eu/nspbuch/4>
- S. 47: 9. Zeile von unten: ! Full() ist die Vor. für einen Aufruf von Ins(a) .
- S. 49: 4. Zeile: Das vorletzte Wort „wenn“ ist zu streichen.
- S. 61: nach der 7. Zeile ist eine mit } einzufügen
- S. 77: Tab. 3.3: Die 5. und die letzte Zeile ist zu streichen.
- S. 77: 12. Zeile: inWaitingRoom
- S. 78: Der letzte Satz ist wie folgt zu ändern: ... gibt es nicht, weil wir für die Schreibzugriffe die atomare Anweisung Store benutzen.
- S. 80: 23. und 24. Zeile: ... sind dabei die Variablen b, c [N]uint...
- S. 82: 16. Zeile: `func Lock (p uint) {`
- S. 99: 13. Zeile von unten: 10.1145/363162.363167
- S. 103: 14. Zeile von unten: ... „zur“ Effektfreiheit ...
- S. 116: 1. Zeile von unten: ... auf einen Stuhl im Wartezimmer
- S. 117: 6. Zeile: ersetzt wird und der gegenseitige Ausschluss ...
- S. 121: 4. Zeile in Tab. 4.2: Bemerkung: 2 blockiert
- S. 122: 11. Zeile und 4 . und 13. Zeile von unten: `s.Unlock()`
- S. 122: 17. Zeile von unten: `s.Lock()`
- S. 124: 9. Zeile: `type addSemaphore struct`
- S. 128: Nach der 6. Zeile ist folgende Zeile einzufügen `import "nU/sem//` das Paket mit dem ADT Semaphoreu
- S. 128: 12. Zeile: `[m]sync.Mutex` ist durch `[M]sem.Semaphore` zu ersetzen

- S. 128: 19. Zeile: `x.b[i].Lock()` ist durch `x.b[i] = sem.New()` zu ersetzen
- S. 128: 32. Zeile: `x.b[n].Lock()` ist durch `x.b[n].P()` zu ersetzen
- S. 129: 2. Zeile: `x.b[n].Unlock()` ist durch `x.b[n].V()` zu ersetzen
- S. 148: Zwischen Zeile 5 und 6 ist „for {“ einzufügen.
- S. 156: 8. Zeile: $(n_W = 0) \wedge (n_R < \infty) = (n_W = 0)$
- S. 157: 1. Zeile: $(k < N)$
- S. 158: 1. Zeile: „emph“ ist zu streichen.
- S. 167: 12. Zeile von unten: und `StmtSpectrum` gebündelt,
- S. 230: 8. Zeile von unten: und die `Unlock`-Operationen
- S. 241: 18. Zeile: „(und ggf. p)“ ist zu streichen
- S. 243: 1./2. Zeile von unten: das Bedingungsspektrum
- S. 293: 4. Zeile von unten: oder dem ...
- S. 295: 5. Zeile von unten: empfängt
- S. 298: In der Typdeklaration von `netChannel` sind drei Komponenten einzufügen:
- ```
isServer, oneOne bool
net.Listener
error
```
- S. 312: Auch hier sind in der Typdeklaration von `netChannel` einzufügen:
- ```
isServer, oneOne bool
net.Listener
error
```
- S. 326: 18. Zeile: `r := rpc.New (input, output, 1, hostname, ...`
- S. 327: 12. Zeile: `r := rpc.New (input, output, 1, hostname, ...`
- S. 366: 13. Zeile: `marked[v] = true`
- S. 397: 13. Zeile von unten: „emph“ ist zu streichen.
- S. 404: 11. Zeile von unten: `G8ring`