

- Ausführung 2
  - Continue: Deaktivieren von CPU 1,  $y$ ; Eingabe:  $1^+$
  - Ergebnis:  $t_{11}$  und  $t_{12}$  werden beendet, dann  $t_{111}$ , dann  $t_{21}$
- Ausführung 3
  - Continue: Deaktivieren von CPU 1,  $y$ ; Eingabe:  $1, 2, 1^3, 3, 1^+$
  - Ergebnis: Suspendieren der Threads von  $t_{g1}, t_{21}$  in Ausführung bis zum Resume von  $t_{g1}$
- Regeln für die Festlegung von Thread-Prioritäten
  - Initial erhält ein Thread  $t$  die kleinere der folgenden Prioritäten:
    - die Priorität  $p'$  des Threads  $t'$ , der ihn erzeugt
    - die maximale Priorität  $pg$  der Threadgruppe  $tg$ , zu der der neu erzeugte Thread  $t$  gehört
  - Bei dynamischer Modifikation seiner Priorität mit `setPriority (int p)` erhält ein Thread  $t$  die kleinere der folgenden Prioritäten:
    - die spezifizierte Priorität  $p$
    - die maximale Priorität  $pg$  seiner Threadgruppe  $tg$
- Regeln zur Festlegung von Threadgruppen-Prioritäten
  - Initial erhält eine Threadgruppe  $tg$  als Priorität  $pg$  die Priorität des Vorgängers
  - Bei dynamischer Modifikation ihrer Priorität mit `setMaxPriority (int pg)` erhält eine Threadgruppe  $tg$  die kleinere der folgenden Prioritäten:
    - die spezifizierte Priorität  $pg$
    - die maximale Priorität  $pg'$  der Vorgänger-Threadgruppe  $tg'$

Danach wird dieser Vorgang rekursiv auf alle Threadgruppen von  $tg$  angewendet.

Die Priorität der Threads der Threadgruppen bleibt unverändert.