

FSST

Felix Hofinger

October 2022

Inhaltsverzeichnis

1	Exception Handling	3
2	Prozesse	4
2.1	Eigenschaften	5
2.2	Repräsentation von Prozessen	5
2.3	Process-Scheduler	6
2.3.1	Ziele des Schedulers	6
2.3.2	Grundsysteme für Scheduling	6
2.3.3	Strategien	7
2.3.3.1	Eingangsreihenfolge (first come, first serve) . . .	7
2.3.3.2	Kürzester Job zuerst (shortest job first)	7
2.3.3.3	Zeitscheibenverfahren (round robin)	7
2.3.3.4	Prioritätssteuerung	7

1 Exception Handling

Exceptions sind Objekte:

- Felder: Informationen über den Fehler
- Methoden: Ausgabe der Fehlerinformationen, etc.

```
try {  
    ...  
    p();  
    ...  
} catch (Exception e) {  
    System.out.println(e.toString());  
}  
  
void p() throws Exception {  
    ...  
    throw new Exception();  
    ...  
}
```

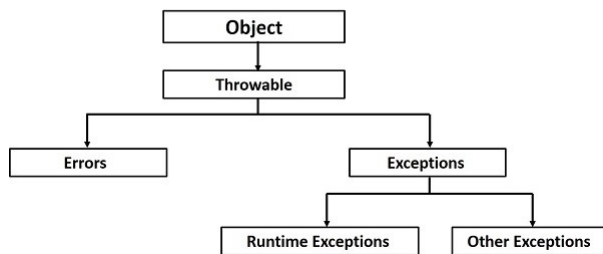


Tabelle 1: Exception Hierarchy

```
try {  
    ...  
} catch(MyException e) { // fängt MyException u. U.  
} catch(Exception e) { // fängt Exception u. U.  
} catch(Throwable e) { // fängt alle Exceptions  
}
```

2 Prozesse

- Ablauf eines sequentiellen Programms
- Kontext / Prozessumgebung
- Betriebssysteme, die Programme mit mehreren parallel ausführbaren Programmabschnitten zulassen.
 - > Threads

2.1 Eigenschaften

- Aus Sicht des Betriebssystems: Objekte die die CPU Kapazität zugeteilt bekommen
- Multitasking Betriebssystem: mehrere Prozesse gleichzeitig im Speicher
Bei ein-Prozessor System:
 - > immer nur ein Prozess aktiv
 - > ständig zwischen Prozessen wechseln
 - > quasi-paralleler Ablauf
- Parallel bzw. quasi-paralleler Ablauf -> nebenläufigkeit
- Zustand
- Kontext: CPU Register, Ressourcen (Arbeitsregister, Dateien, ...)
- "Kinderprozesse" > Prozessbäume
- Prozesse können miteinander kommunizieren
 - > Interprozesskommunikation
- Prozesse können voneinander abhängen
 - > kooperierende Prozesse
 - > Suchorganisation
- Priorität

2.2 Repräsentation von Prozessen

- für jeden Prozess gibt es einen Prozess-Struktur-Block (PCB)
- PCB: Prozess-ID, Priorität, Status, Kontext, ...
 - > werden in verketteter Liste verwaltet (Prozesstabelle)

Zuteilung der CPU an einen bereiten Prozess macht der Scheduler.

-> unterschiedliche Strategien

Prozesswechsel:

- 1, Retten des Prozesskontext
- 2, Aktivieren des neuen Kontext
- 3, PCB und Waketlist aktualisieren
 - > anderer Prozess läuft weiter

2.3 Process-Scheduler

- macht nur bei Multitasking Systemen Sinn
- wählt aus der Liste von bereiten Prozessen den nächsten aktiven Prozess aus
- Zeitpunkte:
 - > Erzeugung neuer Prozesse
 - > Beendung eines Prozesses
 - > Blockierung des aktiven Prozesses
 - > Unterbrechung durch I/O-Gerät
 - > Unterbrechung durch Timer

2.3.1 Ziele des Schedulers

- Entscheiden, wer „rechnen“ darf
- Scheduling Algorithmus
- Gerechtigkeit
- Effizienz
- Antwortzeit
- Verweilzeit
- Durchsatz
- Terminerfüllung

2.3.2 Grundsysteme für Scheduling

- | kooperatives Multitasking
non-preemptive | verdrängendes Multitasking
preemptive |
|---|--|
| <ul style="list-style-type: none">• aktiver Prozess gibt CPU freiwillig her• geringer Verwaltungsaufwand<ul style="list-style-type: none">-> Gefahr: Blockierung durch unkooperativen Prozess | <ul style="list-style-type: none">• dem aktiven Prozess wird die CPU vom Scheduler entzogen (z.B. nach Zeit ablauf -> Timer interrupt)• hoher Verwaltungsaufwand• kein Blockieren |

2.3.3 Strategien

2.3.3.1 Eingangsreihenfolge (first come, first serve)

- Zuteilung d. CPU in Startreihenfolge
- neue Prozesse „hinten anstellen“
- Wechsel: Prozess fertig oder blockiert

-> non-preemptive

-> super Systemauslastung

-> schlechtes Antwort- Zeitverhältnis

-> einfach

2.3.3.2 Kürzester Job zuerst (shortest job first)

- Zuteilung d. CPU nach bekannter/geschätzter Rechenzeit d. Prozesse

2.3.3.3 Zeitscheibenverfahren (round robin)

- CPU wird zyklisch nacheinander allen bereiten Prozessen für eine bestimmte Zeit zugeordnet

-> preemptive

-> alle haben gleiche Priorität

2.3.3.4 Prioritätssteuerung

- Jeder Prozess hat Priorität
- vergabe der CPU in absteigender Priorität
- Prozess mit niedriger Priorität bekommt die CPU erst dann, wenn alle Prozesse mit höherer Priorität fertig sind
- Ein bereit-werdender Prozess mit höherer Priorität als aktiver Prozess
 - > verdrängt den aktiven Prozess