

Bedienungsanleitung – CPU Task Scheduler Simulator

1. Überblick

Der **CPU Task Scheduler Simulator** ist ein Lern- und Visualisierungstool für Betriebssystem-Konzepte.

Es ermöglicht die Simulation verschiedener **Scheduling-Algorithmen** (FCFS, SJF, RR, Priority RR) in einem **Single-CPU-Modell**.

Das Programm zeigt:

- eine **Prozessstabelle** mit den eingegebenen Prozessen,
- eine **Statistik** (z. B. Durchlaufzeit, Wartezeit),
- sowie eine **grafische Gantt-Darstellung** des CPU-Zeitplans.

2. Benutzeroberfläche

Linke Seite

- **Process Table**
Liste aller eingetragenen Prozesse mit ihren Eigenschaften (Arrival Time, Burst Time, Priority).
- **Statistics**
Anzeige der berechneten Kennzahlen nach der Simulation:
 - Completion Time
 - Turnaround Time
 - Waiting Time
 - Durchschnittswerte
- **Gantt Chart**
Visualisierung des CPU-Zeitplans als Zeitachse. Jeder Balken zeigt an, wann welcher Prozess/Thread ausgeführt wurde.

Rechte Seite (Controls)

- **Arrival Time**
Zeitpunkt (in Zeiteinheiten), wann der Prozess im System ankommt.
- **Burst Time**
Rechenzeit (CPU-Zeit), die der Prozess benötigt.
- **Priority**
Statische Priorität (für Priority-basiertes Scheduling).
 - Niedrigere Zahl = höhere Priorität.
- **Insert Process**
Fügt den Prozess mit den angegebenen Werten in die Tabelle ein.
- **Algorithm**
Auswahl des Scheduling-Algorithmus:

- FCFS (First Come First Serve)
- SJF (Shortest Job First)
- RR (Round Robin)
- RRP (Round Robin mit Priorität)
- **Quantum**
Zeitscheibe für Round-Robin-Algorithmen (aktiv bei RR/RRP).
- **Scheduling Mode**
 - *Process Only*: Nur Prozesse werden berücksichtigt.
 - *Thread Kernel*: Threads werden wie eigenständige Prozesse behandelt.
 - *Thread User*: Prozesse werden gescheduled, ihre Threads intern per RR.
- **Thread Quantum**
Zeitscheibe, die bei *User-Level Thread Scheduling* angewendet wird.
- **Context Switch**
Dauer eines Kontextwechsels zwischen zwei Prozessen/Threads.
- **Start Simulation**
Startet die Simulation und erzeugt Gantt-Diagramm + Statistiken.
- **Reset**
Löscht alle Eingaben und setzt den Simulator zurück.

3. Schritt-für-Schritt Nutzung

1. **Prozess eingeben**
 - Arrival Time, Burst Time und ggf. Priority setzen.
 - Mit *Insert Process* hinzufügen.
 - Wiederhole dies für alle gewünschten Prozesse.
2. **Algorithmus auswählen**
 - Im Dropdown *Algorithm* den gewünschten Scheduling-Algorithmus wählen.
 - Falls Round Robin → Quantum anpassen.
3. **Modus und Parameter setzen**
 - Scheduling Mode: Prozess-/Thread-Ebene auswählen.
 - Thread Quantum einstellen (nur bei User-Thread-Modus relevant).
 - Context Switch ggf. auf >0 setzen, um Overhead zu simulieren.
4. **Simulation starten**
 - *Start Simulation* klicken.
 - Ergebnis:

- Prozesstabelle mit berechneten Metriken.
- Statistik-Fenster mit Durchschnittswerten.
- Gantt-Chart mit Ablaufvisualisierung.

5. Neustart

- Mit *Reset* alle Eingaben und Ergebnisse löschen und neue Simulation beginnen.

4. Beispiel

Prozesse eingeben:

- P1: Arrival=0, Burst=5
- P2: Arrival=2, Burst=3
- P3: Arrival=4, Burst=2

Algorithmus: FCFS

Context Switch: 0

Ergebnis:

- Reihenfolge: P1 → P2 → P3
- Gantt: [P1: 0–5] [P2: 5–8] [P3: 8–10]
- Durchschnittliche Wartezeit und Durchlaufzeit werden angezeigt.

5. Tipps & Hinweise

- Für **RR** und **RRP** ist der Quantum-Wert entscheidend für Fairness und Performance.
- Mit **Context Switch >0** wird sichtbar, wie Scheduling-Overhead die Gesamtdauer beeinflusst.
- **Thread-Scheduling-Modi** veranschaulichen den Unterschied zwischen Kernel- und User-Level-Threadverwaltung.