

BACHELORARBEIT

**Entwurf und Implementierung einer  
Web-basierten Anwendung zur  
interaktiven Simulation von  
Algorithmen für das Scheduling von  
Prozessen**

*Rachel Ringe*

Studiengang: Digitale Medien  
Fachbereich 3

Erstkorrektor: Olaf Bergmann  
Zweitkorrektor: Prof. Dr. Ute Bormann

Anmeldung: 20. Mai 2020  
Abgabe: 03. September 2020

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>Eigenständigkeitserklärung</b>                               | <b>iv</b> |
| <b>1 Einleitung</b>   | <b>1</b>  |
| 1.1 Aufgabenstellung und Zielsetzung . . . . .                  | 1         |
| 1.2 Inhaltlicher Aufbau der Arbeit . . . . .                    | 2         |
| <b>2 Existierende Programm-Alternativen</b>                     | <b>3</b>  |
| <b>3 Anforderungen</b>  | <b>4</b>  |
| <b>4 Verhalten von Algorithmen zum Scheduling von Prozessen</b> | <b>6</b>  |
| 4.1 Round Robin . . . . .                                       | 6         |
| 4.2 Shortest Remaining Job First . . . . .                      | 7         |
| 4.3 Kontosystem . . . . .                                       | 7         |
| 4.4 Linuxscheduler . . . . .                                    | 8         |
| 4.5 Custom . . . . .  | 9         |
| <b>5 Implementierung</b>  | <b>10</b> |
| 5.1 Dynamisches Menü . . . . .                                  | 12        |
| 5.2 Berechnung des Algorithmus-Verhaltens . . . . .             | 17        |
| 5.2.1 Round Robin . . . . .                                     | 21        |
| 5.2.2 Shortest Remaining Job First . . . . .                    | 21        |
| 5.2.3 Kontosystem . . . . .                                     | 22        |
| 5.2.4 Linuxscheduler . . . . .                                  | 22        |
| 5.2.5 Custom . . . . .  | 23        |
| 5.3 Diagramm . . . . .  | 25        |
| 5.3.1 Diagramme in chart.js . . . . .                           | 28        |
| 5.3.2 Finale Implementierung des Diagramms . . . . .            | 29        |
| 5.3.3 Anpassung des Diagramms an veränderte Parameter . . . . . | 30        |
| 5.3.4 Schrittweises Navigieren durch das Diagramm . . . . .     | 33        |

|          |   |           |
|----------|---|-----------|
| 5.4      | Paketisierung . . . . .   | 34        |
| 5.5      | Speicherung des Fortschritts . . . . .                              | 34        |
| <b>6</b> | <b>Test der Implementierung</b>                                     | <b>37</b> |
| <b>7</b> | <b>Fazit und Ausblick</b>   | <b>41</b> |
|          | <b>Quellenverzeichnis</b>   | <b>43</b> |
|          | <b>Anhang</b>   | <b>44</b> |
|          | Testprotokoll . . . . .   | 44        |
|          | Vergleichswerte zur Korrektheit des Algorithmusverhaltens . . . . . | 61        |

# Abbildungsverzeichnis

|      |   |    |
|------|---|----|
| 5.1  | Tabellarische Informationsanzeige neben dem Diagramm . . . . .                      | 12 |
| 5.2  | Menü im Ausgangszustand . . . . .   | 13 |
| 5.3  | Custom-Eingabeoptionen im Menü . . . . .  | 14 |
| 5.4  | Menü ohne Eingabefeld für Zeitscheibenlänge . . . . .                               | 15 |
| 5.5  | Eingabezeile im Menü ohne Prioritätsfeld . . . . .                                  | 15 |
| 5.6  | Eingabezeile im Menü mit Prioritätsfeld . . . . .                                   | 15 |
| 5.7  | Warteanzeige im Menü . . . . .  | 16 |
| 5.8  | Fehleranzeige im Menü . . . . .   | 17 |
| 5.9  | Pseudocode der Berechnung . . . . .   | 19 |
| 5.10 | Diagramm mit Lücke bei später startenden Prozessen . . . . .                        | 20 |
| 5.11 | Diagramm mit dynamischer Zeitscheibenlänge . . . . .                                | 24 |
| 5.12 | Horizontal ausgerichtete „Range Column Chart“ mit „apexcharts“ gezeichnet . . . . . | 26 |
| 5.13 | Erster Entwurf mit Verwendung eines „Stacked Bar Chart“-Diagramms .                 | 27 |
| 5.14 | Koordinaten-Format für „Stacked Bar Chart“ . . . . .                                | 27 |
| 5.15 | Finale Version des Diagramms . . . . .  | 28 |
| 5.16 | Verwendete Farbpalette . . . . .  | 29 |
| 5.17 | Diagramm, das Test-Farbpalette verwendet . . . . .                                  | 30 |
| 5.18 | Anzeigefehler bei Überschreitung der maximalen Canvas-Größe . . . . .               | 32 |
| 5.19 | Anwendung beim ersten Besuch . . . . .  | 35 |
| 6.1  | Anzeige der Ergebnisse der Unit-Tests . . . . .                                     | 38 |
| 6.2  | Anzeigefehler in Firefox . . . . .  | 38 |

## ERKLÄRUNG

Ich versichere, den Bachelor-Report oder den von mir zu verantwortenden Teil einer Gruppenarbeit\*) ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

\*) Bei einer Gruppenarbeit muss die individuelle Leistung deutlich abgrenzbar und bewertbar sein und den Anforderungen entsprechen.

Bremen, den 02.09.20

R. Ringe

(Unterschrift)

# 1 Einleitung

Das Scheduling von Prozessen ist elementarer Bestandteil eines Betriebssystems und Inhalt vieler grundlegender Lehrveranstaltungen des Themas. In der Praxis existieren viele verschiedene Algorithmen, die das Scheduling-Verhalten bestimmen. Diese rangieren von simplen und einfachen Varianten, wie „Round Robin“ bis zu sehr komplizierten Verfahren wie dem Linuxscheduler, die viele verschiedene Variablen berücksichtigen. In Lehrveranstaltungen wird das Verhalten dieser Algorithmen oft an einigen wenigen, statischen Beispielen erklärt. Um Veränderungen in diesem Verhalten bei Veränderung der zugrundeliegenden Parameter zu beobachten und zu vergleichen, muss dieses Verhalten berechnet und von Hand in ein Diagramm übertragen werden, was häufig zeitaufwändig ist. Eine Anwendung, die das Verhalten verschiedener Algorithmen zum Scheduling von Prozessen simuliert, könnte das Verständnis für das Verhalten der Algorithmen unter verschiedenen Umständen erheblich erleichtern.

## 1.1 Aufgabenstellung und Zielsetzung

Das Ziel dieser Arbeit ist der Entwurf und die Implementierung einer interaktiven, webbasierten Anwendung, die das Verhalten verschiedener Algorithmen zum Scheduling von Prozessen visuell abbildet. Die Anwendung soll in der Lehrveranstaltung „Technische Grundlagen der Informatik“ verwendet werden, um das Verhalten von Algorithmen zum Scheduling von Prozessen durch eine Visualisierung zu verdeutlichen. Dabei soll der Nutzer die Möglichkeit haben, verschiedene Einstellungen zu verändern und die Effekte auf das Verhalten der Algorithmen zu beobachten. Entsprechend muss die Anwendung gängige Designs zur User-Experience umsetzen, um möglichst ansprechend und übersichtlich zu sein. Eine weitere Anforderung ist die einfache Paketisierung sowie die Installation der Anwendung. Hierdurch kann das Verteilen an die Studenten sowie die Verwendung der Anwendung vereinfacht werden und Probleme durch verschiedene Betriebssysteme und Systemversionen minimiert werden.

## **1.2 Inhaltlicher Aufbau der Arbeit**

Im Folgenden werden zunächst bereits existierende Alternativen zu der Anwendung vorgestellt und erläutert, weshalb die Erstellung dieses Programms sinnvoll ist. Anschließend werden die Anforderungen an das System erklärt. Danach wird die Funktionsweise der implementierten Algorithmen beschrieben, um eine Basis für die Erklärungen im nächsten Kapitel zu bilden. Im fünften Kapitel der Arbeit wird die Implementierung genauer beschrieben und erläutert. Darauffolgend wird wiedergegeben, wie die Anwendung getestet wurde und die Testergebnisse dargelegt. Die Arbeit schließt mit einem Fazit ab, in dem über die Erfüllung der Anforderungen und mögliche weitere Verbesserungen an der Anwendung reflektiert wird.

## 2 Existierende Programm-Alternativen

Es existieren bereits einige private Programmierprojekte, deren Ziel die Simulation von Algorithmen zum Prozess-Scheduling ist. Ein Teil dieser Projekte ist sehr simpel und besteht ausschließlich aus einer Textausgabe, ohne oder mit sehr eingeschränkten Möglichkeiten Parameter zu verändern. Viele andere sind unübersichtlich und enthalten eine Vielzahl von verschiedenen Buttons und Eingabefeldern, um Einstellungen zu verändern. Dazu kommen mehrere Anzeigen verschiedener Werte zu jedem Zeitpunkt in der Visualisierung, die Darstellungen von „Sleep“- und „Ready Queue“ sowie die eigentliche Visualisierung des Scheduling. Alle diese Optionen, Anzeigen und Grafiken sind häufig in einem einzigen, recht kleinen Programmfenster zusammengefasst.

Weiterhin sind alle diese Simulatoren ausführbare Programme. Dies könnte bei der Anwendung in der Lehre für Schwierigkeiten sorgen, da Studenten mit einer Vielzahl unterschiedlicher Betriebssysteme, Systemversionen und Computern arbeiten. Dabei kann es zu Problemen bei der Ausführung von Programmen kommen, die durch eine web-basierte Anwendung umgangen werden könnten. Zudem müssten für eine Verwendung in der Lehre zunächst die nötigen Lizenzen von den Programmierern erworben werden. Eine weitere Möglichkeit, das Verhalten von Algorithmen zum Scheduling von Prozessen zu simulieren, ist das Simulations-Betriebssystem Xv6, das vom MIT zu Lehrzwecken bereitgestellt wird<sup>1</sup>. Dieses benutzt einen simplen Round-Robin-Scheduler (vgl. Cox 2011 S. 60), der umgeschrieben werden kann, um andere Algorithmen zu verwenden. Xv6 ist allerdings sowohl in der Einrichtung als auch in der Benutzung sehr kompliziert und benötigt eine erhebliche Einarbeitungszeit, um den Aufbau des Programms zu verstehen und eine erfolgreiche Verwendung zu ermöglichen. Auch erzeugt es lediglich Textausgaben im Terminal, die bei vielen Prozessen oder langen Prozesslaufzeiten schnell unübersichtlich werden können und generell weniger anschaulich sind als eine visuelle Darstellung. Des Weiteren ist der Scheduler lediglich ein kleiner Teil der Funktionalität von Xv6, so dass ein sehr großer Aufwand betrieben werden müsste, um nur einen kleinen Teil des Programms sinnvoll nutzen zu können.

---

<sup>1</sup><https://github.com/mit-pdos/xv6-riscv/blob/riscv/LICENSE>

# 3 Anforderungen

Die folgenden Anforderungen an das System wurden im Vorfeld der Arbeit festgelegt. Die Anwendung soll web-basiert sein, um eine einfache Verbreitung zu ermöglichen und Probleme mit verschiedenen Betriebssystemen und Geräten zu vermeiden. Auch soll sie als Single-Page-Application (SPA) entwickelt werden, also eine Anwendung, in der aller notwendige Code nur einmal geladen und die Darstellung dann durch JavaScript verändert wird (vgl. Flanagan 2006, S.497). Der Stand der Anwendung soll im Browser gespeichert werden, damit der Nutzer die Arbeit unterbrechen und zu einem späteren Zeitpunkt fortsetzen kann. Zudem sollte beachtet werden, dass die Anwendung später gut paketisierbar sein muss, um sie möglichst problemlos an Studenten weitergeben zu können. Folgende Algorithmen zum Prozess-Scheduling sollen von der Anwendung simuliert werden:

- Shortest Remaining Job First
- Round Robin
- Kontomodell mit Basisprioritätswerten
- Eine vereinfachte Version des Linux-Scheduler-Algorithmus, der ebenfalls prioritätsbasiert ist
- Eine Möglichkeit, Formeln zur Berechnung der Priorität selbst einzugeben

Diese Algorithmen wurden ausgewählt, da sie häufig verwendet werden und oft die Basis für andere, kompliziertere Rechengänge bilden. Außerdem sind die ersten drei genannten Algorithmen Teil der Vorlesung „Technische Grundlagen der Informatik“, in der die Anwendung eingesetzt werden soll.

Verschiedene Parameter, die das Verhalten der Algorithmen beeinflussen, sollen vom Nutzer einzustellen sein. Dabei können allgemeine Einstellungen vorgenommen werden, wie die Anzahl der Prozesse und die Länge der Zeitscheiben. Des Weiteren können für jeden einzelnen Prozess Startzeitpunkt, Gesamtlaufzeit und – sofern relevant – auch die

Basispriorität eingestellt werden. Ist dies im Zeitrahmen der Arbeit möglich, besteht die Gelegenheit, weitere eventuelle Einstellungen zu implementieren, wie zum Beispiel die Aufteilung der Prozesse auf mehrere CPU-Kerne oder I/O-Interrupts bei den einzelnen Prozessen.

Die Anwendung sollte visuell ansprechend und übersichtlich gestaltet werden und eine Benutzeroberfläche haben, die den Benutzer trotz der vielen möglichen Einstellungen nicht überwältigt. Die Darstellung der Scheduling-Abläufe soll visuell sein, zum Beispiel in Form eines Gantt-Diagramms. Sie soll als statisches Bild des Endergebnisses möglich sein sowie als eine schrittweise Abbildung, durch die der Nutzer sich mit verschiedenen Buttons bewegen kann. Die Schritte der Simulation sollen auch als Animation abspielbar sein. Die Anzeige von zusätzlichen Informationen für die einzelnen Schritte, wie zum Beispiel die „Ready Queue“, aktueller Prioritätswert und verbleibende Laufzeit, ist sinnvoll und soll als Funktion vorhanden sein.

# 4 Verhalten von Algorithmen zum Scheduling von Prozessen

In diesem Kapitel der Arbeit wird auf die verschiedenen, in der Anwendung implementierten Arten von Algorithmen zum Scheduling von Prozessen näher eingegangen. Dabei wird ihr Verhalten erläutert und ihre Besonderheiten und Unterschiede zueinander gezeigt. Dieses Kapitel dient dazu, ein grundlegendes Verständnis der Algorithmen zu geben. Dies ist notwendig, da im folgenden Kapitel, das die Implementierung und verschiedene Design-Entscheidungen erklärt, auf die verschiedenen Eigenschaften der Algorithmen eingegangen wird.

Auf einem Computer existieren mehrere Programme und Prozesse, die gleichzeitig Rechenzeit auf der CPU benötigen, um ihre Aufgaben auszuführen. Wenn weniger CPU-Kerne zur Verfügung stehen als Prozesse Rechenzeit benötigen, muss entschieden werden, in welcher Reihenfolge und für wie lange die Prozesse Rechenzeit erhalten. Diese Entscheidung wird vom sogenannten „Scheduler“ mit Hilfe verschiedener Algorithmen zum Scheduling von Prozessen getroffen (vgl. Tanenbaum 2015, S. 149).

Dabei ist das Ziel, die vorhandenen Ressourcen möglichst fair zu verteilen und dabei größtmögliche Effizienz und kurze Antwortzeiten für den Benutzer zu gewährleisten (vgl. Ehes 2005, S. 275). Die Prozesse, die auf Rechenzeit warten, werden in einer Warteschlange, einer sogenannten „Ready Queue“, gespeichert. Je nach Algorithmus enthält diese nur minimale Informationen wie verbleibende Laufzeit oder Gesamtlaufzeit oder detaillierte Informationen wie verschiedene Prioritätswerte.

## 4.1 Round Robin

Das Round-Robin-Scheduling ist eine sehr einfache und weit verbreitete Art des Scheduling. Neue Prozesse werden am Ende der Warteschlange eingereiht und Prozesse werden immer von ihrem Beginn entnommen. Dabei wird jedem Prozess eine feste Zeitspanne Rechenzeit, eine sogenannte Zeitscheibe oder Quantum, in der CPU zugewiesen (vgl.

Ehres 2005, S. 278). Die Länge dieser Zeitscheibe ist dabei immer gleich (vgl. Tanenbaum 2015, S. 158–159). Nach Ablauf dieses Zeitintervalls wird ein Prozesswechsel durchgeführt, der nächste Prozess wird der Warteschlange entnommen und erhält Rechenzeit. Hat der erste Prozess noch verbleibende Rechenzeit, wird er am Ende der Warteschlange wieder eingereiht. Wird ein Prozess unterbrochen oder beendet seine Laufzeit, bevor das Quantum komplett beendet wurde, wird der Prozesswechsel vorzeitig durchgeführt (vgl. Ehres 2005, S. 278).

## 4.2 Shortest Remaining Job First

Eine ebenfalls weit verbreitete Prozess-Scheduling-Strategie ist „Shortest Remaining Job First“, eine weiterentwickelte Variante von „Shortest Job First“. Bei der ursprünglichen Strategie wird davon ausgegangen, dass alle Prozesse zum gleichen Zeitpunkt beginnen. Dann wird der Prozess mit der kürzesten Gesamtlaufzeit ausgewählt und erhält die CPU, bis der Prozess abgeschlossen ist. Danach erhält der Prozess mit der zweitkürzesten Laufzeit die CPU. Dieses Verhalten wird fortgesetzt bis keine Prozesse mehr verbleiben. Das „Shortest Remaining Job First“-Verfahren ist auch in einer Situation nutzbar, in der neue Prozesse zu verschiedenen Zeitpunkten verfügbar werden. Wird ein neuer Prozess der Ready Queue hinzugefügt, wird überprüft, ob seine verbleibende Laufzeit kürzer ist als die des Prozesses, der aktuell die CPU nutzt. Ist dies der Fall, wird der Prozess durch den neuen ersetzt. Der alte Prozess wird der Ready Queue wieder hinzugefügt. Erreicht ein Prozess das Ende seiner Gesamtlaufzeit, wird aus der Ready Queue der Prozess mit der geringsten verbleibenden Laufzeit entnommen (vgl. Tanenbaum 2015, S. 157).

## 4.3 Kontosystem

Bei den bisher vorgestellten Algorithmen zum Scheduling wird grundsätzlich angenommen, dass alle Prozesse die gleiche Wichtigkeit haben und durch gleiche Behandlung eine faire Verteilung der CPU-Rechenzeit gewährleistet ist. In der Praxis gibt es allerdings oft Prozesse, die eine größere Priorität als andere haben, die möglicherweise bereits länger auf die CPU warten oder eine kürzere Gesamtlaufzeit haben. Um dieses Problem zu lösen, wurden verschiedene Algorithmen entwickelt, die an Hand eines Basis-Prioritätswerts die Wichtigkeit eines Prozesses berücksichtigen, aber gleichzeitig

sicherstellen, dass weniger wichtige Prozesse nicht benachteiligt werden und ebenfalls Rechenzeit erhalten. Um dieses Ziel zu erreichen, werden, je nach System und Anwendungszweck, verschiedene Formeln verwendet, um die dynamische Priorität eines Prozesses zu errechnen (vgl. Tanenbaum 2015, S. 159).

Die in dieser Anwendung implementierte Variante des prioritätsbasierten Scheduling von Prozessen verwendet eine aus der Lehrveranstaltung „Technische Grundlagen der Informatik“ bekannte Formel.

Dabei wird zu Beginn jeder gleichlangen Zeitscheibe ein Kontowert berechnet. Dazu wird die Nutzungszeit im aktuellen Intervall auf den bisherigen Kontowert aufaddiert. Anschließend wird dieser Wert halbiert. Der Prioritätswert ergibt sich dann aus der Addition des so errechneten Kontowerts und der beim Prozessstart zugeteilten Basispriorität des Prozesses. Ausgewählt wird dann der Prozess mit der geringsten dynamischen Priorität.

## 4.4 Linuxscheduler

Der Scheduler im Betriebssystem Linux Version 2.6 teilt die Prozesse in verschiedene Kategorien ein: Echtzeit-Prozesse, die nach dem Algorithmus „Round-Robin“ oder dem Prinzip „First In First Out“ behandelt werden, und sonstige konventionelle Prozesse. Letztere werden anhand eines Prioritätssystems behandelt, das dem im vorherigen Abschnitt ähnlich ist. Jeder Prozess erhält zu Beginn eine statische Priorität zwischen 100 und 139, die genutzt wird, um verschiedene Werte auszurechnen, die im Scheduling-Verfahren verwendet werden (vgl. Bovet 2005, 259–265). Die dynamische Priorität wird zu Beginn jeder Zeitscheibe mit folgender Formel berechnet:

$$\text{Dynamische Priorität} = \max(100, \min(\text{statische Priorität} - \text{Bonus} + 5, 139))$$

Der Bonus ist ein Wert zwischen 0 und 10, der abhängig von der bisherigen durchschnittlichen Schlafzeit des Prozesses ist. Er kann errechnet werden durch:

$$\text{Bonus} = \min(\text{durchschn. Schlafzeit in ms}/100, 10)$$

Ein weiterer Unterschied zu den bisher beschriebenen Algorithmen ist, dass im Linux-Prozess-Scheduling-Algorithmus nicht Zeitscheiben mit einer festgelegten Länge verwendet, sondern diese basierend auf der statischen Priorität berechnet werden. Dazu wird folgende Formel verwendet:

$$\text{Zeitscheibe (in ms)} = \left\{ \begin{array}{l} (140 - \text{statische Priorität}) * 20 \text{ wenn } \text{statische Priorität} < 120 \\ (140 - \text{statische Priorität}) * 5 \text{ wenn } \text{statische Priorität} \geq 120 \end{array} \right\}$$

In dieser Anwendung wurde lediglich das Scheduling der konventionellen Prozesse implementiert, da die Anwendung dazu dient, einen Überblick über verschiedene Algorithmen zum Scheduling von Prozessen zu geben und deren Verhalten unter verschiedenen Umständen abzubilden. Dabei ist das Ziel, Studenten die Vorgänge im Inneren von Betriebssystemen näher zu bringen und nicht diese exakt abzubilden. Die anderen im Linux-Scheduler vorhandenen Versionen sind im Fall von „Round Robin“ bereits an anderer Stelle in der Anwendung implementiert und im Fall von „First In First Out“ sehr simpel, so dass eine visuelle Darstellung und Implementierung in der Anwendung eher weniger sinnvoll und notwendig ist. Das Verhalten des Linux-Schedulers hingegen ist eine interessante Alternative zu dem vereinfachten Kontosystem, das im vorherigen Abschnitt beschrieben wurde. So kann durch den Nutzer beobachtet werden, dass prioritätsbasierte Systeme sehr unterschiedliche Ergebnisse liefern können. Auch gibt dies einen Eindruck von der Scheduling-Funktion eines realen Betriebssystems, ohne extensives Wissen über Linux oder die verschiedenen Prozessarten vorauszusetzen. Des Weiteren verwendet der Linux-Scheduler dynamische Zeitscheibenlänge, was ein optional gewünschtes Feature ist. Auch existiert so ein Beispiel, an dem der Nutzer sehen kann, wie ein solches System arbeitet, ohne selbst eine Formel für die Berechnung dieser Zeitscheibenlängen entwerfen zu müssen (vgl. Bovet 2005, 259–265).

## 4.5 Custom

Zuletzt bietet die Anwendung dem Nutzer die Möglichkeit, auf Basis des Wissens über die bisher vorgestellten Algorithmen, einen eigenen Algorithmus zu entwickeln. Dabei ist es ihnen möglich Formeln einzugeben, um die dynamische Priorität und optional einen Kontowert und die Zeitscheibenlänge zu berechnen.

# 5 Implementierung

In diesem Kapitel der Arbeit werden das Design der Webanwendung sowie die Implementierung im Detail erklärt. Dabei wird auf die getroffenen Designentscheidungen und den Arbeitsprozess eingegangen und welche Ansätze zur Lösung der Fragestellung in der Programmierung gewählt wurden

Zunächst wird das Menü geschildert, in dem der Nutzer die gewünschten Einstellungen vornehmen kann. Anschließend wird die Berechnung der einzelnen Schritte in der Simulation des Algorithmen-Verhaltens erläutert. Darauffolgend wird das Zeichnen des Diagramms aus den vorher errechneten Daten erklärt. Auch die Funktionsweise der verwendeten JavaScript-Bibliothek wird thematisiert. Zum Abschluss wird die geplante Paketisierung der Anwendung sowie das Speichern des Arbeitsfortschritts beschrieben. Vor der Implementierung wurde ein Entwurf der Systemstruktur konzipiert. Dabei war die Einteilung in folgende Module vorgesehen:

- Eingabe der Daten über dynamisch anpassbare Benutzeroberfläche
- Berechnung der einzelnen Schritte der Simulation
- Zeichnen und Anpassen des Diagramms sowie Anzeige von Informationen
- Navigation durch das Diagramm durch eine Benutzeroberfläche

Die Benutzeroberfläche der Web-Anwendung wurde mit dem Framework Bootstrap<sup>1</sup> gestaltet. Dieses enthält eine Vielzahl von frei verfügbaren Design-Vorlagen für ganze Webseiten oder einzelne Bestandteile und Design-Elemente. Auch passen sich mit Bootstrap gestaltete Webseiten automatisch der Größe des Browser-Fensters an. Die Verwendung des Frameworks ermöglicht das einfache und schnelle Erstellen einer visuell ansprechenden, modernen und übersichtlichen Webseite.

Die Anwendung ist eine SPA. Beim ersten Aufruf ist auf der Webseite lediglich eine

---

<sup>1</sup>Weitere Informationen zum Framework: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>

Überschrift, die Buttons, um das Menüfenster für Einstellungen zu öffnen sowie ein leeres Diagramm zu sehen. Wird der Menü-Button geklickt, öffnet sich von der linken Seite ein Menü, das alle nötigen Einstellungen für die Simulation enthält. Unter dem Diagramm befinden sich einige Hinweise zum korrekten Einsatz der Anwendung. Diese sind in kleiner Schriftgröße und unauffälliger Schriftfarbe gestaltet, um nicht vom Diagramm abzulenken.

Nach dem Bestätigen der Einstellungen im Menü durch Knopfdruck wird das Verhalten des ausgewählten Algorithmus mit den eingegebenen Parametern berechnet. Die Berechnung erfolgt schrittweise, wobei jedes Intervall, in dem ein Prozess Rechenzeit erhält, als ein Schritt behandelt wird. Danach wird das Diagramm an die Größe der errechneten Daten angepasst und anschließend der erste Schritt in das Diagramm eingezeichnet. Auch werden Buttons aktiviert, mit denen schrittweise durch das Diagramm navigiert werden kann. Des Weiteren wird ein Button angezeigt, mit dem die einzelnen Schritte als Animation abgespielt werden können. Das Diagramm befindet sich in einem HTML-Bereich, in dem ein Element verwendet werden kann, das größer als die Bildschirmbreite ist. Dieses kann durch Scrollen nach rechts und links ganz betrachtet werden. Es wurde sich für dieses Format entschieden, da das Diagramm andernfalls an die Bildschirm- bzw. Fensterbreite angepasst werden würde. Dadurch würde das Diagramm gerade bei einer großen Anzahl von Schritten unübersichtlich und schwer lesbar werden.

Rechts neben dem Diagramm werden relevante Informationen zum Verständnis des Algorithmenverhaltens angezeigt. Diese sind die Nummer des aktuell angezeigten Schritts, sein Anfangs- und Endzeitpunkt sowie eine Auflistung aller Prozesse, die startbereit sind. Unter der Anzeige der Ready Queue befindet sich eine tabellarische Darstellung von Informationen, die für den aktuell ausgewählten Algorithmus relevant sind, zum Beispiel die errechnete dynamische Priorität, der Kontowert der einzelnen Prozesse und die verbleibende Laufzeit (siehe Abbildung 5.1). Die Werte in diesen Anzeigen werden immer zu Beginn des Zeitintervalls entnommen, bevor der nächste Prozess ausgewählt wird, um die Gründe für die Entscheidung besser zu verdeutlichen. So wäre es beispielsweise bei dem Algorithmus „Shortest Remaining Job First“ deutlich schwieriger, die Auswahl nachzuvollziehen, wenn die verbleibenden Rechenzeiten nach Ablauf der Zeitscheibe angezeigt werden würden. Die Schrift in der Zeile des letztendlich ausgewählten Prozesses wird fettgedruckt, um ihn visuell von den anderen Prozessen zu unterscheiden.

**Aktueller Schritt:** 8  
**Aktuelles Intervall:** 700 -  
800

| ReadyQueue |          |
|------------|----------|
| P4         | P2 P5 P1 |

| Processdata |                   |
|-------------|-------------------|
| Name        | Verbleibende Zeit |
| P1          | 1050              |
| P2          | 900               |
| P3          | 1700              |
| <b>P4</b>   | <b>100</b>        |
| P5          | 6000              |

Abbildung 5.1: Tabellarische Informationsanzeige neben dem Diagramm

Es ist möglich, das aktuell abgebildete Diagramm per Knopfdruck als Bilddatei herunterzuladen. Zudem wird der aktuelle Stand der Anwendung gespeichert und kann bei Schließen und erneutem Öffnen des Browsers wiederhergestellt werden.

## 5.1 Dynamisches Menü

Das Menü, in dem die Parameter für die Simulation verändert werden können, ist im Normalfall ausgeblendet, um die Aufmerksamkeit des Nutzers nicht vom Diagramm abzulenken. Außerdem wird eine Vielzahl von Eingabefeldern benötigt, die sonst die Anwendung unübersichtlich erscheinen lassen würden. Aus diesem Grund wird das Menü auf Knopfdruck von der linken Bildschirmseite eingeblendet. Das Menü kann über einen „X“-Button in der rechten oberen Ecke wieder geschlossen werden (siehe Abbildung 5.2). Der Button wurde dort platziert, da dem Nutzer diese Platzierung bereits bekannt ist. Wird das Menü geschlossen und wieder geöffnet, bleiben die Eingaben in den Feldern erhalten.

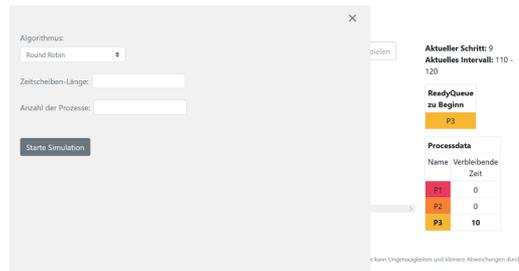


Abbildung 5.2: Menü im Ausgangszustand

Der Inhalt des Menüs verändert sich abhängig von den getätigten Eingaben, so dass nur relevante Eingabefelder angezeigt werden.

Zunächst kann der gewünschte Algorithmus ausgewählt werden. Dabei wird in einem Dropdown-Menü zwischen „Round Robin“, „Shortest Remaining Job First“, „Kontosystem“, „Linuxscheduler“ und „Custom“ unterschieden. Das Dropdown-Menü stellt sicher, dass immer ein Algorithmus als Standard ausgewählt wird, da der Rest des Menüs von dieser Eingabe abhängig ist. Zu Beginn ist „Round Robin“ ausgewählt. Abhängig davon welcher Algorithmus ausgewählt wird, werden irrelevante Eingabefelder ausgeblendet und andere Eingabefelder angezeigt.

Wird „Custom“ als Option ausgewählt, erscheint ein Eingabefeld, in dem eine Formel zur Errechnung der Priorität eingegeben werden muss. Darüber befindet sich eine Liste der zulässigen Zeichen mit Erläuterungen, welche Buchstaben stellvertretend für welche Werte eingesetzt werden können. Das Eingabefeld enthält eine Beispielformel, um die Verwendung zu verdeutlichen. Darunter befinden sich zwei Checkboxen, die mit „Kontowert berechnen“ und „Dynamische Zeitscheibenlänge berechnen“ beschriftet sind. Wird eine der Checkboxen angeklickt, erscheint unter der ausgewählten Option ein weiteres Eingabefeld um die entsprechende Formel einzugeben (siehe Abbildung 5.3).

Algorithmus: Custom

Zulässige Zeichen: ABDKLRST 0123456789 + - / \* ( )

A = Aktuelle Zeit  
 B = Basispriorität des Prozesses  
 D = Dynamische Priorität des Prozesses (beginnt bei 0 und wird jeden Schritt neu berechnet)  
 K = Kontowert des Prozesses (beginnt bei 0 und wird jeden Schritt neu berechnet)  
 L = Durchschnittliche Schlafzeit des Prozesses  
 R = Verbleibende Laufzeit des Prozesses  
 S = Startzeit des Prozesses  
 T = Bisherige Laufzeit des Prozesses

Prioritäts-Formel: (R/4)+5

Kontowert berechnen

Dynamische Zeitscheibenlänge berechnen

Abbildung 5.3: Custom-Eingabeoptionen im Menü

Bei den Formeln wird überprüft, ob nur gültige Zeichen verwendet werden, oder die Eingabe leer ist. Zudem wird bei der Verwendung von „K“ (für Kontowert) geprüft, ob die Option zur Berechnung des Kontowertes aktiviert ist. Auch werden mehrere mögliche Eingabefehler ausgeschlossen, um möglichst viele Fehler bei der Verwendung von selbstgeschriebenen Formeln zu vermeiden. So werden Eingaben, die nur Sonderzeichen enthalten oder in denen die Anzahl von offenen und geschlossenen Klammern unterschiedlich sind, sofort als fehlerhaft markiert. Auch werden Eingaben verhindert, in denen Buchstaben oder Zahlen ohne mathematische Operatoren direkt aufeinander folgen. Diese könnten beim Einsetzen der tatsächlichen Werte falsche Ergebnisse erzeugen. So würde beispielsweise bei einer Eingabe von zwei Buchstaben hintereinander, deren Gegenwert der Zahl „100“ entspricht, nach dem Einsetzen der Wert „100100“ erzeugt werden. Wird eine Eingabe als fehlerhaft markiert, kann die Berechnung des Verhaltens nicht gestartet werden, bevor sie korrigiert wurde.

Wird ein prioritätsbasierter Algorithmus, also „Round Robin“, „Kontosystem“ oder „Custom“ (mit nicht-aktivierter Berechnung von dynamischer Zeitscheibenlänge), ausgewählt, kann darunter die Länge der Zeitscheiben angegeben werden. In diesem Eingabefeld werden nur positive Zahlen akzeptiert. Leere Eingaben oder Eingaben, die Buchstaben oder sonstige Sonderzeichen enthalten, werden als fehlerhaft markiert. Wird ein anderer Algorithmus ausgewählt, wird dieses Feld entfernt (siehe Abbildung 5.4).

Algorithmus:  
 Shortest Remaining Job First ▾  
 Anzahl der Prozesse:   
 Starte Simulation

Abbildung 5.4: Menü ohne Eingabefeld für Zeitscheibenlänge

Anschließend kann der Nutzer die gewünschte Anzahl der Prozesse in der Simulation angeben. Dieses Eingabefeld akzeptiert nur positive Zahlen zwischen 1 und 10, da die Simulation auf maximal zehn Prozesse begrenzt ist. Auf diese Entscheidung wird in Abschnitt 5.3.2. weiter eingegangen. Sobald der Nutzer die „Enter“-Taste drückt oder mit der Maus in irgendeinen anderen Bereich der Seite klickt, erscheint für jeden Prozess eine Eingabezeile. Diese enthält Felder für Gesamtlauzeit und Startzeitpunkt des Prozesses (siehe Abbildung 5.5).

Anzahl der Prozesse: 3

|           |              |                      |        |                      |
|-----------|--------------|----------------------|--------|----------------------|
| PROZESS 1 | Gesamtdauer: | <input type="text"/> | Start: | <input type="text"/> |
| PROZESS 2 | Gesamtdauer: | <input type="text"/> | Start: | <input type="text"/> |
| PROZESS 3 | Gesamtdauer: | <input type="text"/> | Start: | <input type="text"/> |

Abbildung 5.5: Eingabezeile im Menü ohne Prioritätsfeld

Wurde ein prioritätsbasierter Algorithmus ausgewählt, ist auch ein Feld zur Eingabe der Basispriorität vorhanden (siehe Abbildung 5.6).

Anzahl der Prozesse: 3

|           |              |                      |        |                      |            |                      |
|-----------|--------------|----------------------|--------|----------------------|------------|----------------------|
| PROZESS 1 | Gesamtdauer: | <input type="text"/> | Start: | <input type="text"/> | Priorität: | <input type="text"/> |
| PROZESS 2 | Gesamtdauer: | <input type="text"/> | Start: | <input type="text"/> | Priorität: | <input type="text"/> |
| PROZESS 3 | Gesamtdauer: | <input type="text"/> | Start: | <input type="text"/> | Priorität: | <input type="text"/> |

Abbildung 5.6: Eingabezeile im Menü mit Prioritätsfeld

Für diese Eingaben wurde ein Format gewählt, in dem die Felder nicht freistehen sondern innerhalb einer Tabellenzeile zusammengefasst sind. Dadurch werden die Zeilen optisch eindeutig voneinander getrennt, und es ist für den Nutzer einfacher, sie beim Ausfüllen voneinander zu unterscheiden.

Mit dem Button „Simulation starten“ wird die Berechnung der einzelnen Schritte der Simulation des Algorithmusverhaltens gestartet. Der Button wird deaktiviert und eine Nachricht angezeigt, die den Nutzer informiert, dass die Rechnung durchgeführt wird (siehe Abbildung 5.7). Dies dient der Vermeidung von Neueingaben oder wiederholtem Klicken, da die Berechnung von Simulationen mit besonders vielen Schritten oder bei einem langsamen Rechner oder Browser einige Sekunden dauern kann. Danach schließt sich das Menü automatisch und wird auf seinen Ursprungszustand zurückgesetzt.

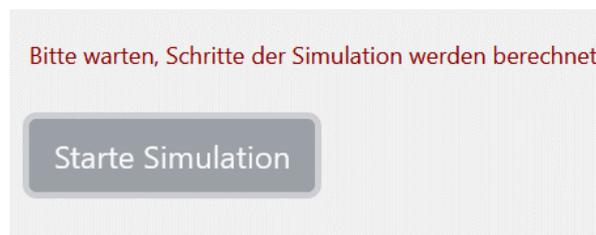


Abbildung 5.7: Warteanzeige im Menü

Der Input in allen Eingabefeldern wird bei Eintreffen des HTML-Events „onChange“ überprüft. Dieses Event tritt ein, wenn der Nutzer in einem Eingabefeld die Enter-Taste drückt oder mit der Maus auf einen anderen Bereich der Webseite klickt. Dabei wird die Eingabe auf Korrektheit geprüft und etwaige von der Eingabe abhängige andere Optionen im Menü angepasst, ausgeblendet oder eingeblendet. Sicherheitshalber werden alle Eingaben auch bei Klick auf den Bestätigungsbutton erneut geprüft. Dies dient der Vermeidung von Fehlern. Würde direkt nach einer fehlerhaften Eingabe der Bestätigungsbutton gedrückt, würde erst der Knopfdruck das „onchange“-Event im Browser auslösen, das die Überprüfung startet. So könnte die Berechnung beginnen, ohne dass die letzten ungültigen Eingaben erfasst werden würden, was zu Fehlern oder dem Abstürzen des Systems führen kann. Ist eine Eingabe ungültig, wird unter oder neben dem entsprechenden Eingabefeld eine Nachricht angezeigt, die den Fehler erklärt (siehe Abbildung 5.8). Wird eine Eingabe als fehlerhaft markiert, wird die Berechnung abgebrochen und der Startbutton wieder aktiviert.

Zeitscheiben-Länge:  Eingabe enthält ungültige Zeichen

Abbildung 5.8: Fehleranzeige im Menü

## 5.2 Berechnung des Algorithmus-Verhaltens

Nach der Überprüfung der eingegebenen Parameter auf Gültigkeit beginnt die Berechnung des Verhaltens der einzelnen Algorithmen zum Scheduling von Prozessen. Zuerst werden die eingegebenen Daten der Prozesse in das Array `processdata` eingelesen. Dazu wird für jeden Prozess ein Objekt angelegt, das die folgenden Attribute enthält:

- **name:** Name des Prozesses, bestehend aus dem String „Prozess “ und der entsprechenden Nummer
- **number:** eindeutige ID zur Identifizierung des Prozesses, die für spätere Berechnungen wie Zeile im Diagramm und Zuordnung der Farbe im Diagramm verwendet wird
- **start:** Startzeitpunkt des Prozesses
- **totaltime:** Gesamtlaufzeit des Prozesses
- **remainingtime:** Noch verbleibende Laufzeit des Prozesses. Dieses Attribut enthält zu Beginn den gleichen Wert wie `totaltime`.

Wird ein Algorithmus ausgewählt, der prioritätsbasiert ist, also „Kontosystem“, „Linux-scheduler“ oder „Custom“, werden auch folgende Attribute verwendet:

- **startprio:** Basispriorität des Prozesses
- **currentprio:** Aktuelle Priorität des Prozesses. Dieses Attribut enthält zu Beginn den Wert 0.

Bei Wahl des Algorithmus „Kontosystem“ oder aktivierter Option „Kontowert berechnen“ des Algorithmus „Custom“ wird ein Attribut verwendet, in dem der aktuelle Kontowert gespeichert wird:

- **account:** Aktueller „Kontostand“ für die Berechnung des kontobasierten Prioritätswerts (siehe Abschnitt 4.3). Dieses Attribut wird mit dem Wert 0 initialisiert.

Wird der „Linuxscheduler“-Algorithmus oder die Option „Dynamische Zeitscheibenlänge berechnen“ des Algorithmus „Custom“ ausgewählt, wird auch dafür ein Attribut initialisiert.

- **quantum**: Dynamisch berechnete Zeitscheibenlänge (siehe Abschnitt 4.4). Dieses Attribut erhält den Startwert 0.

Des Weiteren wird ein leeres Array `steps` erzeugt, in dem später die einzelnen Schritte der Simulation gespeichert werden.

Die Berechnung der einzelnen Schritte der Simulation erfolgt in einer einzigen Funktion, `calcZeitscheibe(algotype)`, die den Namen des Algorithmus als String erhält. Die Anwendung wurde an dieser Stelle so entworfen, da die Funktionsweisen der Zeitscheibenbasierten Algorithmen elementar gleich sind und nur die Kriterien zur Auswahl des nächsten Prozesses und die Bestimmung der Zeitscheibenlänge sich unterscheiden. Dabei macht es in der Funktionalität keinen größeren Unterschied, ob die Länge der Zeitscheibe dynamisch in jedem Schritt für jeden Prozess neu berechnet wird, oder ob sie von Beginn an statisch festgelegt ist. So wird die Zeitscheibe für die Algorithmen, bei denen sie statisch ist, wie bei „Round Robin“ oder dem Linuxscheduler, zu Beginn festgelegt oder berechnet und in den anderen Fällen bei jedem Prozesswechsel neu berechnet.

Das Vorgehen der Berechnung wird im folgenden erklärt. Zum besseren Verständnis wurde der Programmablauf auch als vereinfachter Pseudocode abgebildet (siehe Abbildung 5.9).

```

startTime = erhalteStartzeit
zeitscheibenlängeEntsprechendAlgorithmusAuslesenOderBerechnen
readyQueue = prozesseBeiStartHinzufügen
processesLeft = processdata.length
while processesLeft > 0:
    next = ermittleNächstenProzess
    if next.remainingtime > zeitscheibe:
        endTime = startTime + zeitscheibe
        next.remainingtime -= zeitscheibe
    else:
        endTime = startTime + next.remainingtime
        next.remainingtime = 0
        processesLeft --
    neuenSchrittErzeugenUndZuStepsHinzufügen
    processdataAktualisieren
    startTime = endTime
    if RoundRobinAusgewählt:
        processdataNachStartzeitSortieren
    fügeInzwischenGestarteteProzesseZuReadyQueueHinzu
    if next.remainingtime > 0:
        nextZuReadyQueueHinzufügen
    if RoundRobinAusgewählt:
        processdataWiederNachEingabereihenfolgeSortieren
    if readyQueueLeer and spätererProzessVorhanden:
        startTime = nächstenStartzeitpunktFinden

```

Abbildung 5.9: Pseudocode der Berechnung

Zunächst wird der Startzeitpunkt der Simulation ermittelt und in dem Wert `startTime` gespeichert. Dieser gibt immer die aktuelle Zeit in der Berechnung der Simulation an und wird am Ende jedes Schritts aktualisiert. Anschließend werden alle Prozess-Objekte, die zu diesem Zeitpunkt beginnen, zum Array `readyQueue` hinzugefügt. In diesem befinden sich zu jedem Zeitpunkt in der Simulation alle Prozesse, die momentan auf die Zuteilung von Rechenzeit warten. Danach wird die Länge der Zeitscheibe entsprechend des ausgewählten Algorithmus der Zeitscheibe ermittelt. Die folgende Berechnung von Schritten wird solange wiederholt, bis keine Prozesse mit Laufzeit mehr verbleiben.

Zunächst wird der aktuelle Inhalt des Arrays `readyQueue` in HTML übertragen. Anschließend wird anhand von algorithmusabhängigen Kriterien der nächste Prozess ausgewählt, der Rechenzeit erhalten soll und aus dem Array `readyQueue` entfernt. Das Auswahlvorgehen der einzelnen Algorithmen wird in den folgenden Abschnitten genauer erläutert. Nachdem der nächste Prozess ausgewählt wurde, wird der Zustand des Arrays `processdata` in HTML umgewandelt. Dies geschieht nach der Auswahl des nächsten Prozesses, damit dieser in der Tabelle optisch hervorgehoben werden kann.

Nachdem der nächste Prozess, der Rechenzeit erhält, ausgewählt wurde, wird die Endzeit des aktuellen Simulationsschritts bestimmt und die verbleibende Zeit des ausgewählten

Prozesses aktualisiert. Hat ein Prozess keine verbleibende Rechenzeit, wird die Anzahl der verbleibenden Prozesse reduziert.

Anschließend wird aus den im vorherigen Schritt ermittelten Informationen ein Daten-Objekt generiert, das später in das Diagramm eingetragen werden kann und dem Array `steps` hinzugefügt wird. Dieses enthält Start- und Endzeitpunkt des Intervalls, in dem der Prozess Zugriff auf die CPU hat, die Indexnummer des Prozesses sowie Zustand der Arrays `readyQueue` und `processdata` vor dem Auswahlprozess in HTML-Form für die Anzeige neben dem Diagramm.

Danach wird die Endzeit des aktuellen Schritts zur neuen Startzeit. Anschließend wird überprüft, ob in der Zwischenzeit weitere Prozesse gestartet sind. Ist dies der Fall, werden sie zum Array `readyQueue` hinzugefügt. Hat der ausgewählte Prozess noch verbleibende Rechenzeit wird er ebenfalls hinzugefügt. Ist das Array `readyQueue` an diesem Punkt leer wird überprüft, ob Prozesse mit einem späteren Startzeitpunkt existieren. Ist dies der Fall, wird der nächste Startzeitpunkt zum neuen aktuellen Zeitpunkt. So wird vermieden, dass die Simulation vorzeitig endet, wenn das Array `readyQueue` leer ist, aber noch Prozesse verbleiben, die erst zu einem späteren Zeitpunkt beginnen (gemeint ist damit ein Szenario wie in Abbildung 5.10 zu sehen ist).

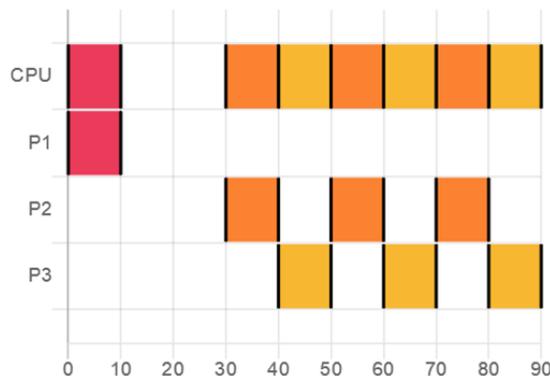


Abbildung 5.10: Diagramm mit Lücke bei später startenden Prozessen

Ist die Berechnung aller Schritte der Simulation abgeschlossen, wird das Diagramm an die neuen Werte angepasst und die Werte aus dem Array `steps` eingezeichnet. Dies wird in Abschnitt 5.3. näher erläutert. Alternativ zu der Berechnung aller Schritte zu Beginn wäre es auch möglich gewesen, die Schritte beim Navigieren durch das Diagramm einzeln zu berechnen, um lange Wartezeiten bei komplizierten Berechnungen zu vermeiden. Allerdings wäre es dabei nicht auszuschließen gewesen, dass es beim Zurückgehen zum vorherigen Schritt zu Rundungsfehlern und abweichenden Ergebnissen kommt.

Im Folgenden wird genauer darauf eingegangen, wie in der Implementierung der einzelnen Algorithmen der nächste Prozess ausgewählt wird und welche Besonderheiten durch die Implementierung erzeugt werden.

### 5.2.1 Round Robin

Bei der Implementierung dieses Algorithmus zum Scheduling wird der nächste Prozess ausgewählt, indem das erste Objekt aus der Warteschlange entnommen wird. Da neue Objekte immer am Ende des Arrays hinzugefügt werden, wird so immer das am längsten in der Warteschlange vorhandene Objekt ausgewählt. Für diesen Algorithmus werden nur die von JavaScript bereitgestellten Array-Funktionen genutzt, da keine besonderen Auswahlkriterien existieren.

Prozesse mit verbleibender Laufzeit werden immer am Ende ihrer Zeitscheibe wieder am Ende der Warteschlange hinzugefügt (vgl. Abschnitt 4.1). In dieser Implementierung werden neue Prozesse, die seit Beginn der letzten Zeitscheibe begonnen haben, ebenfalls immer am Ende der neuen Zeitscheibe der Ready Queue hinzugefügt. Aus diesem Grund wurde die Reihenfolge im Programmablauf so gewählt, dass zunächst alle Prozesse der Ready Queue hinzugefügt werden, die zwischen Beginn und Ende der aktuellen Zeitscheibe begonnen haben. Um die korrekte Reihenfolge zu gewährleisten, werden die Prozesse aufsteigend nach ihrem Startzeitpunkt sortiert. Fällt der Startzeitpunkt eines Prozesses exakt auf das Ende einer Zeitscheibe, wird dieser Prozess ebenfalls an diesem Zeitpunkt hinzugefügt. Erst danach wird der aktuell ausgewählte Prozess an das Ende der Ready Queue angehängt. Dadurch wird die korrekte Reihenfolge der Prozesse in der Warteschlange gewährleistet. Die exakte Reihenfolge der Vorgänge wenn das Ende der Zeitscheibe und der Beginn eines Prozesses exakt zusammentreffen wird in den Beispielen in der Fachliteratur nicht spezifiziert. In dieser Implementierung traf ich die Entscheidung die neuen Prozesse zuerst in die Warteschlange einzureihen, da der aktuelle Prozess bereits Rechenzeit erhalten hat. Haben zwei Prozesse den gleichen Startzeitpunkt, so werden sie in Eingabereihenfolge der Ready Queue hinzugefügt.

### 5.2.2 Shortest Remaining Job First

Dieser Scheduling-Algorithmus ist so implementiert, dass zur Auswahl des nächsten Prozesses zunächst alle Prozesse im Array `readyQueue` durchlaufen werden, um durch Vergleich die kürzeste, verbleibende Zeit zu ermitteln. Anschließend wird überprüft, ob Prozesse existieren, deren Startzeitpunkt noch nicht eingetroffen ist. Dies wird ermittelt,

indem die Anzahl der Elemente im Array `readyQueue` mit der Menge der verbleibenden Prozesse, gespeichert in der Variable `processesLeft`, verglichen wird. Sind diese nicht identisch, wird der nächste Startzeitpunkt mit der Funktion `findNextStart(startTime)` ermittelt. Der Wert der Variable `zeitscheibe` wird ermittelt, in dem die aktuelle Zeit vom nächsten Prozessstartzeitpunkt abgezogen wird. Andernfalls wird die Länge der Zeitscheibe als die verbleibende Laufzeit des Prozesses gesetzt. Durch die Anpassung der Zeitscheibenlänge in jedem Schritt der Simulation kann nach der Auswahl des nächsten Prozesses das gleiche Verfahren zur Erstellung eines Datenobjekts verwendet werden, ohne in einer separaten Funktion mit ähnlichem Ablauf den Endzeitpunkt für die einzelnen Simulationsschritte anders zu bestimmen. Dies schien nach mehreren Programmierexperimenten sinnvoller, als für jeden Algorithmus eine eigene Funktion mit sehr ähnlichem Vorgehen zu schreiben.

Befinden sich zwei Prozesse mit der gleichen verbleibenden Laufzeit in der Warteschlange, wird der Prozess ausgewählt, der einen geringeren Index hat, also sich bereits länger in der Ready Queue aufhält.

### 5.2.3 Kontosystem

Beim Auswahlverfahren in der Implementierung des „Kontosystem“-Algorithmus werden zunächst Kontowert und dynamische Priorität für jeden Prozess nach der in Abschnitt 4.3. erklärten Formel berechnet. Danach werden die Informationen für die einzelnen Prozess-Objekte im Array `readyQueue` aktualisiert. Anschließend wird der Prozess mit der geringsten Priorität ausgewählt. Dazu wird die Funktion `findLowestPriority(readyQueue)` verwendet. Existieren zwei Prozesse, die die gleiche Priorität haben, wird der Prozess ausgewählt, der an früherer Stelle im Array `readyQueue` steht, sich also schon längere Zeit in der Warteschlange befindet. Dies ist bei allen implementierten prioritätsbasierten Algorithmen der Fall.

### 5.2.4 Linuxscheduler

In der Anwendung wurde auch eine vereinfachte Version des Schedulers, der in Version 2.6 des Linux-Kerns Verwendung findet, implementiert. Deren Funktionsweise wurde in Abschnitt 4.4 näher erläutert. Beim Auswahlverfahren des nächsten Prozesses werden zunächst für jeden Prozess die dafür relevanten Werte berechnet. Danach wird die durchschnittliche Schlafzeit oder Wartezeit des entsprechenden Prozesses pro Simulationsschritt ermittelt. Dafür wird die Funktion `calculateAvgsleep(process, startTime)`

verwendet. Anschließend wird aus diesem Wert, wie im vorherigen Kapitel erklärt, der Bonus mit der Funktion `calculateBonus(avgsleep)` berechnet. Danach wird die dynamische Priorität kalkuliert und diese Information im Array `readyQueue` aktualisiert. Anschließend wird aus der Warteschlange der Prozess mit der geringsten Priorität entnommen. Die Länge der Zeitscheibe in der Variable `zeitscheibe` erhält den Wert `quantum` aus dem Prozessobjekt, der zu Anfang der Simulation für jeden Prozess berechnet wurde.

## 5.2.5 Custom

Bei der Option einen eigenen Algorithmus zu konstruieren, kann der Nutzer im Menü Formeln für die Berechnung von dynamischer Priorität, Kontowert und Zeitscheibenlänge eingeben. Dabei stehen dem Nutzer zusätzlich zu Klammern und den mathematischen Operatoren für Addition, Subtraktion, Multiplikation und Division folgende Werte zur Verfügung:

- A: der aktuelle Zeitpunkt in der Simulation
- R: die verbleibende Laufzeit des Prozesses
- T: die Gesamtlaufzeit des Prozesses
- S: der Startzeitpunkt des Prozesses
- B: die Basispriorität des Prozesses
- L: die durchschnittliche Schlafzeit des Prozesses bisher
- K: der errechnete Kontowert

Des Weiteren kann die dynamische Priorität `D` verwendet werden, allerdings nur in der Berechnung der dynamischen Priorität selbst und der Berechnung des Kontowertes. Der Grund dafür ist, dass der Kontowert meistens für die Berechnung der Priorität genutzt wird. Aus diesem Grund muss er ermittelt werden, bevor die Priorität berechnet wird, damit der Wert in die Prioritätsformel eingesetzt werden kann. Würde er im Kontowert verwendet werden, würde stets der vorherige Prioritätswert benutzt werden, was andere Ergebnisse als vom Nutzer gewünscht oder erwartet erzeugen könnte.

In der Implementierung des Auswahlverfahrens wird zunächst für jeden Prozess, der sich im Array `readyQueue` befindet, überprüft, ob die Kontowert-Berechnung im Menü aktiviert wurde. Ist dies der Fall, wird der String aus dem Menüfeld, in das die Formel

eingegeben wird, abgerufen und der Funktion `replaceGeneral(equation, process, startTime)` übergeben. Diese ersetzt die oben genannten Buchstaben durch die aktuellen Werte in der Simulation und gibt den geänderten String zurück. Dieser wird dann mit Hilfe der Funktion „evaluate“ aus der Bibliothek „math.js“<sup>2</sup> geparsed, so der Kontowert ausgerechnet und im Prozess-Objekt im Array `readyQueue` aktualisiert. Anschließend wird der Vorgang für die dynamische Priorität wiederholt. Danach wird überprüft, ob die Berechnung der Zeitscheibenlänge aktiviert wurde. Ist dies der Fall, werden auch in dieser Formel die Buchstaben ersetzt, der Wert errechnet und das Prozess-Objekt aktualisiert. Es wurde in der Implementierung die Entscheidung getroffen, die Zeitscheibenlänge jedes Prozesses in jedem Schritt zu berechnen, im Gegensatz zum Linuxscheduler, in dem dies nur einmal am Anfang nötig ist. Der Grund dafür ist, dass in der Formel auch Werte verwendet werden können, die sich in jedem Schritt der Simulation verändern. So können mit entsprechenden Formeln auch Zeitscheibenlängen erzeugt werden, die sich im Verlauf der Simulation verändern (ein Beispiel dafür ist in Abbildung 5.11 zu sehen). Dafür ist allerdings die wiederholte Berechnung notwendig.

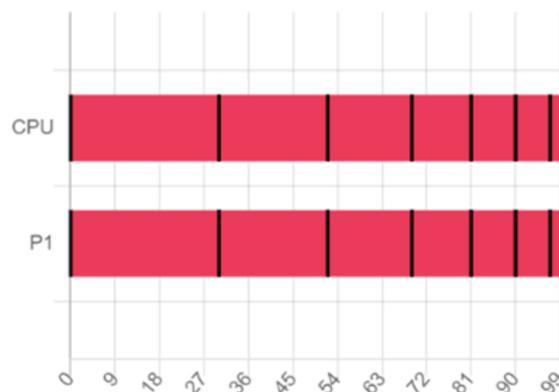


Abbildung 5.11: Diagramm mit dynamischer Zeitscheibenlänge

Anschließend wird auch die Information im Array `processdata` aktualisiert. Danach wird mit der Funktion `findLowestPriority(readyQueue)` der Prozess mit der geringsten Priorität ausgewählt. Sofern die Berechnung der Zeitscheibenlänge aktiviert wurde, wird der Wert des Attributs `quantum` des gewählten Prozesses in die Variable `zeitscheibe` übertragen.

<sup>2</sup><https://mathjs.org/docs/reference/functions/evaluate.html>

## 5.3 Diagramm

Es existieren eine Vielzahl von Bibliotheken für JavaScript, die das Zeichnen von Diagrammen verschiedener Art unterstützen. Zunächst musste eine Bibliothek ausgewählt werden, die es ermöglicht, ein simples Gantt-Diagramm zu erstellen, das das Scheduling-Verhalten der verschiedenen Algorithmen sinnvoll und visuell ansprechend abbildet. Das Aussehen des Diagramms ist wie folgend konzipiert:

In der ersten Zeile sollte die Gesamtverwendung der CPU, eingeteilt in Zeitscheiben, abgebildet werden. Dabei sollten die verschiedenen Prozesse als verschiedenfarbige, aneinandergereihte Blöcke dargestellt werden. Für Zeiträume, in denen die CPU von keinem Prozess verwendet wird, sollte das Diagramm einen leeren Zwischenraum enthalten. Darunter sollte für jeden Prozess eine eigene Zeile existieren, in der nur die Blöcke abgebildet werden, in denen der Prozess die CPU verwendet. Die Zwischenräume zwischen diesen Blöcken sollten ebenfalls leer sein.

Bei der Recherche und ersten Programmierexperimenten wurde schnell klar, dass viele der Bibliotheken, die Gantt-Diagramme unterstützen, nicht für die geplante Anwendung geeignet sind. Viele dieser Bibliotheken unterstützen ausschließlich Projektmanagement und erzeugen sehr detaillierte Gantt-Diagramme mit Kalenderdaten, Aufteilung von Teammitgliedern und genauer Aufzeichnung der Reihenfolge der einzelnen Schritte. Die Bibliotheken „DHTMLXGantt“<sup>3</sup>, „AnyChart“<sup>4</sup>, „jQuery.Gantt“<sup>5</sup>, „jsgantt-improved“<sup>6</sup> wurden aus diesem Grund nach Recherche ausgeschlossen. Die Bibliothek „jscharting“<sup>7</sup>, deren Gantt-Diagramme vereinfacht werden konnten, setzen für die Verwendung den Erwerb einer Lizenz voraus. Damit war sie für den Einsatz in dieser Anwendung nicht geeignet.

Im nächsten Schritt experimentierte ich mit verschiedenen anderen Diagramm-Arten in verschiedenen Bibliotheken, um die gewünschte Ansicht zu erzeugen. Zunächst konzentrierte ich mich auf das Zeichnen von horizontal ausgerichteten Balkendiagrammen. Dabei war meine Intention, jede Zeitscheibe als einzelnen Balken einzuzichnen. Ausgeschlossen wurden die Bibliotheken „Highcharts“<sup>8</sup>, „amcharts“<sup>9</sup>. Diese beinhalteten zwar vielversprechende Diagrammtypen, benötigen zur Verwendung allerdings eben-

---

<sup>3</sup><https://dhtmlx.com/docs/products/dhtmlxGantt/>

<sup>4</sup>[https://docs.anychart.com/Gantt\\_Chart/Overview](https://docs.anychart.com/Gantt_Chart/Overview)

<sup>5</sup><http://taitem.github.io/jquery.Gantt/>

<sup>6</sup><https://jsganttimproved.github.io/jsgantt-improved/>

<sup>7</sup><https://jscharting.com/examples/chart-types/gantt/>

<sup>8</sup><https://www.highcharts.com/blog/products/highcharts/>

<sup>9</sup><https://www.amcharts.com/javascript-charts/>

falls eine kostenpflichtige Lizenz. Daher wurden im Folgenden nur Bibliotheken weiter berücksichtigt, deren Lizenzen eine den Anforderungen entsprechende kostenfreie Verwendung ermöglichen.

Viele der existierenden Bibliotheken unterstützen das Zeichnen von horizontalen Balkendiagrammen, die mehrere, aneinandergereihte Blöcke enthalten. Allerdings konnte ich nur eine Bibliothek ermitteln, mit der es möglich ist, leere Flächen in Balken-Diagramme einzufügen. Zunächst experimentierte ich mit „ToastUI“<sup>10</sup>, einer Bibliothek, die verschiedene Arten von Balkendiagrammen beinhaltet. In dieser Bibliothek konnten allerdings keine leeren Blöcke im Diagramm platziert werden. Anschließend versuchte ich die Bibliothek „chartist.js“<sup>11</sup> zu verwenden; allerdings erzeugte bereits der in der Dokumentation vorhandene Beispielcode Anzeigefehler, deren Ursache ich auch durch ausführliche Recherche in der Dokumentation nicht feststellen konnte.

Danach wurde die Bibliothek „apexcharts“<sup>12</sup> evaluiert. Die Balkendiagramme schienen zunächst vielversprechend, da die Bibliothek die Verwendung von Kategorien als Skala an den Achsen und eine horizontale Ausrichtung ermöglichte. Allerdings konnten verschiedenfarbige Blöcke nur versetzt in den Zeilen dargestellt werden (siehe Abbildung 5.12), was nicht dem Konzept entsprach.



Abbildung 5.12: Horizontal ausgerichtete „Range Column Chart“ mit „apexcharts“ gezeichnet

Daher entschied ich mich, im Weiteren mit „chart.js“<sup>13</sup> zu arbeiten – einer OpenSource Bibliothek, die das Zeichnen einer Vielzahl von Diagramm-Arten ermöglicht. Auch unterstützt sie die Verwendung von Kategorien statt einer numerischen Skala an den Achsen, einer Funktionalität, die bei mehreren anderen Bibliotheken zwar laut Dokumentation vorhanden ist, aber häufig für Anzeigeprobleme sorgt oder in einigen Diagrammtypen nicht anwendbar ist. Zudem unterstützt „chart.js“ die Verwendung eines Tooltips, mit dem bei Berühren eines Elements mit dem Mauszeiger relevante Infor-

<sup>10</sup><https://ui.toast.com/>

<sup>11</sup><http://gionkunz.github.io/chartist-js/>

<sup>12</sup><https://apexcharts.com/>

<sup>13</sup><https://www.chartjs.org/>

mationen angezeigt werden können<sup>14</sup>. Weitere Experimente zeigten allerdings, dass die Verwendung einer „Stacked Bar Chart“ zwar das Zeichnen eines Diagramms mit dem gewünschten Aussehen ermöglicht, die Handhabung allerdings kompliziert ist. Bei dieser Art von Diagramm werden die Blöcke mit Koordinaten von Beginn und Ende des Balkens in das Diagramm eingezeichnet. Nach jedem Block wird jedoch das Koordinatensystem für den Folgeblock auf null zurückgesetzt. So hätte beispielsweise ein Block für die Zeitscheibe von 0–10 ms die x-Start- und Endkoordinaten 0 und 10. Der darauffolgende Block für eine Zeitscheibe von 20–30 ms hätte allerdings die x-Start- und Endkoordinaten 10 und 20 (siehe Abbildung 5.13 und 5.14). Diese Umrechnung von Koordinaten ist zwar umsetzbar, aber eine direkte Übertragung von Zeit auf die angegebenen Koordinaten wäre einfacher in der Verwendung.

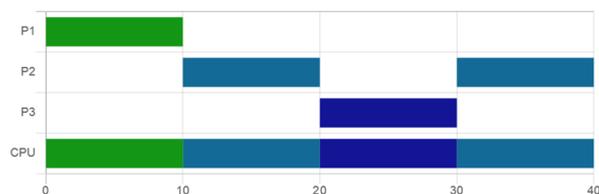


Abbildung 5.13: Erster Entwurf mit Verwendung eines „Stacked Bar Chart“-Diagramms

```
datasets: [{
  label: 'P1, Priority 1',
  data: [[0,10], null, null, [0,10]],
  backgroundColor: '#139615'
},{
  label: 'P2, Priority 3',
  data: [null, [10,20], null, [0,10]],
  backgroundColor: '#136a96'
},{
  label: 'P3, Priority 5',
  data: [null, null, [20,30], [0,10]],
  backgroundColor: '#131596'
},{
  label: 'P2, Priority 4',
  data: [null, [10,20], null, [0,10]],
  backgroundColor: '#136a96'
}
]
```

Abbildung 5.14: Koordinaten-Format für „Stacked Bar Chart“

Außerdem werden bei dieser Art von Diagramm alle Blöcke, die sich in einer Zeile

<sup>14</sup><https://www.chartjs.org/docs/latest/charts/bar.html>

befinden, als zusammengehörig angesehen, so dass bei Verwendung des Tooltips Informationen über alle Blöcke in der Zeile angezeigt werden statt lediglich über den mit der Maus berührten Block. Dies macht die Verwendung des Tooltips zur Anzeige von Informationen wie der aktuellen Priorität oder verbleibender Ausführungszeit unübersichtlich und nicht sinnvoll.

Weitere Experimente mit den anderen, von „chart.js“ unterstützten Diagrammartarten führten zu dem in der finalen Implementierung verwendeten Diagramm, einem stark abgewandelten Linien-Diagramm. Im Folgenden wird zunächst auf die allgemeine Funktionsweise von Diagrammen in der Bibliothek „chart.js“ eingegangen, um anschließend die Abwandlungen zu erläutern, die am Liniendiagramm vorgenommen wurden, um das finale Aussehen (siehe Abbildung 5.15) zu erzeugen.

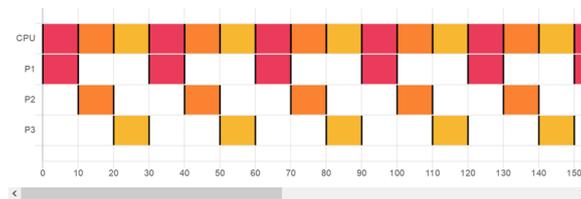


Abbildung 5.15: Finale Version des Diagramms

### 5.3.1 Diagramme in chart.js

Die Bibliothek 'chart.js' verwendet Elements des Typs HTML5-Canvas als Basis für die Grafiken. Ein Canvas ist ein Bestandteil von Webseiten, in dem mit JavaScript Text geschrieben und Grafiken gezeichnet werden können (vgl. MDN Contributors 2020).

Um ein neues Diagramm zu zeichnen, wird ein Objekt der Klasse Chart erstellt, dem das Ziel-Canvas-Element, die Art des Diagramms, die im Diagramm enthaltenen Daten sowie ausgewählte Optionen als JavaScript-Objekte übergeben werden.

Zu den Optionen zählen Einstellungen wie automatische Anpassung des Diagramms an die Browserfenstergröße, Seitenabstand zwischen Diagramm und Rand des Canvas-Elements, Animationen bei erstem Zeichnen des Diagramms und Hinzufügen von neuen Daten<sup>15</sup>. Auch die Beschriftung und Maßstäbe der Achsen und Anfangs- und Endpunkte werden hier definiert. Dabei können die Art der Achsen (unterschieden wird unter anderem zwischen numerisch, Zeitfortschritt und selbst festgelegten Kategorien) sowie die Platzierung der Beschriftungen und Platzierung und Aussehen des Rasters im Hin-

<sup>15</sup><https://www.chartjs.org/docs/latest/getting-started/usage.html>

tergrund des Diagramms festgelegt werden.<sup>16</sup>

Das Daten-Objekt enthält, je nach Diagramm-Typ, allgemeine Einstellungen für Farbe, Liniendicke und Form der Punkte sowie ein Array von Werten oder Objekte, die einen Wert und speziell auf diesen Wert bezogene Farbwerte etc. enthalten.<sup>17</sup>

### 5.3.2 Finale Implementierung des Diagramms

Die finale Implementierung ist ein Linien-Diagramm, das stark abgewandelt ist, um ein Gantt-Diagramm nachzuahmen<sup>18</sup>. Dabei werden die Kategorien „CPU“ und die Bezeichner/Namen der Prozesse als Skala auf der y-Achse und eine numerische Skala, die der verstrichenen Zeit entspricht, als Skala auf der x-Achse verwendet. Die Blöcke werden dargestellt, indem Start- und Anfangszeitpunkt des Blocks mit einer sehr dicken Linie verbunden werden. Anfangs- und Endpunkt werden mit schwarzen Strichen gekennzeichnet, um die Unterscheidung von nebeneinanderliegenden Blöcken gleicher Farbe zu erleichtern. Die Achsenbeschriftung und das Hintergrund-Raster des Diagramms sind auf der y-Achse so verschoben, dass sie sich nicht auf den Punkten selbst sondern dazwischen befinden. So erhält jede Kategorie eine eigene Zeile, in der die Blöcke platziert werden. Diese Basis-Einstellungen werden beim ersten Laden der Seite festgelegt. Wenn die Parameter im Menü geändert werden, werden die Größe des Diagramms und die Beschriftung der Achsen angepasst, die anderen Optionseinstellungen bleiben erhalten. Die verwendeten Farben sind 10 statische Farben, die als hexadezimale Farbwerte im Array „colors“ gespeichert sind (siehe Abbildung 5.16).



Abbildung 5.16: Verwendete Farbpalette

Die Anzahl der verwendbaren Prozesse ist auf zehn begrenzt, um die Einzigartigkeit und Unterscheidbarkeit der verwendeten Farben zu gewährleisten. So wurde zunächst eine unendliche Zahl von Farben eingesetzt, die entsprechend der Anzahl an Prozessen zwischen zwei Farbwerten interpoliert wurden. Bei einer größeren Anzahl von Prozessen war die Unterscheidung anhand der Farbe allerdings häufig schwierig (siehe Abbildung 5.17).

<sup>16</sup><https://www.chartjs.org/docs/latest/axes/>

<sup>17</sup><https://www.chartjs.org/docs/latest/configuration/>

<sup>18</sup><https://www.chartjs.org/docs/latest/charts/line.html>

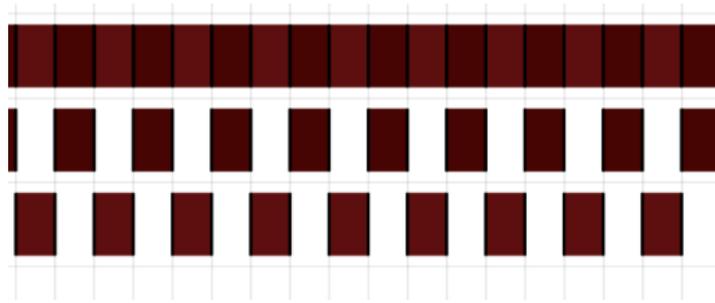


Abbildung 5.17: Diagramm, das Test-Farbpalette verwendet

Letztendlich entschied ich mich gegen die Verwendung eines Tooltips, um relevante Informationen zu den einzelnen Schritten anzuzeigen. Diese werden stattdessen in einer Anzeige rechts neben dem Diagramm abgebildet und aktualisiert. Der Grund für diese Entscheidung war, dass bei Verwendung des in „chart.js“ vorhandenen Tooltip dieses nicht auf Berührung des Mauszeigers mit dem bunt gefärbten Balken reagierte sondern auf Berührung der schwarzen Striche zu Beginn und Ende eines Balkens. Da diese sich gerade in der CPU-Zeile oft mit einem anderen Block überschneiden, war häufig schwer zu erkennen, zu welchem Block die angezeigten Informationen gehörten. Auch in mehreren Experimenten mit verschiedenen möglichen Einstellungen des Tooltip konnte ich dieses Problem nicht lösen. Zudem war die Anzeige von Informationen an der Seite des Diagramms deutlich übersichtlicher als die kleine, durch das Tooltip erzeugte Anzeige. So kann nicht nur die Priorität eines einzigen Blocks übersichtlich und gut lesbar angezeigt werden, sondern Informationen über alle Prozesse und die Ready Queue.

### 5.3.3 Anpassung des Diagramms an veränderte Parameter

Das Diagramm wird dynamisch an die im Menü eingegebenen Parameter angepasst und verändert. Dies geschieht in der Funktion `resizeChart(chart)`. Wird ein neu berechnetes Diagramm gezeichnet, wird zunächst das vorherige Diagramm komplett gelöscht, da es bei Veränderungen der Größe einer vorhandenen Darstellung zu Unschärfe und Artefakten sowie anderen Anzeigefehlern kommen kann. Danach werden die Beschriftung und Intervalle der Achsen für das neue Diagramm festgelegt. Wird ein Diagramm aus einer vorherigen Berechnung wiederhergestellt, werden die dafür nötigen Werte aus dem Browserpeicher abgerufen.

Die Raster-Intervalle der x-Achse entsprechen bei Algorithmen, die feste Zeitscheibenlängen verwenden, der eingegebenen Länge einer Zeitscheibe. Andernfalls wird mit Hilfe der Funktion `calcShortestInterval()` die kleinste Schrittlänge ermittelt und diese

für das Raster und die Skala auf der x-Achse verwendet. Dieser Wert wird dann im Browser gespeichert, um das Diagramm nach Schließen wiederherstellen zu können (siehe Abschnitt 5.3.6). Die Beschriftung der x-Achse geht vom Startzeitpunkt des ersten Schritts bis zum Endzeitpunkt des letzten Schritts. Dann werden die einzelnen Beschriftungen zur y-Achse hinzugefügt. Dabei wird mit einer leeren Kategorie begonnen um zu gewährleisten, dass die erste Zeile, die Blöcke enthält, vollständig dargestellt wird. Wird diese nicht eingetragen, würde die erste Zeile nur in der Hälfte ihrer vollständigen Höhe angezeigt werden. An zweiter Stelle wird die CPU-Zeile eingetragen. Anschließend werden der eingegebenen Anzahl von Prozessen entsprechend weitere Zeilen hinzugefügt. Zum Abschluss wird eine weitere leere Kategorie eingesetzt, damit auch die letzte Zeile komplett dargestellt wird. Diese Werte werden im bereits erwähnten Array `options` verändert.

Anschließend wird die Größe des Canvas-Elements, welches das Diagramm enthält, errechnet. Diese werden von Hand ermittelt und eingestellt statt die automatische Größenbestimmung und Anpassung der Bibliothek „charts.js“ zu verwenden, da diese Diagramme mit vielen Werten in der Breite häufig stark verkleinert und an die Bildschirmmaße angepasst. Dies führt oft dazu, dass die einzelnen Blöcke nicht mehr erkennbar sind. Um die nötige Breite zu errechnen, wird die Funktion `calcCanvasWidth(algotype)` verwendet. Dabei wird die Anzahl der errechneten Schritte mit 50 multipliziert. Die Ausnahme bildet dabei der Algorithmus „Shortest Remaining Job First“, da in diesem häufig nur eine geringe Anzahl an Schritten existiert und die Darstellung mit der sonst verwendeten Formel sehr gedrängt und unübersichtlich wird. In diesem Fall wird der Endzeitpunkt des letzten Schritts durch die kürzeste Schrittlänge geteilt und dieser Wert anschließend mit 50 multipliziert. Danach wird überprüft, ob der so errechnete Wert unter der Mindest- oder über der Maximalbreite liegt und ansonsten auf diese gesetzt wird. Die Mindestbreite ist lediglich der Mindestwert um ein lesbares Diagramm zu erzeugen. Die Maximalbreite des Canvas-Elements muss festgelegt werden, da eine Maximalgröße dieser Elemente existiert. Wird diese überschritten, wird das Diagramm nur bis zu diesem Punkt korrekt eingezeichnet (siehe Abbildung 5.18). Diese Maximalgröße ist sowohl von der Maximalbreite als auch von der größtmöglichen Anzahl Pixel abhängig.<sup>19</sup>

Die hier verwendeten Konstanten wurden durch Tests in den Browsern Firefox, Microsoft Edge und Google Chrome ermittelt bis eine übersichtliche Darstellung erreicht wurde, da auch Bildgrößen, die unter den in der Dokumentation genannten Maximalwerten lagen, das eben genannte Problem erzeugten. So wurden bei einer in der Anwendung einge-

---

<sup>19</sup><https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas>

stellten Maximalgröße von 15.000 Pixeln (die damit unter dem Maximalwert von 16.384 liegt) bei einem Download Bilddateien erzeugt, deren Breite weit über 15.000 lag und die nur bis zur Grenze von 16.384 korrekt gerendert wurden. Bei einer eingestellten Maximalgröße von 10.000 Pixeln wurde eine Bilddatei mit 15.000 Pixeln Breite erzeugt. Da dieser Fehler nur auf Laptops auftrat, die das Betriebssystem Windows 10 und als Standard einen Zoom von 150% verwenden und auf Desktopcomputern mit 100% Zoom Bilder mit 10.000 Pixeln erzeugt wurden, vermute ich den Fehler in dieser Einstellung. Diese Frage konnte allerdings auch nach ausführlicher Recherche nicht endgültig geklärt werden. Da auf den drei Laptops, die zum Testen verwendet wurden, eine empfohlene Skalierung von 150% verwendet wurde, entschied ich mich die Maximalgröße von 10.000 Pixeln Breite beizubehalten. Auch bei einer Breite von 10.000 Pixeln sind die Diagramme auch mit beispielsweise 800 Schritten noch gut lesbar, so dass diese Skalierung für Nutzer von Rechnern mit anderen Bildschirmereinstellungen oder Betriebssystemen keine Nachteile hat. Die Höhe des Canvas-Elements wird in der Funktion `calcCanvasHeight()` erzeugt,



Abbildung 5.18: Anzeigefehler bei Überschreitung der maximalen Canvas-Größe

in dem die eingegebene Anzahl von Prozessen mit 60 multipliziert wird. Sollte der errechnete Wert unter 300 liegen (zum Beispiel bei der Eingabe von nur zwei Prozessen), wird er auf 300 gesetzt, da das Diagramm sonst ebenfalls schlecht lesbar ist. Anschließend wird der erste Schritt in das Array `datasets` eingetragen. Dieses wird beim Erstellen eines Diagramms übergeben und enthält alle Daten, die gezeichnet werden sollen. Dazu wird die Funktion `createStep` verwendet, die immer aufgerufen wird, wenn Daten zum Diagramm hinzugefügt werden. Sie dient dazu, die Daten aus dem Array `steps` in ein darstellbares Format für das Array `datasets` der „chart.js“-Bibliothek umzuwandeln. Dieses besteht aus den jeweiligen Start- und Endkoordinaten eines Blocks, Farbe des Blocks, Linienbreite sowie festgelegten Einstellungen zu Farbe und Form der Anfangs- und Endpunkte. Es werden jeweils zwei Dateneinträge von der Funktion erzeugt und zurückgegeben, einer in der „CPU“-Zeile und einer in der dem Prozess zugeordneten Zeile. Dabei wird die Linienbreite mit der Funktion `calcBordwid()` abhängig von der Anzahl der Prozesse errechnet.

Dann wird aus dem Canvas-Objekt, den Arrays `datasets` und `options` ein „Chart“-

Objekt erzeugt und auf das Canvas gezeichnet. Dieses wird im späteren Verlauf immer übergeben, wenn es direkt verändert werden soll, zum Beispiel bei der Anzeige von weiteren Schritten. Anschließend wird die Anzeige rechts vom Diagramm eingerichtet und mit den Informationen zum ersten Schritt aktualisiert.

Zuletzt werden die Buttons zur Navigation durch das Diagramm und Abspielen der Animation aktiviert.

### 5.3.4 Schrittweises Navigieren durch das Diagramm

Mehrere Buttons über dem Diagramm erlauben es dem Nutzer, schrittweise durch das Diagramm zu klicken, um den Vorgang besser nachvollziehen zu können. Dabei kann zum vorherigen und nächsten Schritt sowie zum ersten und letzten Schritt navigiert werden.

Beim „Vorwärts-Navigieren“ werden zusätzliche Daten zum Diagramm hinzugefügt, und die Anzeige wird aktualisiert. Dafür wird jeweils die Funktion `createStep(step)` verwendet, um die nötigen Daten zu erzeugen. Diese werden dann zum existierenden Array `datasets` des „Chart“-Objekts hinzugefügt. Anschließend wird die Funktion „update“-Funktion der Bibliothek „chart.js“ eingesetzt, um das Diagramm zu aktualisieren.

Bei der „Rückwärts-Navigation“ zum ersten Schritt werden alle Einträge mit Ausnahme der ersten zwei Einträge aus dem Array `datasets` des „Chart“-Objekts entfernt. Anschließend wird die Anzeige aktualisiert. Ähnlich wird bei der Navigation zum vorherigen Schritt verfahren. Dabei werden allerdings nur die letzten zwei Einträge aus dem Array `datasets` entfernt.

Bei all diesen Operationen wird überprüft, ob eine Navigation noch möglich ist oder der Nutzer bereits beim ersten oder letzten Schritt angekommen ist. Ist dies der Fall, werden die entsprechenden Buttons deaktiviert.

Des Weiteren können die Schritte als Animation abgespielt werden. Nach Klick auf den Button „Animation abspielen“ wird immer im Abstand von einer Sekunde ein neues Tuple aus Einträgen dem Diagramm hinzugefügt und dieses aktualisiert. Die Navigationsbuttons sowie der Button, der das Einstellungsmenü öffnet, werden deaktiviert und die Schrift auf dem Animationsbutton ändert sich zu „Animation stoppen“. Läuft die Animation ungestört bis zum letzten Schritt durch, werden der Einstellungs- und die beiden Rückwärts-Navigationsbuttons wieder aktiviert und der Text auf dem Animationsbutton zurückgesetzt und dieser deaktiviert. Wird die Animation unterbrochen,

werden alle Buttons wieder aktiviert und die Animation stoppt. Zudem wird der Text auf dem Button zurückgesetzt. Ist man nicht beim letzten Schritt, kann die Animation jederzeit wieder gestartet werden.

## 5.4 Paketisierung

Da die Anwendung in einer Lehrveranstaltung eingesetzt werden soll, muss es möglichst einfach sein, sie an Studenten weiterzugeben. Aus diesem Grund wurden alle JavaScript-Skripte und CSS-Style-Dateien in ein einziges HTML-Dokument integriert. So muss lediglich eine einzelne Datei ohne Abhängigkeiten verschickt oder zum Download bereitgestellt werden. Dadurch wird das Fehlerpotenzial vermindert, da andere Dateien, die zur Funktion des Programms zwingend notwendig sind, nicht versehentlich gelöscht oder nicht mit entpackt werden können.

Die zur Funktion notwendigen Bibliotheken „chart.js“, „math.js“ und das Framework „bootstrap“ wurden in das Dokument per CDN(Content Delivery Network)-Link integriert. Dies bedeutet, dass die zur Verwendung notwendigen Dateien bei Seitenaufruf heruntergeladen werden um die Verwendung zu ermöglichen statt sie lokal zu speichern. So ist zwar ein Internet-Zugriff für die Nutzung notwendig; allerdings müssen keine weiteren Dateien, die deren Offline-Versionen enthalten, mit bereitgestellt werden.

## 5.5 Speicherung des Fortschritts

Die Anwendung sollte den aktuellen Stand im Browser speichern, so dass der Nutzer den Browser schließen kann und anschließend beim nächsten Öffnen der vorherige Zustand wiederhergestellt wird. Auch kann so bei einem versehentlichen Schließen des Fensters die Arbeit direkt fortgesetzt werden, ohne dass der Nutzer die Einstellungen erneut setzen und sich zu einem bestimmten Schritt klicken muss.

In dieser Implementierung wurde dazu die „Web Storage-API“<sup>20</sup> verwendet. Diese ermöglicht es, Paare zusammengesetzt aus Schlüsseln und Wert im Browser zu speichern. Es wurde sich für diese Möglichkeit der Speicherung entschieden, da sie deutlich intuitiver und einfacher ist als die Verwendung von Cookies und von allen neueren Browsern unterstützt wird. Dabei ist es möglich sowohl „sessionStorage“ zu verwenden, die gültig ist, solange der Browser nicht geschlossen wird, als auch „localStorage“, auf die auch beim Neuöffnen des Browsers zugegriffen werden kann. Letztere Daten können nur durch

---

<sup>20</sup>[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Storage\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API)

Löschen der im Browser gespeicherten Daten oder JavaScript-Befehle entfernt werden. Außerdem erlaubt sie größere Datenmengen zu speichern als Cookies oder die „sessionStorage“. Im Hinblick auf die Anforderungen wurde „localStorage“ als Speicherort für die Daten ausgewählt.

Konkret wurde die Speicherung der Daten wie folgend implementiert: Beim ersten Laden der Anwendung wird überprüft, ob in der „localStorage“ ein Wert mit dem Schlüssel `firstTime` vorhanden ist. Ist dies nicht der Fall, wird eine Startseite mit einem leeren Diagramm angezeigt (siehe Abbildung 5.19).

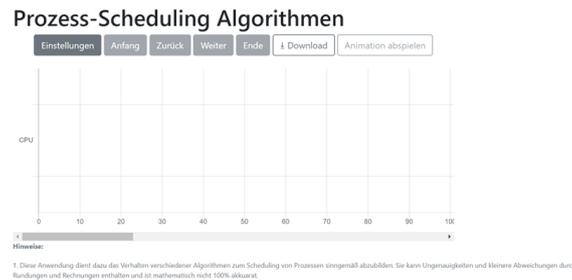


Abbildung 5.19: Anwendung beim ersten Besuch

Nach der erfolgreichen Berechnung der Schritte einer Simulation und Zeichnen des Diagramms wird dieser Wert auf „true“ gesetzt, und weitere für die Wiederherstellung des Diagramms relevante Informationen werden im Browser gespeichert. Diese sind der ausgewählte Algorithmus sowie die Arrays `processdata` und `steps` als JSON-Strings. Während des Zeichnens des Diagramms werden auch die Abstände in der Skalierung der x-Achse gespeichert, damit das Diagramm später wieder in der gleichen Formatierung wiederhergestellt werden kann. Dieser Zeitpunkt wurde ausgewählt um zu vermeiden, dass fehlerhafte Menüeingaben gespeichert werden, die nicht verwendet werden können und von der Validation im Menü abgefangen werden.

Wird die Seite zum wiederholten Mal besucht, werden zunächst die Arrays `processdata` und `steps` aus dem Browserspeicher abgerufen und in den Variablen gespeichert. Anschließend wird das Diagramm mit der im vorherigen Abschnitt erklärten Funktion `resizeChart(chart)` den gespeicherten Daten angepasst und das letzte erfolgreich gezeichnete Diagramm wiederhergestellt. Anschließend wird zum im Browser gespeicherten, letzten angezeigten Schritt navigiert.

Bei Navigation durch das Diagramm wird der aktuell angezeigte Schritt im Browserspeicher stets aktualisiert, damit der Nutzer nach Schließen und erneutem Öffnen der Anwendung gerade bei Simulationen mit vielen Schritten nicht wieder zum letzten Punkt

navigieren muss.

Wird trotz Tests und Vorsichtsmaßnahmen ein fehlerhaftes Diagramm erzeugt, wird bei Neu-Laden der Anwendung eine leere Fläche angezeigt. Das Einstellungs-Menü kann dennoch genutzt werden und ein neues Diagramm gezeichnet werden, so dass die elementare Funktion der Anwendung dadurch nicht beeinträchtigt wird.

Die Anwendung enthält unter dem Disclaimer zu Richtigkeit der Simulation auch die Information, dass das Speichern des Fortschritts nur möglich ist, falls das Speichern von lokalen Daten im Browser aktiviert ist und die Anwendung nicht im „Inkognito-Modus“ genutzt wird. In diesem Fall kann die Anwendung normal genutzt werden, um die Simulation zu erzeugen, zeigt aber bei jedem Neustart die leere Startseite an.

Zudem ist es möglich, das aktuell im Canvas gezeichnete Diagramm als Bild per Knopfdruck herunterzuladen. So kann der Nutzer besonders interessante Diagramme aufbewahren, ohne erneut alle Eingaben vorzunehmen. Die Anwendung zeigt außerdem nur ein Diagramm zur Zeit an. Diese Funktion der Anwendung ermöglicht es, mehrere Diagramme direkt miteinander zu vergleichen.

## 6 Test der Implementierung

Die Anwendung wurde mit einer Kombination aus manuellen Tests und Unit-Tests überprüft. Das Verhalten des Algorithmus „Round Robin“ wurde anhand der korrigierten Lösungen eines Übungszettels aus der Veranstaltung „Technische Grundlagen der Informatik“ überprüft. Das Verhalten der Implementierung des Algorithmus „Kontosystem“ wurde mit Vorlesungsfolien der gleichen Veranstaltung geprüft. Da ich entsprechende Werte für den Algorithmus „Shortest Remaining Job First“ nicht finden konnte, habe ich die Schritte in einem Beispiel von Hand berechnet und dann mit dem Ergebnis verglichen. Diese Vergleichswerte sind im Anhang der Arbeit zu finden. Zum „Linuxscheduler“ Algorithmus existieren im Buch, aus dem die Formeln entnommen wurden, keine Vergleichswerte. Allerdings wurden die Funktionen, die die Werte berechnen und die Prozesse mit der geringsten Priorität auswählen wie im Folgenden geschildert durch Unit-Tests auf ihre Korrektheit überprüft. Aus diesem Grund kann davon ausgegangen werden, dass das Endergebnis entsprechend der verwendeten Formeln korrekt ist. Das gleiche gilt für das Verhalten der selbstkonzipierten Algorithmen, deren Ergebnisse mit einer externen Bibliothek errechnet werden, von deren Korrektheit ausgegangen werden kann.

Bestandteile, die nur intern innerhalb der Berechnung agierten, wurden mit Hilfe von Unit-Tests überprüft, um ihr korrektes Verhalten sicherzustellen. Dazu wurde die Bibliothek „QUnit“<sup>1</sup> verwendet. Dies ist eine Bibliothek für JavaScript, die über einen CDN-Link im Code verwendbar ist und eine Anzeige der Testergebnisse im Browser ermöglicht. Um die Übersichtlichkeit der Anwendung nicht zu beeinträchtigen, wurde eine Kopie der Datei erstellt und um die Tests und die Anzeige der Testergebnisse im Browser erweitert (siehe Abbildung 6.1). Diese Datei wird gemeinsam mit dem Quellcode der Anwendung der Arbeit beigelegt.

---

<sup>1</sup>(<https://qunitjs.com/about/>)

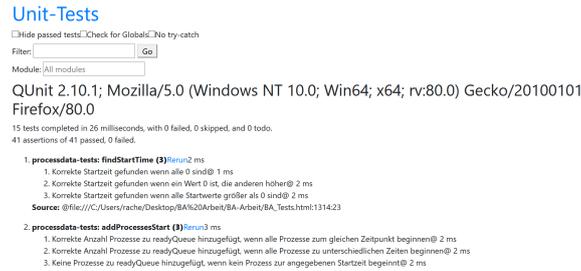


Abbildung 6.1: Anzeige der Ergebnisse der Unit-Tests

Funktionen, die UI-Elemente und das Diagramm beeinflussen, wurden manuell getestet, da die korrekten Reaktionen und Darstellungen so einfacher überprüft werden konnten. Zudem konnte so die Performance in verschiedenen Browsern verglichen und mögliche Anzeigefehler (zum Beispiel durch unterschiedliche Maximalgrößen für Canvas-HTML-Elemente) gefunden werden. Die Anwendung wurde in den Browsern Google Chrome(84.0.4147.135 (64 bit)), Mozilla Firefox (Version 79.0 (64 bit)), Microsoft Edge(84.0.522.52 (64bit)) und Safari(Version 13.1.2 (15609.3.5.1.3)) getestet. Dabei wurde ein Testprotokoll angefertigt, das im Anhang der Arbeit zu finden ist.

Die Anwendung ist in jedem dieser Browser voll funktionsfähig. Bei der Verwendung im Browser Mozilla Firefox konnten allerdings einige kleinere Probleme festgestellt werden. So benötigt sie mit den gleichen Eingabewerten mehr Arbeitsspeicher als in den anderen getesteten Browsern. Bei einer großen Anzahl von Schritten wird die Anwendung langsamer und kann einige Sekunden benötigen, um zum letzten Schritt zu navigieren. Außerdem kann es zu kleineren Anzeigefehlern bei breiteren Diagrammen kommen. Dabei wird zunächst nur ein Teil des Diagramms angezeigt, der Rest ist lediglich eine leere weiße Fläche. Scrollt man zur Seite, benötigt Firefox einige Sekunden, um den nächsten Teil des Diagramms zu laden(siehe Abbildung 6.2).

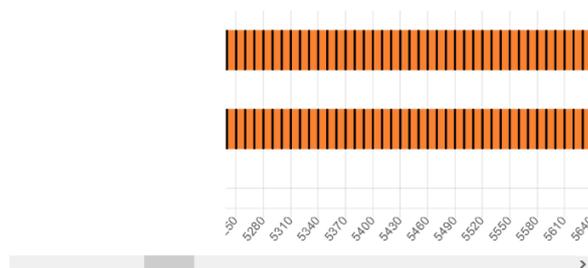


Abbildung 6.2: Anzeigefehler in Firefox

Dieses Problem konnte nicht zuverlässig reproduziert werden. So trat die Schwierigkeit bei den gleichen Eingabewerten am gleichen Rechner nicht immer auf. Wird zu diesem Zeitpunkt das Diagramm als Bild heruntergeladen, ist allerdings das ganze Diagramm ohne weiße Fläche korrekt zu sehen. Es ist daher davon auszugehen, dass der Fehler nicht in der Berechnung oder in der Anwendung liegt sondern browserintern ist. Dieses Problem konnte auch nach Recherche nicht gelöst werden. Da die Wartezeit auf das Laden allerdings gering ist, die Anwendung weiterhin voll funktionsfähig und der Fehler ausschließlich in Firefox zu beobachten ist, kann die Anwendung dennoch sinnvoll eingesetzt werden. Die Anwendung wurde auch im Internet Explorer 12 getestet. Dort ist zwar das Raster des Diagramms sichtbar, allerdings werden die Daten im Diagramm nicht angezeigt. Da der Internet Explorer 12 jedoch nicht mehr zu den gebräuchlichen Browsern gehört und durch Microsoft Edge als vorinstallierter Standardbrowser in Windows ersetzt wurde, wird dieses Problem ebenfalls als eher weniger bedeutsam eingeschätzt.

Die Anwendung wurde ebenfalls auf Geräten mit verschiedenen Bildschirmgrößen getestet und war auf allen getesteten Laptopbildschirmen voll funktionsfähig. Am besten funktioniert die Anwendung in einem Browserfenster in Bildschirmgröße auf einem Desktoprechner oder Laptop. Auch bei kleineren Browserfenstern passt sich die Anzeige an, allerdings sind größere Diagramme weniger angenehm lesbar. Eine Anwendung auf einem Smartphone wurde getestet, ist allerdings auf Grund der Bildschirmgröße nicht übersichtlich oder gut lesbar und daher nicht zu empfehlen. In einem Informatik-Studiengang wie Digitale Medien ist wohl anzunehmen, dass die Studenten Zugang zu einem eigenen Rechner oder dem Rechnerpool der Universität haben. Aus diesem Grund sollte die Anwendung von allen Studenten unter ausschließlicher Berücksichtigung von technischen Voraussetzungen verwendbar sein.

Die Tests bestätigen, dass viele mögliche Eingabefehler gefunden werden und so fehlerhafte und abgebrochene Berechnungen oder Endlosschleifen verhindern. Es ist allerdings nicht möglich, diese Gefahr durch Nutzereingaben bei der eigenen Formeleingabe völlig auszuschließen. So verhindert die Anwendung zwar so gut wie möglich fehlerhafte Eingaben durch fehlende mathematische Operatoren und Klammern etc., kann allerdings Endlosschleifen durch wenig durchdachte eingegebene Formeln nicht ausschließen. Gibt der Nutzer beispielsweise eine Formel zur Berechnung der Zeitscheibenlänge an, die immer oder unter bestimmten Umständen eine Länge von 0 ergibt, werden Prozessen unendlich lange Zeitscheiben zugeordnet, ohne dass ihre Rechenzeit abnimmt. Dieser Vorgang kann im Falle eines Fehlers nicht automatisch abgebrochen werden, da unklar ist, ob der Algorithmus sich in einer Endlosschleife befindet oder lediglich sehr lange

Zeit benötigt, um das Ergebnis zu berechnen. Dieses Problem bezeichnet man in der theoretischen Informatik auch als das Halteproblem. In den Hinweisen in der Anwendung wird der Nutzer darüber informiert, dass er in diesem Fall die Seite aktualisieren muss, um erneute Berechnungen zu beginnen. Auch wird im Menü darauf hingewiesen, welche Werte zu Beginn 0 enthalten. So kann der Nutzer das Teilen durch null möglichst vermeiden.

## 7 Fazit und Ausblick

Im Rahmen dieser Bachelor-Arbeit wurde eine Webanwendung erstellt, die das Verhalten von Algorithmen zum Scheduling von Prozessen abbildet und für die Verwendung in einer Lehrveranstaltung konzipiert wurde. Die Anwendung ist eine SPA, die in einer einzigen HTML-Datei zusammengefasst wurde um möglichst einfach an Studenten verteilbar zu sein. Zudem wurden lediglich Bibliotheken benutzt, die nicht den Erwerb einer Lizenz voraussetzen, um die Verwendung in einer Lehrveranstaltung zu ermöglichen.

Bei der Gestaltung der Benutzeroberfläche wurde darauf geachtet, dass diese ansprechend, übersichtlich und weitgehend selbsterklärend ist. Die Anwendung ist darauf ausgelegt, die in der Lehrveranstaltung erworbenen Grundkenntnisse zu vertiefen und zu erweitern und die dort erlernten Konzepte anzuwenden. Aus diesem Grund enthält die Anwendung nur allgemeine Hinweise zur Benutzung, allerdings keine Erklärungen der einzelnen Algorithmen oder Fachbegriffe. Über das dynamische Menü kann der Nutzer verschiedene Parameter eingeben, die als Basiswerte für die Simulation dienen. Dabei werden Scheduling nach dem Prinzip „Round Robin“ und „Shortest Remaining Job First“ sowie der Linux-Scheduling-Algorithmus und ein aus der Vorlesung „Technische Grundlagen der Informatik“ bekannter kontobasierter Scheduling-Algorithmus als Auswahlmöglichkeiten angeboten. Des Weiteren hat der Nutzer die Möglichkeit, eigene Scheduling-Algorithmen zu konzipieren und deren Verhalten durch Eingabe von Formeln in der Anwendung zu simulieren.

Die Eingaben in diesem Menü werden überprüft, um Fehler durch falsche oder ungültige Eingaben so weit möglich zu vermeiden. Die aus diesen Eingaben generierten Simulationsdaten werden in einem Diagramm dargestellt, durch das der Nutzer mit verschiedenen Buttons navigieren kann. Außerdem kann der Nutzer die Simulation als Animation abspielen oder das aktuell angezeigte Diagramm als Bilddatei herunterladen. Neben dem Diagramm werden Informationen angezeigt, die wichtig sind, um das Verhalten der Scheduling-Algorithmen nachzuvollziehen. Sowohl bei der Gestaltung des Diagramms als auch der Informations-Anzeige wurde auf Übersichtlichkeit und Lesbarkeit besonderer Wert gelegt. Wird die Anwendung geschlossen, wird die aktuelle Ansicht gespeichert

und später wiederhergestellt. Auch wurde die Anwendung auf Korrektheit und Kompatibilität mit verschiedenen gebräuchlichen Browsern getestet. Die Tests zeigten, dass die Anwendung in den verschiedenen Browsern nutzbar ist, und nur im Firefox-Browser kleinere Performance-Probleme zeigt. Das Ziel dieser Arbeit war eine Anwendung zu erstellen, die das Verhalten der ausgewählten Algorithmen sinngemäß abbildet und verdeutlicht. Sie ist nicht absolut mathematisch akkurat, da Werte bei den Berechnungen gerundet werden, erfüllt sie dieses Ziel allerdings erfolgreich. Mit all diesen Eigenschaften erfüllt die Anwendung alle verpflichtenden Ziele und auch ein Großteil der optionalen Anforderungen wurde erreicht.

Nicht umgesetzt wurden die Funktionalitäten für die Einbindung von „I/O Interrupts“ in die Simulation sowie die Möglichkeit, die Rechenzeit der Prozesse auf mehrere verschiedene CPU-Kerne zu verteilen. Die Implementierung dieser Funktionalitäten wurde begonnen, konnte allerdings im Zeitrahmen der Arbeit nicht zufriedenstellend und fehlerfrei umgesetzt werden. Aus diesem Grund wurden diese experimentellen Bestandteile aus der finalen Version der Arbeit entfernt. Die Implementierung dieser Funktionalitäten würde das Verständnis von Vorgängen innerhalb des Betriebssystems und des Verhaltens von Algorithmen zum Scheduling von Prozessen unterstützen und wäre eine Möglichkeit diese Anwendung in der Zukunft weiter auszubauen.

Abschließend kann man sagen, dass eine Anwendung implementiert wurde, die den vorher festgelegten Anforderungen entspricht und geeignet ist, um diese begleitend zu einer Lehrveranstaltung zum Thema Betriebssysteme zu verwenden, um das Gelernte zu vertiefen. Während es noch Bereiche gibt, in denen die Funktionalität der Anwendung ausgebaut werden könnte, konnten fast alle der gewünschten Funktionen im Zeitrahmen der Arbeit implementiert werden.

# Quellenverzeichnis

- Bovet, Daniel P.; Cesati, Marco (2005): Understanding the Linux Kernel. O'Reilly: Sebastopol.
- Cox, Russ; Kaashoek, Frans; Morris, Robert (2011): xv6 – a simple, Unix-like teaching operating system. URL: <https://pdos.csail.mit.edu/6.828/2011/xv6/book-rev6.pdf> (Stand 30.08.20).
- Ehse, Erich (2005): Betriebssysteme – Ein Lehrbuch mit Übungen zur Systemprogrammierung in UNIX/Linux. Pearson Studium: München.
- Flanagan, David(2006): JavaScript – The Definitive Guide. O'Reilly: Sebastopol.
- MDN Contributors(2020): <canvas>: The Graphics Canvas Element. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas> (Stand 30.08.20).
- Tanenbaum, Andrew S. (2015): Modern Operating Systems. Pearson: New Jersey.

# Anhang

## Testprotokoll

|                |                 |                     |                        |                |
|----------------|-----------------|---------------------|------------------------|----------------|
| <b>Browser</b> | Mozilla Firefox | Microsoft Edge      | Google Chrome          | Safari         |
| <b>Version</b> | 79.0 (64 bit)   | 84.0.522.52 (64bit) | 84.0.4147.135 (64 bit) | Version 13.1.2 |

### Speichern des Fortschritts

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Erster Aufruf der Anwendung                   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Startseite mit leerem Diagramm wird angezeigt |                   |                  |           |
| <b>Browser</b>                  | Firefox: X                                    | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Anwendung wird wieder geschlossen, bevor eine erfolgreiche Simulation durchgeführt wurde. Anwendung wird wieder geöffnet. |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Startseite mit leerem Diagramm wird angezeigt   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Round Robin<br>Zeitscheibenlänge: 10<br>Anzahl Prozesse: 3<br>P1: Gesamtlänge: 30; Start: 0<br>P2: Gesamtlänge: 30; Start: 0<br>P3: Gesamtlänge: 30; Start: 0<br>„Simulation starten“-Button wird geklickt<br>„Weiter“-Button wird geklickt bis zu Schritt 4<br>Anwendung wird geschlossen<br>Anwendung wird wieder geöffnet |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anwendung zeigt beim Start das gleiche Diagramm wie vorher bei Schritt 4  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | Funktioniert nicht wenn die Anwendung im „Inkognito-Modus“, „Privatsphäre-Modus“ etc. geöffnet ist  |                   |                  |           |

## Navigieren durch das Diagramm und Buttons

|                                 |                                       |                   |                  |           |
|---------------------------------|---------------------------------------|-------------------|------------------|-----------|
| <b>Test</b>                     | „Einstellungen“-Button wird geklickt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Menü öffnet sich von der linken Seite |                   |                  |           |
| <b>Browser</b>                  | Firefox: X                            | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -                                     |                   |                  |           |

|                                 |                          |                   |                  |           |
|---------------------------------|--------------------------|-------------------|------------------|-----------|
| <b>Test</b>                     | „X“-Button wird geklickt |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Menü schließt sich       |                   |                  |           |
| <b>Browser</b>                  | Firefox: X               | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -                        |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Round Robin<br>Zeitscheibenlänge: 10<br>Anzahl Prozesse: 3<br>P1: Gesamtlänge: 30; Start: 0<br>P2: Gesamtlänge: 30; Start: 0<br>P3: Gesamtlänge: 30; Start: 0<br>„Simulation starten“-Button wird geklickt |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Menü schließt sich<br>„Anfang“- und „Zurück“-Button sind deaktiviert<br>Zwei Blöcke werden im Diagramm angezeigt, einer in der „CPU“-Zeile, einer in der „P1“-Zeile   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Weiter“-Button wird geklickt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | „Anfang“- und „Zurück“-Button sind wieder aktiviert<br>Zwei neue Blöcke werden im Diagramm angezeigt, einer in der „CPU“-Zeile, einer in der „P2“-Zeile |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Ende“-Button wird geklickt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | „Weiter“-, „Ende“- und „Animation abspielen“-Button sind deaktiviert<br>Es sind jetzt im Diagramm 18 Blöcke zu sehen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Zurück“-Button wird geklickt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | „Weiter“-, „Ende“- und „Animation abspielen“-Button sind wieder aktiviert<br>Es sind jetzt im Diagramm 16 Blöcke zu sehen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Anfang“-Button wird geklickt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | „Anfang“- und „Zurück“-Button sind deaktiviert<br>Es sind jetzt im Diagramm 2 Blöcke zu sehen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Animation abspielen“-Button wird geklickt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Alle Buttons außer „Download“ sind deaktiviert<br>Schrift auf dem „Animation abspielen“-Button ändert sich zu<br>„Animation stoppen“<br>Nach ca. 1 Sekunde erscheinen 2 neue Blöcke im Diagramm |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Animation stoppen“-Button wird geklickt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Alle Buttons werden wieder aktiviert<br>Schrift auf dem „Animation stoppen“-Button ändert sich zu<br>„Animation abspielen“<br>Es erscheinen keine neuen Blöcke mehr im Diagramm |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Animation abspielen“-Button wird geklickt<br>Animation wird bis zum Ende abgespielt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | „Anfang“- „Zurück“- und „Einstellungen“-Button werden aktiviert<br>Schrift auf dem „Animation stoppen“-Button ändert sich zu<br>„Animation abspielen“<br>„Animation abspielen“-Button wird deaktiviert<br>Es erscheinen keine neuen Blöcke mehr im Diagramm |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Download“-Button wird geklickt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Es öffnet sich ein Download Fenster für die Datei „scheduler.png“<br>Die heruntergeladene Datei entspricht dem aktuell abgebildeten Diagramm |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

## Menü

### Algorithmus-Dropdown

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Shortest Remaining Job First“ wird ausgewählt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzahl der Prozesse-Eingabefeld wird angezeigt<br>Zeitscheibenlänge-Eingabefeld wird nicht angezeigt |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Kontosystem“ wird ausgewählt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzahl der Prozesse- und Zeitscheibenlänge-Eingabefelder werden angezeigt |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Linuxscheduler“ wird ausgewählt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzahl der Prozesse-Eingabefeld wird angezeigt<br>Zeitscheibenlänge-Eingabefeld wird nicht angezeigt |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Round Robin“ wird ausgewählt   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzahl der Prozesse- und Zeitscheibenlänge-Eingabefelder werden angezeigt |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Round Robin“ wird ausgewählt<br>Testeingabe:<br>Anzahl der Prozesse: 3<br>„Shortest Remaining Job First“ wird ausgewählt |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Eingabefeld „Anzahl der Prozesse“ wird zurückgesetzt  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzahl der Prozesse-Eingabefeld wird angezeigt<br>Infotext zu Formeln wird angezeigt<br>„Kontowert berechnen“- und „Dynamische Zeitscheibenlänge berechnen“-Checkboxen werden angezeigt |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

## Custom-Optionen

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe:<br>Prioritäts-Formel: A5            |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Buchstaben muss ein Operator folgen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel: x+5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Formel enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel: A*                             |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Operator muss eine Zahl oder ein Buchstabe folgen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel: A**B                           |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Operator muss eine Zahl oder ein Buchstabe folgen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel leer lassen und Enter drücken oder aus dem Feld klicken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Eingabe ist leer  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel: A+B; |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Formel enthält ungültige Zeichen  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel: (A+B)) |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Anzahl der Klammern stimmt nicht    |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>Testeingabe: Prioritäts-Formel: (A+K)  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Kontowert kann nur verwendet werden, wenn Kontowertberechnung aktiviert ist |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Formel-Eingabefeld wird unter der Checkbox angezeigt                       |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel: x+5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Formel enthält ungültige Zeichen  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel: A5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Buchstaben muss ein Operator folgen                                    |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel: A* |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Operator muss eine Zahl oder ein Buchstabe folgen                      |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel: A**B |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Operator muss eine Zahl oder ein Buchstabe folgen                        |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel leer lassen und Enter drücken oder aus dem Feld klicken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Eingabe ist leer   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel: A+B; |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Formel enthält ungültige Zeichen   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Kontowert berechnen“-Checkbox wird angeklickt<br>Testeingabe: Kontowert-Formel: A+D |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Dynamische Priorität kann hier nicht verwendet werden                             |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Formel-Eingabefeld wird unter der Checkbox angezeigt  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel: x+5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Formel enthält ungültige Zeichen   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel: A5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Buchstaben muss ein Operator folgen   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel: A* |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Operator muss eine Zahl oder ein Buchstabe folgen   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel: $A^{**}B$ |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Auf einen Operator muss eine Zahl oder ein Buchstabe folgen  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel leer lassen und Enter drücken oder aus dem Feld klicken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Eingabe ist leer  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel: $A+B;$ |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Formel enthält ungültige Zeichen  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | „Custom“ wird ausgewählt<br>„Dynamische Zeitscheibenlänge berechnen“-Checkbox wird angeklickt<br>Testeingabe: Zeitscheibenlänge-Formel: $(A+B)$ |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Anzahl der Klammern stimmt nicht   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

## Zeitscheibenlänge

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Zeitscheibenlänge: 0  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Länge der Zeitscheibe muss größer als null sein |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Zeitscheibenlänge: -5                               |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Eingabe enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Zeitscheibenlänge: B                                |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Eingabe enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Zeitscheibenlänge-Eingabefeld leer lassen und Enter drücken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Eingabe ist leer                          |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

## Anzahl der Prozesse

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Anzahl der Prozesse: B                              |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Eingabe enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Anzahl der Prozesse-Eingabefeld leer lassen und Enter drücken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Eingabe ist leer                            |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Anzahl der Prozesse: 0                                 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Prozessanzahl muss mindestens 1 sein |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Anzahl der Prozesse: 13                                       |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige neben dem Eingabefeld: Prozessanzahl kann nicht größer als 10 sein |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Round Robin<br>Anzahl der Prozesse: 4  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | 4 Prozesseingabezeilen werden angezeigt<br>Eingabezeilen enthalten „Gesamtdauer“- und „Start“-Eingabefelder |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Kontosystem<br>Anzahl der Prozesse: 4   |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | 4 Prozesseingabezeilen werden angezeigt<br>Eingabezeilen enthalten „Gesamtdauer“- „Start“- und „Priorität“-Eingabefelder |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

## Prozessdaten

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Gesamtdauer: 0  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Gesamtdauer muss größer als null sein |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Gesamtdauer: -10  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Gesamtdauer enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Gesamtdauer: x  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Gesamtdauer enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Gesamtdauer-Eingabefeld leer lassen und Enter drücken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Gesamtdauer ist leer   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Start: -10  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Start enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe: Start: x  |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Start enthält ungültige Zeichen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Start-Eingabefeld leer lassen und Enter drücken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Start ist leer   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X                                      | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Kontosystem<br>Anzahl der Prozesse: 4<br>Priorität-Eingabefeld leer lassen und Enter drücken |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Priorität ist leer   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Kontosystem<br>Anzahl der Prozesse: 4<br>Priorität: -5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Priorität enthält ungültige Zeichen                  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Kontosystem<br>Anzahl der Prozesse: 4<br>Priorität: -5 |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Priorität enthält ungültige Zeichen                  |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Kontosystem<br>Anzahl der Prozesse: 4<br>Priorität: X |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Priorität enthält ungültige Zeichen                 |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Linuxscheduler<br>Anzahl der Prozesse: 4<br>Priorität: 0         |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Priorität muss beim Linuxscheduler zwischen 100 und 139 liegen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

|                                 |   |                   |                  |           |
|---------------------------------|---|-------------------|------------------|-----------|
| <b>Test</b>                     | Testeingabe:<br>Algorithmus: Linuxscheduler<br>Anzahl der Prozesse: 4<br>Priorität: 180       |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Anzeige unter dem Eingabefeld: Priorität muss beim Linuxscheduler zwischen 100 und 139 liegen |                   |                  |           |
| <b>Browser</b>                  | Firefox: X  | Microsoft Edge: X | Google Chrome: X | Safari: X |
| <b>Anmerkungen/Abweichungen</b> | -   |                   |                  |           |

## Canvas-Größe

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Windows 10-Laptop mit 150% Bildschirmzoom<br>Testeingabe:<br>Algorithmus: Round Robin<br>Zeitscheibenlänge: 10<br>Anzahl der Prozesse: 1<br>P1: Gesamtdauer: 8000; Start: 0<br>Ende-Button klicken<br>Download-Button klicken und Bild herunterladen |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Größe des heruntergeladenen Bilds: 15000 Pixel   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: - |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

|                                 |  |                   |                  |           |
|---------------------------------|--|-------------------|------------------|-----------|
| <b>Test</b>                     | Windows 10-Laptop mit 100% Bildschirmzoom<br>Testeingabe:<br>Algorithmus: Round Robin<br>Zeitscheibenlänge: 10<br>Anzahl der Prozesse: 1<br>P1: Gesamtdauer: 8000; Start: 0<br>Ende-Button klicken<br>Download-Button klicken und Bild herunterladen |                   |                  |           |
| <b>Erwartetes Ergebnis</b>      | Größe des heruntergeladenen Bilds: 10000 Pixel   |                   |                  |           |
| <b>Browser</b>                  | Firefox: X   | Microsoft Edge: X | Google Chrome: X | Safari: - |
| <b>Anmerkungen/Abweichungen</b> | -  |                   |                  |           |

# Vergleichswerte zur Korrektheit des Algorithmusverhaltens

## Kontosystem

Aus den Vorlesungsfolien der Veranstaltung „Technische Grundlagen der Informatik“ im SoSe2018

### Beispiel

- ▶  $Konto = Konto + Nutzung$
- ▶  $Konto = Konto / 2$
- ▶  $Priorität = Konto + Basis$
  
- ▶ zwei Prozesse:
  - A: Basis-Priorität 0
  - B: Basis-Priorität 60

| t      | 0  | 1   | 2   | 3   | 4   | 5   |
|--------|----|-----|-----|-----|-----|-----|
| $K_A$  | 0  | 0   | 50  | 75  | 37  | 68  |
| $N_A$  | 0  | 100 | 100 | 0   | 100 | 100 |
| $K'_A$ | 0  | 50  | 75  | 37  | 68  | 84  |
| $P_A$  | 0  | 50  | 75  | 37  | 68  | 84  |
| $K_B$  | 0  | 0   | 0   | 0   | 50  | 25  |
| $N_B$  | 0  | 0   | 0   | 100 | 0   | 0   |
| $K'_B$ | 0  | 0   | 0   | 50  | 25  | 12  |
| $P_B$  | 60 | 60  | 60  | 110 | 85  | 72  |

Universität Bremen

## Round Robin und Shortest Remaining Job First

Werte und Lösung für Round Robin übernommen von Übungszettel 5 aus der Vorlesung „Technische Grundlagen der Informatik“ im SoSe2018

|           | Gesamtlaufzeit | Start |
|-----------|----------------|-------|
| <b>P1</b> | 1250           | 250   |
| <b>P2</b> | 1000           | 400   |
| <b>P3</b> | 1700           | 1000  |
| <b>P4</b> | 500            | 0     |
| <b>P5</b> | 6000           | 700   |

Round Robin  
Shortest Remaining Job First

