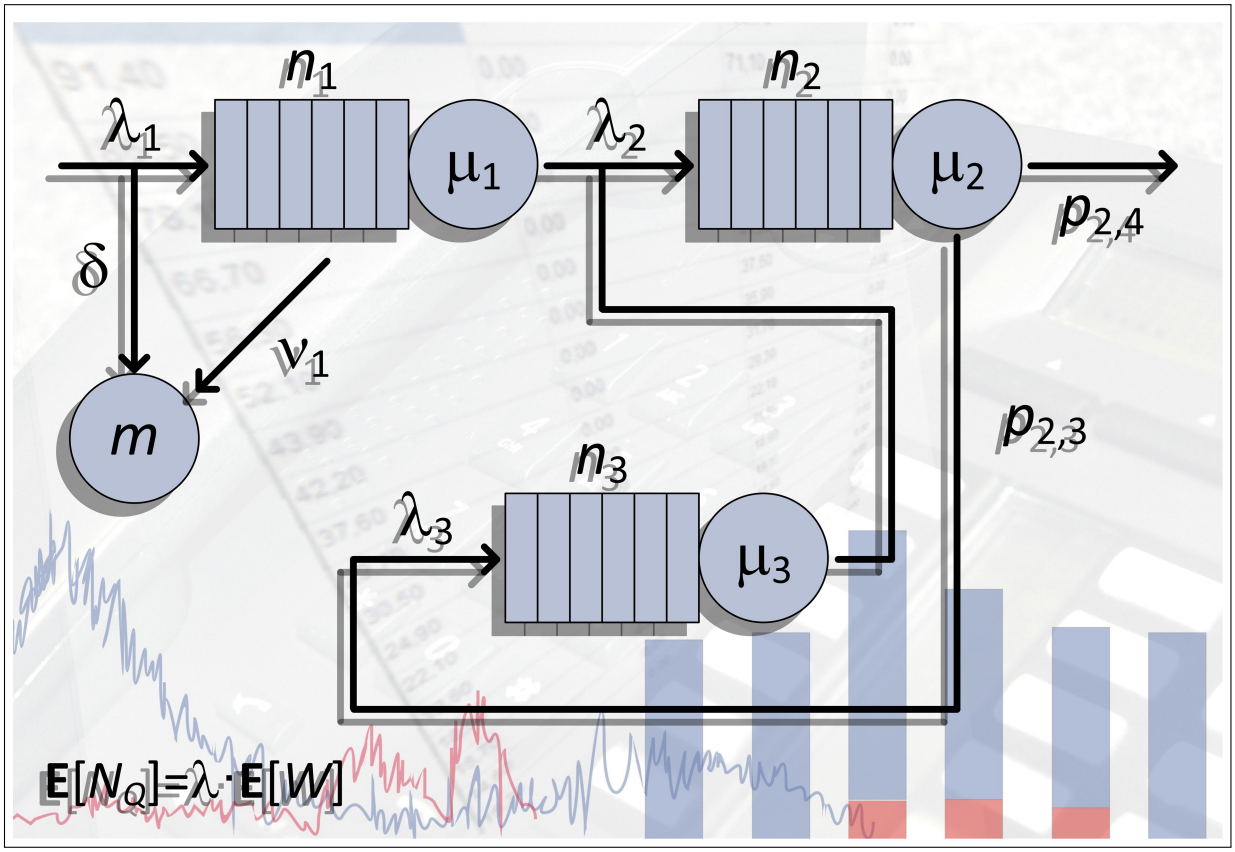
Kommandozeilenbefehlsreferenz Warteschlangensimulator

Alexander Herzog (alexander.herzog@tu-clausthal.de)



Diese Referenz bezieht sich auf die Version 5.9.0 des Warteschlangensimulators.
Download-Adresse: https://a-herzog.github.io/Warteschlangensimulator/.

Inhaltsverzeichnis

Kapitel 1
Allgemeines

Bei dem Warteschlangensimulator handelt es sich um ein Java-Programm, d. h. zur Ausführung wird eine Java-Laufzeitumgebung benötigt. Ist diese vorhanden, so kann der Simulator über folgenden Befehl gestartet werden:

```
java -jar Simulator.jar
```

Wird an diese Befehlszeile durch ein Leerzeichen abgetrennt der Dateiname einer Model-, Statistik-, Parameterreihen- oder Optimiererdatei angehängt, so wird diese unmittelbar nach dem Programmstart geladen.

Kapitel 2
Liste der Befehle

Alternativ kann der Warteschlangensimulator mit einem der in den folgenden Abschnitten beschriebenen Befehle aufgerufen werden. In diesem Fall wird kein Programmfenster angezeigt, sondern alle Ausgaben erfolgen direkt auf der Kommandozeile. Das Aufrufschema der Befehle lautet dabei:

```
java -jar Simulator.jar Befehl OptionenFürBefehl
```

Alle im Folgenden beschriebenen Befehle können auch direkt über die grafische Programmoberfläche über den Befehl „Kommandozeilenbefehl ausführen...“ im „Extras“-Menü ausprobiert werden.

2.1 `Ausgabetabelle`

Verarbeitet eine an einer Speichern+Ausgang-Station erzeugte Tabelle.

Dieser Befehl erwartet zwei weitere Parameter:
1. Eingabe-Tabellendatei
2. Eingabe-Tabelledatei
Die Eingabedatei muss existieren, die Ausgabedatei darf nicht existieren.

2.2 `Dezimaltrenner`

Stellt den Modus für den Dezimaltrenner ein.

Der Befehl erwartet den zukünftig zu verwendenden Dezimaltrenner-Modus als Parameter.
Zur Auswahl stehen die Modi „os„, „language„, „comma“ und „point„.

2.3 `Export`

Exportiert ein Modell.

Dieser Befehl erwartet genau zwei weitere Parameter:
1. Name der Eingabe-Datei
2. Name der Ausgabedatei, die die Exportergebnisse aufnehmen soll
Die Eingabedatei muss existieren, die Ausgabedatei darf nicht existieren.

2.4 Filter

Datei mit Statistikergebnissen filtern.

Dieser Befehl erwartet genau drei weitere Parameter:
1. Name der Eingabe-Statistik-Datei
2. Name der Eingabe-Filterkonfigurationsdatei
3. Name der Ausgabedatei, die die Filterergebnisse aufnehmen soll
Wenn die Ausgabedatei bereits existiert, werden die neuen Ergebnisse angehängt.
Ansonsten wird die Datei neu angelegt.

2.5 Hilfe

Zeigt diese Hilfe an.

Dieser Befehl erwartet einen oder keine weiteren Parameter.
Wird ein Befehl als zusätzlicher Parameter angegeben, so wird die Hilfe zu diesem Befehl angezeigt.
Ansonsten wird die Hilfe zu allen Befehlen angezeigt.

2.6 Interaktiv

Startet den interaktiven Modus.

Dieser Befehl erwartet keine weiteren Parameter.

2.7 JavaProperties

Gibt die Java-Umgebungsvariablen aus.

Gibt die Java-Umgebungsvariablen aus.
Führt keine weiteren Verarbeitungen durch.

2.8 Optimierung

Führt eine Modell-Optimierung durch.

Dieser Befehl erwartet genau zwei weitere Parameter:
1. Eingabe-Modell-Datei
2. Optimiererkonfigurations-Datei
Beide Eingabedateien müssen existieren.

2.9 Parameterreihe

Führt eine Parameterreihen-Simulation durch.

Dieser Befehl erwartet genau zwei weitere Parameter:
1. Eingabe-Parameterreihen-Datei
2. Ausgabe-Parameterreihen-Datei

Die Eingabedatei muss existieren.
Die Ausgabedatei darf nicht existieren.

2.10 ParameterreiheTabelle

Exportiert die Ergebnistabelle einer Parameterreihe.

Dieser Befehl erwartet genau zwei weitere Parameter:
1. Eingabe-Parameterreihen-Datei
2. Ausgabe-Tabellendatei
Die Eingabedatei muss existieren.
Die Ausgabedatei darf nicht existieren.

2.11 ParameterreiheVarianzanalyse

Erstellt eine Parameterreihen-Konfiguration für eine Varianzanalyse

Dieser Befehl erwartet drei oder vier weitere Parameter:
1. Eingabe-Modell-Datei
2. Ausgabe-Parameterreihen-Datei
3. Anzahl an Wiederholungen des Modells
4. Neuer Wert für die Anzahl an Ankünften (optional)
Die Eingabedatei muss existieren.
Die Ausgabedatei darf nicht existieren.

2.12 Server

Simulator als Rechenserver starten.

Dieser Befehl erwartet 0 bis 2 weitere Parameter.
Werden entsprechend viele Parameter angegeben, so haben diese folgende Bedeutungen:
1. Zu verwendender Port
2. Passwort für die verschlüsselte Datenübertragung
Wird kein Parameter übergeben, so wird der Port per Kommandozeile abgefragt.

2.13 ServerLimited

Simulator als Rechenserver starten.

Dieser Befehl erwartet 0 bis 2 weitere Parameter.
Werden entsprechend viele Parameter angegeben, so haben diese folgende Bedeutungen:
1. Zu verwendender Port
2. Passwort für die verschlüsselte Datenübertragung
Wird kein Parameter übergeben, so wird der Port per Kommandozeile abgefragt.
Der Server begrenzt die Anzahl an gleichzeitigen Anfragen basierend auf der Anzahl an verfügbaren CPU-Kernen.

2.14 ServerMQTT

Simulator als MQTT-basierten Rechenserver starten.

Dieser Befehl erwartet zwei bis vier weiteren Parameter:
die Adresse des zu verwendenden MQTT-Brokers, den Namen des MQTT-Topic und optional den Namen des Statusmeldungen-Topics und optional Nutzername und Passwort getrennt durch „:„.

2.15 ServerMQTTFixed

Simulator als MQTT-basierten Rechenserver für ein festes Modell starten starten.

Dieser Befehl erwartet drei bis fünf weiteren Parameter:
die Adresse des zu verwendenden MQTT-Brokers,
den Namen des MQTT-Topic,
optional den Namen des Statusmeldungen-Topics und
optional Nutzername und Passwort getrennt durch „:„.

2.16 ServerMQTTTest

Simulator als MQTT-basierten Echo-Testserver starten.

Dieser Befehl erwartet zwei oder drei weiteren Parameter:
die Adresse des zu verwendenden MQTT-Brokers, den Namen des MQTT-Topic optional Nutzername und Passwort getrennt durch „:„.

2.17 ServerSocket

Simulator als Socket-basierten Rechenserver starten.

Dieser Befehl erwartet einen oder zwei weiteren Parameter:
den zu verwendenden Port und optional einen Timeout-Wert (in Sekunden) für einzelne Simulationen.

2.18 ServerWeb

Simulator als webbasierten Rechenserver starten.

Dieser Befehl erwartet einen oder zwei weiteren Parameter:
den zu verwendenden Port und optional Nutzername und Passwort getrennt durch „:„.

2.19 ServerWebFixed

Simulator als webbasierten Rechenserver für ein festes Modell starten.

Dieser Befehl erwartet zwei oder drei weitere Parameter:
den zu verwendenden Port, den Dateinamen

der Modelldatei
und optional Nutzername und Passwort getrennt durch „:„.

2.20 Setup

Interaktive Konfiguration des Simulators.

Dieser Befehl erwartet keine weiteren Parameter.

2.21 Simulation

Führt einen einzelnen Simulationslauf durch.

Dieser Befehl erwartet zwei oder drei weitere Parameter:
1. Eingabe-Modell-Datei
2. Eingabe-Datentabelle (optional)
3. Ausgabe-Statistik-Datei
Die Eingabedatei und die Eingabe-Datentabelle (sofern angegeben) müssen existieren, die Ausgabedatei darf nicht existieren.

2.22 SimulationTimeout

Führt einen einzelnen Simulationslauf (mit Timeout) durch.

Dieser Befehl erwartet drei oder vier weitere Parameter:
1. Eingabe-Modell-Datei
2. Eingabe-Datentabelle (optional)
3. Ausgabe-Statistik-Datei
4. Timeout-Wert (in Sekunden)
Die Eingabedatei und die Eingabe-Datentabelle (sofern angegeben) müssen existieren, die Ausgabedatei darf nicht existieren.

2.23 Sprache

Stellt die Programmsprache ein.

Der Befehl erwartet das Kürzel der zukünftig zu verwendenden Sprache als Parameter.

2.24 Version

Gibt die aktuelle Versionsnummer aus.

Gibt die Versionsnummer des Warteschlangensimulators aus.
Führt keine weiteren Verarbeitungen durch.

2.25 Verzeichnis

Simuliert alle Modelle und Parameterreihen in einem Verzeichnis.

Dieser Befehl erwartet als Parameter den Namen des zu verarbeitenden Verzeichnisses.

2.26 VerzeichnisFilter

Alle Statistikdateien in einem Verzeichnis filtern.

Dieser Befehl erwartet genau drei weitere Parameter:
1. Name des Verzeichnisses
2. Name der Eingabe-Filterkonfigurationsdatei
3. Name der Ausgabedatei, die die Filterergebnisse aufnehmen soll
Wenn die Ausgabedatei bereits existiert, werden die neuen Ergebnisse angehängt.
Ansonsten wird die Datei neu angelegt.

2.27 Zusammenfassung

Exportiert einen Teil oder die gesamten Simulationsergebnisse für ein Modell.

Dieser Befehl erwartet genau drei weitere Parameter:
1. „Inline„, „Einzeldateien„, „Liste„, „Text„, „PDF“ , „LaTeX„, „HTMLApp“ oder ein Listeneintrag je nach dem, ob
a) ein HTML-Report mit eingebetteten Bildern,
b) ein HTML-Report mit Bildern in separaten Dateien,
c) eine Übersicht über alle verfügbaren Einzeldokumente,
d) ein DOCX-Report,
e) ein PDF-Report,
f) ein LaTeX-Report,
g) ein HTML-Web-App-Report oder
h) ein bestimmtes Einzeldokument ausgegeben werden soll.
2. Dateiname der Eingabedatei
3. Dateiname der Ausgabedatei