# Bezeichnung von Warteräume

Es gibt unterschiedliche Arten von Warteräume, diese werden nach der Kendall-Symbolik bezeichnet.

A/B/m/n

* A und B kennzeichnen die Art des Warteraumes und der Bedienung der Kunden,  
  wichtige Arten sind
  + M … Markov Exponentielle Verteilung der Zeitspanne zwischen zwei Eintritten
  + D … Deterministisch Ankunftszeitpunkte sind festgelegt
  + G … General Es liegt eine beliebig andere Funktion zugrunde
* m gibt die Anzahl der Bedienstationen an
* n gibt die Größe des Warteraums an.

# Formel- und Variablenbedeutung

Einfache Formeln ergeben sich für MM1∞, also für exponentialverteilte Ankünfte und Bedienungen und einen unendlich großen Warteraum, in dem keine Kunden abgewiesen werden müssen.

*Ta* Zwischenankunftszeit (die Zeit, die zwischen der Ankunft von 2 Kunden vergeht)  
 Ankunftsrate = 1/Ta (die Anzahl der Kunden, die pro Zeit im System ankommen)  
*Ts* Bedienzeit (die Zeit, die benötigt wird, um einen Kunden zu bedienen)  
 Bedienrate (die Anzahl an Kunden, die pro Zeit bedient werden können)  
 Auslastung des Systems (muss < 1 sein, damit die Warteschlange nicht unaufhörlich wächst)  
N Anzahl der Kunden, die gerade im System sind

= Gesamte Auslastung des Systems, wenn darin mehrere Warteschlangen mit unterschiedlichen Bedienraten bedient werden.

*Tv* Verweilzeit = Tw + Ts (Gesamtzeit, die der Kunde im System verbleibt)

Verweilzeit eines Kunden – dann ergeben sich evtl für jeden Kunden andere Tw

mittlere Wartezeit in der Schlange vor der Bedienung = ca der Mittelwert aller Tw, die man aus der Formel für Tv berechnet hat in dem man Tw=Tv-Ts verwendet

Ts =

**Beispiel Server der gleichzeitig ein Druckserver und Authentifizierungsserver ist:**

Drucker: Authentifizierung:  
Eingangsrate: 5 Druckjobs / Stunde Eingangsrate: 10 Anfragen/ Stunde  
Bedienzeit: 3 Minuten Bedienzeit: 1 Minute

Erklärung Formel

Tw = Durchschnittliche Wartezeit eines Jobs im System

Durchschnittliche Ankunftsrate in das System

Gesamte Auslastung des Systems =

Drucker Authentifizierung:

= 5 = 10  
 = 60/Bedienrate = 60/3 = 20 = 60/Bedienrate = 60/1 = 60

**Die Auslastung des Servers beträgt 41%.**

Wartezeit = Zeit von Ankunft bis zur Bedienung

Bedienzeit = Benötigte Zeit für die Bearbeitung eines Auftrages

Verweilzeit = Zeit des gesamten Auftrags (Wartezeit + Bedienzeit)

Wartezeit:

Bedienzeit:

tbDrucker = 3 min

tbAuthentifizierung = 1 min

Verweilzeit:

tvDrucker = tbDrucker + twDurchschnitt = 3 + 1,13966 = **4,13966**

tvAuthentifizierung = tbAuthentifizierung + twDurchschnitt = 1 + 1,13966 = **2,13966**

Jobs Im System:

= **0,6949 Jobs im System**

# Simulation von Warteschlangen

Zur Simulation von Warteschlangen gibt es zwei Möglichkeiten:

## Threads (paralell)

Es werden 2 Threads benötigt.

Der erste Thread übernimmt die Ankunft, währenddessen der zweite die Bedienung übernimmt. Als Warteraum sollten Semaphore verwendet werden.

Beide Threads simulieren die Bearbeitungszeit bzw. Zeitspanne zwischen den Eintritten von Paketen mittels sleep. Dabei muss beachtet werden, dass die Zeitmessung von sleeps auf Betriebssystemen recht ungenau sind. Man kann höchstens 20 Kunden pro Sekunden simulieren, falls eine halbwegs gute Genauigkeit erfordert wird.

## Sequentiell

Man gibt eine gewisse Simulationsdauer T vor, teilt diese in kleine regelmäßige Zeitabschnitte dt beobachtet was in einem Zeitabschnitt geschieht:

while (gesamtzeit<T)

{

gesamtzeit=gesamtzeit+dt;

while (azeit <= gesamtzeit)

{

n++;

a=exporand(lambda);

azeit = azeit + a;

}

while (bzeit <= gesamtzeit && n>0)

{

n--;

b=exporand(mu);

bzeit = bzeit + b;

}

if(azeit>bzeit&&n==0) bzeit=azeit;

N = N + n;

steps++;

}

N = N / steps;

Erklärung:   
gesamtzeit gibt an wieviel von der gesamten Simulationsdauer schon vergangen ist  
azeit … die Ankunftsintervalle zusammengezählt, da kann es keine Lücken geben  
  
erste while Schleife: Ankünfte. während des Intervalls dt können solange Leute ankommen, solange der Abstand zwischen zwei Ankünften klein genug sind

Zweite while Schleife: Bedienung es können solange Klienten bedient werden, bis der Abschluss der Bedienung über das gerade betrachtete Intervall hinaus dauern würde. Wenn kein Kunde im System ist, gibt es keine Bedienung, deshalb das if > 0

if(azeit>bzeit …) hier wird berücksichtigt, dass ein Kunde sofort bedient werden kann, wenn er der erste ist.  
n enthält die Anzahl der Kleinsten in der Warteschlange und wird aufsummiert um die mittlere Länge der Warteschlange zu berechnen.