

D I P L O M A R B E I T

Analytische Untersuchung von Schedulingalgorithmen unter Echtzeitbedingungen

ausgeführt am Institut für Automation

der Technischen Universität Wien

unter Anleitung von

Univ.Doz. Dipl.-Ing. Dr. Ulrich Schmid

und

Univ.Ass. Dipl.-Ing. Dr. Johann Blieberger

durch

Helmut Reinwein

Matr.Nr. 8025312

Castelligasse 14/1/8

2020 Hollabrunn

September 1996

Zusammenfassung

Analytische Untersuchung
 von Schedulingalgorithmen
 unter Echtzeitbedingungen

von Helmut Reinwein

Wir untersuchen das Echtzeitverhalten von Computersystemen im Rush-Hour-Case bei verschiedenen Schedulingalgorithmen (FCFS, pLCFS, nLCFS), wobei das Hauptaugenmerk auf pLCFS gelegt wird. Man erkennt jedoch bald, daß pLCFS nur als Stellvertreter für alle möglichen Schedulingalgorithmen (nicht nur für die oben angeführten) im Rush-Hour-Case fungiert und das Ergebnis daher für alle solche gilt.

Konkret wird die Zufallsvariable $SRD(T)$ untersucht, welche ein Maß für die Zeit ist, bis zu welcher das Computersystem ohne Verletzung einer vorgegebenen Deadline T der Task-Service-Time läuft. Zu deren Berechnung wird ein Baum-Modell herangezogen.

Es stellt sich heraus, daß für *alle* Schedulingalgorithmen im Rush-Hour-Case — nicht nur für die hier untersuchten, sondern tatsächlich für alle möglichen! — bei wachsender Deadline T die mittlere $SRD(T)$ gegen einen gleichen, nur von der Wahrscheinlichkeitsverteilung der ankommenden Tasks (genauer: der ankommenden Cycles) abhängenden Wert strebt, d. h., daß $E[SRD(T)] \rightarrow \frac{1}{1-\beta} \cdot \frac{\beta}{1-P'(\beta)}$ (wenn $T \rightarrow \infty$), wo $P(x)$ die PGF eben dieser Wahrscheinlichkeitsverteilung ist und β ein Fixpunkt davon.

Das Echtzeitverhalten des Schedulers im Rush-Hour-Case ist also bei großer Deadline T *gänzlich unabhängig* von der gewählten Scheduling-Strategie!

SCHLÜSSELWÖRTER: Schedulingalgorithmen, FCFS, pLCFS, nLCFS, Echtzeitverhalten, Rush-Hour-Case, Baum-Modell, wahrscheinlichkeitserzeugende Funktionen (englisch: Probability Generating Function PGF), asymptotisches Verhalten, (Sub-)Busy-Period, Successful-Run.

DANKSAGUNG

An dieser Stelle möchte ich mich besonders bei meinem Betreuer für die sehr kompetente und unkomplizierte Art der Betreuung bedanken. Er verstand es auch, mich geduldig und mit aufmunternden Worten durch die Fährnisse des Schreibens der Diplomarbeit zu geleiten.

Weiters möchte ich auch meinem Arbeitgeber, Fa. Siemens AG Österreich, dafür danken, daß ich die dortige HW- und SW-Infrastruktur für die Erstellung der Diplomarbeit nutzen durfte (SINIX-Rechner mit komplettem \LaTeX -System, PC mit X-Terminal-Emulation, um die \LaTeX -Zwischenergebnisse leicht kontrollieren zu können, und einen E-Mail- und Internet-Anschluß, um damit anfangs \LaTeX holen zu können und später leichter mit meinem Diplomarbeits-Betreuer kommunizieren zu können).

Und zuletzt möchte ich auch meinem zwiespältigen Verhältnis zu \LaTeX Ausdruck verleihen, das mich einerseits zu Beginn sehr viel Zeit und Arbeit für den Transfer der Sourcen, das Übersetzen und Konfigurieren gekostet hat, mich aber andererseits für diese Mühen dann beim Schreiben der Diplomarbeit mit einer leichten und intuitiven Bedienung und einem sehr professionellen Layout belohnt hat.

Inhaltsverzeichnis

I	Die Problemstellung	1
Kapitel 1	Einleitung	2
1.1	Problemstellung	2
1.2	Begriffsbestimmungen	3
1.3	Vorschau	11
II	Wahrscheinlichkeits-Modell und Baum-Modell	13
Kapitel 2	Wahrscheinlichkeits-Modell	14
2.1	Grundzüge der Wahrscheinlichkeitstheorie	14
2.2	Definitionen: $A(z)$, $L(z)$, $P(z)$	17
2.2.1	Anzahl der Task-Ankünfte während eines Cycle: $A(z)$	17
2.2.2	Task-Execution-Time (= Task-Länge): $L(z)$	18
2.2.3	Anzahl der Cycle-Ankünfte während eines Cycle: $P(z)$	18
2.3	Aussagen über $P(z)$	18
2.3.1	Herleitung von $P(z)$ aus $A(z)$ und $L(z)$	19
2.3.2	Weitere Aussagen über $P(z)$ im Rush-Hour-Case	19
2.3.3	Weitere Aussagen über $P(x)$, x reell, im Rush-Hour-Case	21
2.3.4	Schranken von $P'(\beta)$ im Rush-Hour-Case	22
Kapitel 3	Modellierung der ankommenden Tasks mittels eines Baum-Modells	24
3.1	Baum-Modell bei FCFS	25
3.2	Baum-Modell bei pLCFS	27
3.3	Baum-Modell bei nLCFS	28
3.4	Symbolische Notation von Bäumen, die OGF	30
III	Busy-Period und Sub-Busy-Period	32

Kapitel 4	Die Busy-Period-Duration BPD	33
4.1	Definition der BPD: $B(z)$	33
4.2	Herleitung der BPD aus $P(z)$, Berechnung von $B(z)$	33
4.3	Aussagen über die BPD	34
4.3.1	Berechnung von $B'(z)$	34
4.3.2	Berechnung von $B(1)$	35
4.3.3	Berechnung von $B'(1)$	37
4.4	Unabhängigkeit der BPD von der Scheduling-Strategie	38
Kapitel 5	Die Sub-Busy-Period-Duration SBPD	40
5.1	Unabhängigkeit der SBPD von der Scheduling-Strategie	40
5.2	Symbolische Notation der SBP, die OGF der SBP bei pLCFS	41
5.3	Definition der SBPD bei pLCFS: $\overline{B}(z)$	43
5.4	Herleitung der SBPD aus BPD bei pLCFS, Berechnung von $\overline{B}(z)$	43
5.5	Aussagen über die SBPD bei pLCFS	45
5.5.1	Berechnung von $\overline{B}'(z)$ bei pLCFS	45
5.5.2	Berechnung von $\overline{B}(1)$ bei pLCFS	45
5.5.3	Berechnung von $\overline{B}'(1)$ bei pLCFS	46
Kapitel 6	BPD und SBPD bei Berücksichtigung der Deadline T	48
6.1	Definition der BPD bei Deadline T : $B_T(z)$	48
6.2	Definition der SBPD bei Deadline T : $\overline{B}_T(z)$	49
6.3	Herleitung der BPD bei Deadline T , Berechnung von $B_T(z)$	50
6.4	Aussagen über die BPD und SBPD bei Deadline T	52
6.4.1	Berechnung von $B_T'(z)$	52
6.4.2	Berechnung von $B_T(1)$	52
6.4.3	Berechnung von $B_T'(1)$	53
6.5	Abschätzung und Grenzwert der BPD bei Deadline T ($B_T(z)$)	53
6.5.1	Abschätzung von $B_T(z)$	53
6.5.2	Grenzwert von $B_T(z)$	54
6.6	Abschätzung und Grenzwert der SBPD bei Deadline T ($\overline{B}_T(z)$)	56
6.6.1	Abschätzung von $\overline{B}_T(z)$	56
6.6.2	Grenzwert von $\overline{B}_T(z)$	57
6.7	Grenzwerte der (höheren) Ableitungen von $B_T(z)$ und $\overline{B}_T(z)$	58

6.8	Unabhängigkeit der Grenzwerte (der Ableitungen) von $B_T(z)$ bzw. von $\overline{B}_T(z)$ von der Scheduling-Strategie	59
IV	Der Successful-Run	61
Kapitel 7	Die Successful-Run-Duration $SRD(T)$	62
7.1	Definition der $SRD(T)$: $S_T(z)$	62
7.2	Herleitung der $SRD(T)$	62
7.3	Berechnung des Mittelwertes der $SRD(T)$	64
7.4	Grenzwert des Mittelwertes der $SRD(T)$	64
Kapitel 8	Quintessenz und Ausblick	66
8.1	Quintessenz	66
8.2	Ausblick	68
	Abbildungsverzeichnis	69
	Tabellenverzeichnis	70
	Literatur	71

Teil I

Die Problemstellung

Kapitel 1

EINLEITUNG

1.1 Problemstellung

Während bei herkömmlichen Computern (PCs, Server, etc.), welche im interaktiven Modus oder auch im Batch-Betrieb verwendet werden, ein gutes Antwortzeitverhalten erwünscht und benutzerfreundlicher ist, gibt es sehr viele Computer-Anwendungen, wo es nicht nur erwünscht, sondern unbedingt (oft sogar lebens-) notwendig und daher eine wesentliche Voraussetzung für den funktionierenden Ablauf ist, daß “das System” binnen kürzester Zeit, genauer binnen einer genau definierten maximalen Antwortzeit, auf Aktionen (Terminaleingabe, Eintritt in einen bestimmten Zustand, Auftreten eines bestimmten Ereignisses, etc.) reagiert. Als Beispiel seien hier nur Prozeßrechner fast jeder Art angeführt, welche in der Industrie vielfältig zur Anwendung kommen, sei es bei vollautomatischen Fertigungsstraßen, in der Robotertechnik oder bei sonstigen zeitkritischen Prozeßabläufen.

Computer oder Anwendungen, welche speziell für solche Zwecke gebaut bzw. programmiert wurden, heißen *Echtzeitrechner* bzw. *Echtzeitanwendungen*. Um die Untersuchung des Antwortzeitverhaltens bei solchen Echtzeit-Aufgabenstellungen geht es im hier vorliegenden Papier.

Konkret werden hier einige Aspekte des Echtzeitverhaltens eines Servers mit verschiedenen Schedulingalgorithmen untersucht, und zwar in erster Linie der Mittelwert derjenigen Zeit, bis zu welcher die maximale Antwortzeit (die Task-Deadline) zum ersten Mal überschritten wird. Folgende Schedulingalgorithmen werden hierbei näher behandelt: “*First Come First Serve*”, “*preemptive Last Come First Serve*” und “*nonpreemptive Last Come First Serve*”, jeweils im *Rush-Hour-Case*, d. h. bei hoher Task-Ankunftsrate (genauer: wenn die Ankunftsrate größer ist als die Abarbeitungsrate; siehe dazu Kapitel 1.2), wofür respektive die selbsterklärenden Abkürzungen FCFSr, pLCFSr und nLCFSr verwendet werden.

Als Untersuchungsmethode verwende ich hier jedoch nicht die eventuell naheliegende Queuing-Theorie, sondern die Methode unter Zuhilfenahme einer Baum-Modellierung, welche ja aus der Analyse von Datenstrukturen bekannt ist (siehe [10] und [11]) und welche sich auch bei ähnlichen Untersuchungen (siehe [16], [3],

[17], [2]) sehr bewährt hat. Grundlagen über die hier verwendeten *wahrscheinlichkeitserzeugende Funktionen* (englisch: *Probability Generating Function* PGF) findet man auch in [10], während verschiedene asymptotische Methoden recht gut in [1] beschrieben sind. Aussagen über die Grenzverteilung in Queueing-Systemen werden in [5] getroffen.

1.2 Begriffsbestimmungen

Wir betrachten ein idealisiertes, vereinfachtes Computersystem bestehend aus einem Task-Scheduler, einer (potentiell unendlich großen) Task-List und einem Server, welcher den aktuell ausgewählten Task bearbeitet. Neu beim Computersystem ankommende Tasks werden vom Scheduler gemäß der Scheduling-Strategie in die Task-List eingefügt, während der Server immer den Task am anderen Ende der Task-List bearbeitet. Je nach Scheduling-Strategie genügt es, die neuen Tasks der Reihe nach vorne in der Task-List anzufügen (FCFS) oder der Scheduler muß auch (bei pLCFS und nLCFS) die Tasks innerhalb der Task-List neu anordnen, sodaß eben am Ende der Task-List genau der Task steht, welcher vom Server im nächsten Schritt bearbeitet werden soll. Sollte die Task-List leer werden, so generiert der Scheduler einen Dummy-Task, welcher sodann auch vom Server abgearbeitet wird, d.h. es wird dann eben nichts getan.

Vorweg sei noch erwähnt, daß gerade in einem Fachgebiet wie der Computertechnologie die englische Sprache doch sehr dominierend ist und daher auch viele Fachausdrücke entweder ohnehin auf englisch allgemein verständlich sind oder sie eingedeutscht zumindest ungewöhnlich, wenn nicht sogar verwirrend klingen. Daher wird bei der folgenden Begriffsbestimmung und im ganzen vorliegenden Papier großteils der englische Fachausdruck beibehalten und nur in der Hinsicht abgeändert, daß er bei hauptwörtlichem Gebrauch mit großem Anfangsbuchstaben geschrieben wird bzw. daß ein Fachausdruck, der aus mehreren englischen Wörtern besteht, durch das Verbinden dieser Wörter jeweils mit einem Bindestrich (“-”) zu einem einzigen verständlichen Fachausdruck zusammengezogen wird.

Im Folgenden werden die Grundlagen und in diesem Zusammenhang immer wieder auftretende Begriffe erklärt:

CYCLE: Das Neu-Anordnen der Task-List durch den Scheduler möge in unserem Modell immer zu genau definierten, diskreten Zeitpunkten erfolgen, welche immer eine konstante Zeitdauer voneinander entfernt sind. Ein Cycle sei nun

genau diese konstante Zeitdauer zwischen zwei dieser diskreten Zeitpunkte. Dem zu Folge können wir nun auch jeden Task in seine kleinsten unteilbaren (=atomaren) Einheiten mit der Dauer von je genau einem Cycle aufteilen.

DUMMY-TASK: Ist ein vom Scheduler generierter Task der Länge eines Cycle, welcher genau dann generiert wird, wenn die Task-List leer wird.

IDLE: Das System ist im Zustand idle, wenn der Server einen Dummy-Task bearbeitet, d. h. also, wenn es sonst nichts zu tun gibt.

BUSY: Ansonsten ist das System im Zustand busy, wenn also der Server einen "echten" Task (und keinen Dummy-Task) bearbeitet.

TASK-EXECUTION-TIME (= TASK-LÄNGE): Ist die Anzahl von Cycles, welche für die vollständige Bearbeitung des Tasks benötigt werden, wenn sich der Server von Anfang bis zum Schluß des Tasks exklusiv um diesen Task kümmert. Sie ist daher sozusagen die Länge des Tasks, gemessen in Cycles.

TASK-SERVICE-TIME: Ist die Anzahl von Cycles vom Eintreffen des Tasks beim Computersystem (genauer: vom Ende des Cycles an, in welchem der Task eintrifft, gemessen) bis zum Abarbeiten des letzten Cycle des Tasks (bis zum Ende dieses letzten Cycle). Die Wartezeit, bis der Task bearbeitet wird, und eventuell anfallende Unterbrechungen seiner Bearbeitung durch andere, höherpriorie Tasks auf Grund der Scheduling-Strategie werden bei der Task-Service-Time mitgezählt und sind daher ein Teil dieser.

Daraus folgt unmittelbar: $\text{Task-Execution-Time} \leq \text{Task-Service-Time}$, wobei "=" nur für den Fall zutrifft, daß der Task unmittelbar nach seiner Ankunft sofort in einem Zug bearbeitet wird (also nicht zu warten braucht und auch nicht mehr unterbrochen wird).

DEADLINE T DER TASK-SERVICE-TIME: Wie bereits im Kapitel 1.1 erwähnt, befassen wir uns hier mit *Echtzeitanwendungen*, d. h., daß es für jeden Task eine genau definierte maximale Antwortzeit gibt, binnen welcher der Task fertig bearbeitet sein muß. Die Deadline T der Task-Service-Time ist genau diese maximale Antwortzeit. Sie ist in unserem Modell für alle Tasks gleich. Sie beginnt mit dem Eintreffen des Tasks beim Computersystem zu laufen.

DEADLINE-VIOLATION: Eine Deadline-Violation (durch einen Task) tritt also genau dann auf, wenn die Task-Service-Time eines Tasks die Deadline T über-

schreitet, wenn also vom Eintreffen des Tasks beim Computersystem bis zur vollständigen Abarbeitung des Tasks mehr Zeit verstreicht als die Deadline T erlaubt.

IDLE-PERIOD: Ist die ununterbrochene Zeit, während welcher das System im Zustand idle ist.

BUSY-PERIOD (BP): Ist die ununterbrochene Zeit, während welcher das System im Zustand busy ist. Offensichtlich besteht also die Zeitachse aus einer Abfolge von Idle-Periods und Busy-Periods.

Eine BP beginnt genau einen Cycle vor dem Cycle, in welchem der erste Cycle des ersten (dieser Busy-Period) zu bearbeitenden Tasks bearbeitet wird (d. h. sie beginnt am Beginn des Idle-Cycle, während dessen ein oder mehrere Tasks beim Computersystem eintreffen), und reicht genau bis zum Ende des letzten Busy-Cycle in ununterbrochener Folge, auf welchen mindestens ein Idle-Cycle folgen muß, nämlich der Beginn der folgenden Idle-Period (wenn es nur genau ein Idle-Cycle ist, welchem gleich wieder Busy-Cycles folgen, so ist dieser eine Idle-Cycle allerdings der Beginn einer neuen BP). Wir nehmen also (aus Gründen der besseren Modellierung und Berechnung) am Beginn der BP auch diesen ersten Idle-Cycle hinzu.

SUB-BUSY-PERIOD (SBP): Eine SBP ist zur Gänze ein Teil einer BP, d. h. sie liegt komplett in einer BP.

Sie beginnt entweder genau am Beginn der BP (wenn sie die erste SBP der BP ist) oder genau am Ende der unmittelbar vorhergehenden SBP. Sie endet genau dann, wenn sich bis auf den soeben bearbeiteten Task kein weiterer mehr in der Task-List befindet und von diesem einen Task gerade der letzte Cycle bearbeitet wird (und zwar endet sie am Beginn dieses letzten Busy-Cycle, er selbst wird nicht mehr zur SBP hinzugezählt).

Sollten während dieses letzten Cycle (welcher ja nicht mehr zu dieser SBP gehört) neue Tasks (einer oder beliebig viele) eintreffen, so wird (gemäß obiger Regel für das Ende der SBP) die aktuelle SBP dennoch beendet und eine neue beginnt (und zwar bereits am Beginn dieses letzten Cycle des zuletzt bearbeiteten Tasks der soeben beendeten SBP). Die BP selbst erfährt hingegen dadurch keine Beeinträchtigung, sie geht weiter bis der Server in den Zustand idle fällt (siehe Erklärung der BP) und umfaßt damit zumindest diese beiden SBP.

Eine BP besteht daher also entweder aus genau einer SBP + dem letzten, nicht mehr zur SBP zählenden Busy-Cycle oder aus mehreren, unmittelbar aufeinander folgenden SBP + dem letzten, nicht mehr zur SBP zählenden Busy-Cycle.

Der Grund für die Einführung der SBP ist, daß man dadurch die Aufgabenstellung noch besser modellieren kann, weil nämlich eine Deadline-Violation nur innerhalb einer SBP auftreten kann, da ja unmittelbar nach dem Ende der SBP soeben der letzte Cycle des letzten Tasks der SBP abgearbeitet wird. Treffen während dieses letzten Cycle des letzten, innerhalb dieser SBP bearbeiteten Tasks wieder weitere Tasks ein, so beginnt mit diesen ja bereits eine neue SBP, wodurch sich die Überwachung der Deadline-Violation abermals auf eine (diese neue) SBP beschränkt und nicht auf die ganze BP.

RUN: Eine Folge von BP (beginnend beim Start des Computersystems), welche die Deadline T der Task-Service-Time nicht verletzen und auf welche eine BP folgt, welche die Deadline T verletzt (durch eine oder mehrere Deadline-Violations), ist ein Run.

SUCCESSFUL-RUN: Wie Run, jedoch ohne die letzte BP, in welcher die Deadline-Violation passiert.

BPD: Die *Busy-Period-Duration* ist genau die Dauer einer BP, gemessen in Cycles. Die BPD ist unabhängig von der Deadline T und auch von der Scheduling-Strategie, wie wir später sehen werden. Sie ergibt sich lediglich aus der Ankunftsrate der Tasks und aus deren Task-Service-Time. Sie gibt daher Auskunft über den "Ladezustand" der Task-List und daher auch des ganzen Computersystems.

SBPD: Die *Sub-Busy-Period-Duration* ist genau die Dauer einer SBP, gemessen in Cycles. Die SBPD ist genauso wie die BPD unabhängig von der Deadline T und auch von der Scheduling-Strategie. Weiters gelten auch alle sonstigen Eigenschaften der BPD für die SBPD.

TTE(T): Die *Time-To-Exceed* ist die Zeitdauer vom ersten Idle-Cycle bis zum Beginn desjenigen Cycle, in welchem die erste Deadline-Violation stattfindet, wo also zum ersten Mal die Deadline T der Task-Service-Time verletzt wird. Damit ist die TTE(T) also genau die Zeit vom Start des Computersystems bis zum ersten "schweren Fehler", wo die vorgeschriebene maximale Antwortzeit nicht mehr eingehalten wird.

SRD(T): Die *Successful-Run-Duration* ist genau die Dauer eines Successful-Run, gemessen in Cycles.

Da ja ein Successful-Run die BP, bei welcher die Deadline-Violation passiert, nicht mehr enthält, während die $TTE(T)$ ihr Ende in der BP hat, in der die Deadline verletzt wird (nämlich genau am Beginn des Cycle, in dem die Deadline verletzt wird), gilt offensichtlich folgende Beziehung: $SRD(T) < TTE(T)$. In diesem Papier werden wir uns auf die Untersuchung der $SRD(T)$ beschränken.

REGULAR-CASE: Ist dann der Fall, wenn die Task-Ankunftsrate gering ist, genauer wenn die Cycle-Ankunftsrate kleiner ist als die Cycle-Abarbeitungsrate. Siehe dazu Kapitel 2.3. Zu diesem Fall gibt es bereits einige Abhandlungen, siehe z. B. [16], [3] und [17].

BALANCED-CASE: Ist dann der Fall, wenn die Task-Ankunftsrate mittel ist, genauer wenn die Cycle-Ankunftsrate gleich ist mit der Cycle-Abarbeitungsrate. Siehe dazu Kapitel 2.3.

RUSH-HOUR-CASE: (auch: Rush-Hour-Condition) Ist dann der Fall, wenn die Task-Ankunftsrate hoch ist, genauer wenn die Cycle-Ankunftsrate größer ist als die Cycle-Abarbeitungsrate. Siehe dazu Kapitel 2.3.

FCFS, FCFSR: FCFS (First Come First Serve) ist eine Scheduling-Strategie nach dem Motto “wer zuerst kommt, mahlt zuerst”. Die neu ankommenden Tasks werden in der Reihenfolge ihrer Ankunft auf der einen Seite in die Task-List eingetragen und auf der anderen Seite in der gleichen Reihenfolge vom Server bearbeitet.

Das angefügte ‘r’ bedeutet, daß es sich um einen Rush-Hour-Case handelt.

PLCFS, PLCFSR: Diese Scheduling-Strategie pLCFS (preemptive Last Come First Serve) arbeitet ganz strikt nach dem Motto “die Letzten werden die Ersten sein”, d. h., daß Tasks, die zuletzt eintreffen, sofort — also sogar durch Unterbrechung des momentan bearbeiteten Tasks — im nächsten Cycle vom Server bearbeitet werden. Der hierdurch unterbrochene Task wird danach fortgesetzt. Diese Strategie wird strikt angewendet, womit unter Umständen (bei immer wieder neuem Eintreffen weiterer Tasks) immer wieder der soeben eingetroffene Task den soeben laufenden unterbricht, aber seinerseits nach nur kurzer Bearbeitungsdauer selbst unterbrochen wird,

weil bereits wieder ein neuer Task eintrifft, welcher natürlich seinerseits den nun laufenden unterbricht.

Wenn also genau ein Task eintrifft, startet seine Bearbeitung im nächsten Cycle, wenn mehrere Tasks während eines Cycle eintreffen, wird einer von diesen im nächsten Cycle zu bearbeiten begonnen, die anderen kommen unmittelbar danach der Reihe nach dran. Da ja solche während eines Cycle eintreffenden Tasks erst am Ende des Cycle als eingetroffen erkannt werden, sind sie aus Sicht des Schedulers alle gleichzeitig eingetroffen. In welcher Reihenfolge diese Tasks daher bearbeitet werden, hat nichts mit der Scheduling-Strategie selbst zu tun, sondern kann unabhängig davon gewählt werden. Vereinbaren wir also, daß in diesem Falle die Tasks einfach mit aufsteigender Numerierung bearbeitet werden. Siehe dazu auch Abbildung 1.2.

Das angefügte 'r' bedeutet, daß es sich um einen Rush-Hour-Case handelt.

nLCFS, nLCFSr: Die Scheduling-Strategie nLCFS (nonpreemptive Last Come First Serve) arbeitet ebenso wie pLCFS nach dem Motto "die Letzten werden die Ersten sein", aber nicht ganz so strikt, d. h., daß Tasks, die zuletzt eintreffen, bei der nächsten Gelegenheit abgearbeitet werden, ohne jedoch den derzeit in Arbeit befindlichen Task zu unterbrechen.

Wenn also genau ein Task während der Bearbeitung eines Tasks eintrifft, startet seine Bearbeitung unmittelbar nach dem letzten Cycle des derzeit bearbeiteten Tasks. Treffen mehrere Tasks während der Bearbeitung eines Tasks ein, so wird der derzeit bearbeitete natürlich fertig bearbeitet und danach der zuletzt eingetroffene gestartet; wenn dieser fertig ist, der vorletzte Task und so weiter.

Wenn während eines Cycle mehrere Tasks eintreffen, so gelten diese als gleichzeitig eingetroffen (siehe Beschreibung von pLCFS/pLCFSr) und es gilt bezüglich der Bearbeitungsreihenfolge dieser Tasks dasselbe wie bei pLCFS/pLCFSr. Siehe dazu auch Abbildung 1.3.

Das angefügte 'r' bedeutet, daß es sich um einen Rush-Hour-Case handelt.

Die Abbildungen 1.1, 1.2 und 1.3 veranschaulichen die soeben erwähnten Begriffe und zeigen auch die Unterschiede auf, wie die Tasks, abhängig von der jeweiligen Scheduling-Strategie, abgearbeitet werden. Hier eine Erläuterung dieser Abbildungen:

Die Abszisse gibt das Fortschreiten der Zeit (Einheit: 1 Cycle) an, auf der Ordinate wird die in der Task-List befindliche Anzahl von Tasks (inklusive dem gerade bearbeiteten) aufgetragen. Unter der Abszisse sind die Ankünfte der Tasks mittels kleiner Pfeilchen in ihrer zeitlichen Reihenfolge markiert. Um den Unterschied der Scheduling-Strategien leicht zu erkennen, wurden bei allen drei Beispielen die Task-Ankünfte gleich gewählt. Über den waagrechten Linien in der Abbildung steht immer der Task, welcher gerade bearbeitet wird. Die Länge der waagrechten Linie gibt die Zeit an, wie lange der Task (oder ein Teil davon) vom Server bearbeitet wird, die Summe dieser Längen pro Task ergibt genau dessen Task-Execution-Time. Die Task-Execution-Time der Tasks ist hier sehr klein gewählt, um genügend Information in der Abbildung unterzubringen, und nur symbolisch zu verstehen.

Man beachte insbesondere den senkrechten Strich in der Abbildung 1.1 zwischen dem Task T_1 und dem Task T_2 bzw. zwischen T_8 und T_9 , in der Abbildung 1.2 zwischen den Tasks T_7 und T_9 und in der Abbildung 1.3 zwischen T_1 und T_4 bzw. T_7 und T_9 : Dieser macht deutlich, daß sich soeben ein Task beendet hat und während des letzten Cycle dieses Tasks ein (oder mehrere) neuer Task eingetroffen ist. Reicht dieser senkrechte Strich in der Abbildung bis zur Abszisse hinunter (in der Abbildung 1.1 zwischen T_8 und T_9 , in der Abbildung 1.2 zwischen den Tasks T_7 und T_9 und in der Abbildung 1.3 zwischen T_7 und T_9), so ergibt sich dadurch das Ende einer SBP — genauer: die SBP endet per definitionem am Beginn dieses letzten Cycle, die neue SBP beginnt eben dort.

Weiters sieht man in den Abbildungen auch die bei der Begriffsbestimmung für pLCFS und nLCFS erwähnte Bearbeitungsreihenfolge von Tasks, welche während eines Cycle eintreffen, sehr schön, nämlich daß sie, weil sie ja ohnehin als gleichzeitig eingetroffen gelten, einfach aufsteigend numeriert bearbeitet werden.

In den Abbildungen sind auch jeweils die BP und SBP eingezeichnet. Auch hier wird deutlich, daß die BP und SBP unabhängig von der Scheduling-Strategie sind (was man später bei der Berechnung noch eindeutig erkennen wird), weil sie für alle drei Abbildungen jeweils gleich sind mit jeder der anderen Abbildungen, sie hängen hingegen nur von der Anzahl der Cycles ab, die in einem gewissen Intervall eintreffen, mit anderen Worten von der Cycle-Ankunftsrate. Die “Gebirge” (d. h. die jeweiligen Graphen) sind natürlich unterschiedlich, weil sie im Unterschied dazu die Scheduling-Strategie widerspiegeln.

Nehmen wir beispielhaft eine Task-Service-Time Deadline T von 7 Cycles an, so ergeben sich die in den Abbildungen gezeigten Deadline-Violations, $\text{SRD}(T)$

und $TTE(T)$.

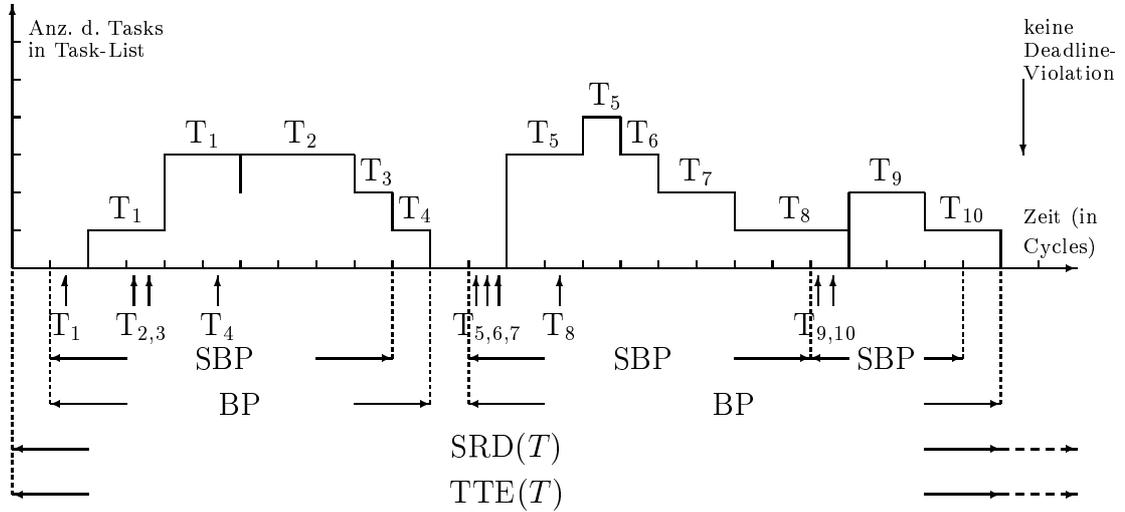


Abbildung 1.1: FCFS: Tasks in der Task-List

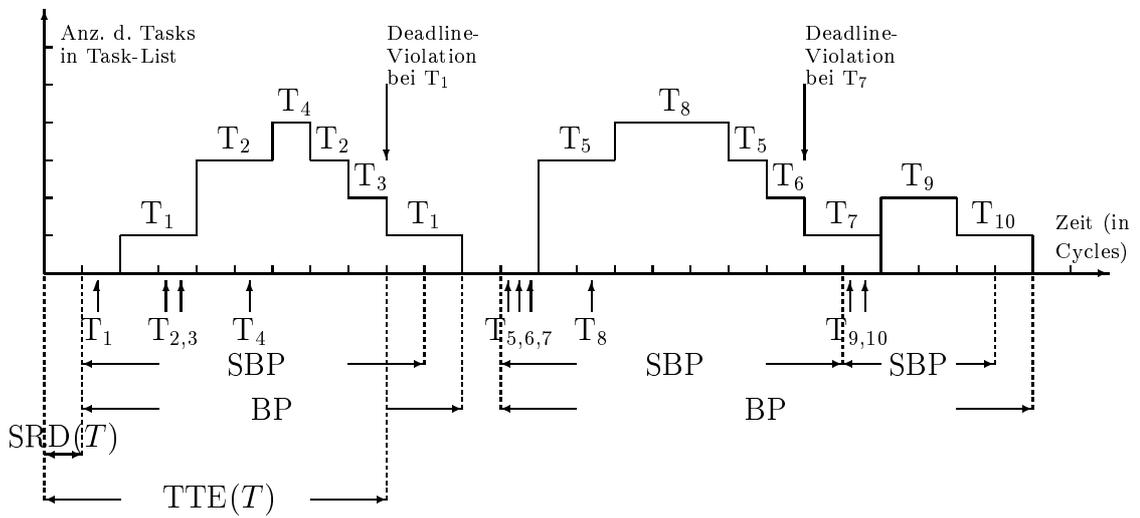


Abbildung 1.2: pLCFS: Tasks in der Task-List

Weiters möchte ich an dieser Stelle die folgenden bekannten Notationen nochmals in Erinnerung rufen, welche wir bei den Berechnungen in diesem Papier des

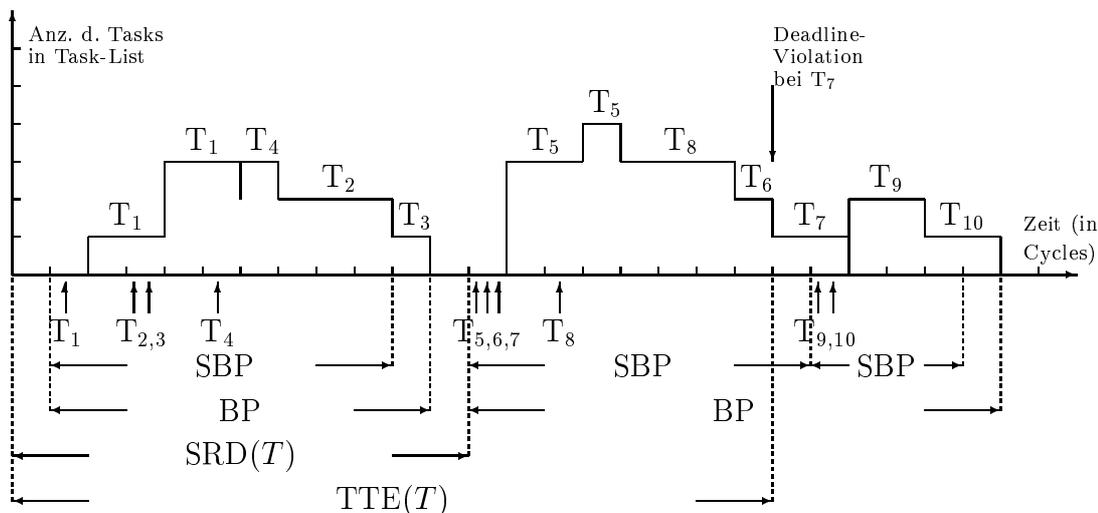


Abbildung 1.3: nLCFS: Tasks in der Task-List

öfteren verwenden werden:

$$f(x) = O(g(x)) \quad \text{für } x \rightarrow x_0, \quad \text{wenn } |f(x)|/|g(x)| \leq C \text{ für eine reelle Konstante } C > 0 \text{ unabhängig von } x, \forall x \text{ in einer passenden Umgebung um } x_0;$$

$$f(x) = o(g(x)) \quad \text{für } x \rightarrow x_0, \quad \text{wenn } \lim_{x \rightarrow x_0} f(x)/g(x) = 0;$$

$$f(x) \sim g(x) \quad \text{für } x \rightarrow x_0, \quad \text{wenn } \lim_{x \rightarrow x_0} f(x)/g(x) = 1;$$

$$[z^n]f(z) \quad \dots \quad \text{ist der } n\text{-te Koeffizient in der formalen Potenzreihe } f(z).$$

Diese Notationen sind auch gültig für $x_0 \rightarrow \infty$, d. h. für $x \rightarrow \infty$.

1.3 Vorschau

Vorweg sei hier ein kurzer Überblick über den Aufbau dieses hier vorliegenden Papiers gegeben:

Im Kapitel 1 werden die Aufgabe dargestellt und die verwendeten Begriffe erklärt. Im Kapitel 2 werden die benötigten Grundlagen der Statistik zusammengefasst und wird konkret auf den Rush-Hour-Case und dessen Eigenschaften eingegangen.

Im Kapitel 3 wird erstmals das Baum-Modell, welches ja die Grundlage des Lösungsweges bildet, genau vorgestellt, während in den folgenden Kapiteln 4 und 5 die BPD und die SBPD genau untersucht werden.

Sodann widmet sich das Kapitel 6 genauer der BPD und der SBPD unter Berücksichtigung der Deadline T und geht auch auf diverse Grenzwerte genauer ein.

Das Kapitel 7 liefert die gesuchten Ergebnisse bezüglich der $SRD(T)$, während im abschließenden Kapitel 8 noch ein Beispiel und diverse Schlußbemerkungen folgen.

Teil II

Wahrscheinlichkeits-Modell und Baum-Modell

Kapitel 2

WAHRSCHEINLICHKEITS-MODELL

Nach den Vorbetrachtungen im Kapitel 1 werde ich im hiesigen Kapitel das Wahrscheinlichkeits-Modell für die späteren Berechnungen aufstellen.

2.1 Grundzüge der Wahrscheinlichkeitstheorie

Da in diesem Papier im Endeffekt Aussagen über die Momente von Zufallsvariablen (Mittelwert, Varianz) gemacht werden sollen, seien an dieser Stelle die dafür notwendigen Grundzüge erläutert — allerdings ohne genauer darauf einzugehen. Über diese Erläuterungen hinaus- und tiefergehende Abhandlungen über die Wahrscheinlichkeitstheorie möge man bitte in der entsprechenden Literatur nachlesen.

ZUFALLSVARIABLE: Eine Zufallsvariable X ist ein Ereignis, welches verschiedene Merkmale (oder Werte) annehmen kann. Gibt es höchstens abzählbar viele Merkmale für X , so spricht man von einer *diskreten* Zufallsvariablen, ansonsten von einer *kontinuierlichen*.

Betrachtet man z.B. die Anzahl der Task-Ankünfte pro Cycle, die Task-Länge (= Task-Execution-Time) oder die Anzahl von Cycle-Ankünften pro Cycle, so sind dies typische Vertreter von diskreten Zufallsvariablen. Im Folgenden werden wir uns daher auch nur solchen widmen.

WAHRSCHEINLICHKEITSVERTEILUNG: Je nachdem, wie und wo die Zufallsvariable ihre Werte annimmt, kann man sie in verschiedene Kategorien einteilen, d.h. sie folgt einer gewissen *Wahrscheinlichkeitsverteilung*. *Diskrete* Verteilungen (also solche von diskreten Zufallsvariablen) sind z.B. die Alternativ-Verteilung, die Binomial-Verteilung, die Hypergeometrische-Verteilung, die Geometrische-Verteilung oder die Poisson-Verteilung, *kontinuierliche* Verteilungen (also solche von kontinuierlichen Zufallsvariablen) sind z.B. die Exponential-Verteilung, die Gamma-Verteilung, die Chi-Quadrat-Verteilung oder die (Gauß'sche-)Normal-Verteilung.

Als eine mögliche und gut passende Wahrscheinlichkeitsverteilung für die Task-Ankünfte und die Task-Execution-Time kann man z. B. die Poisson-Verteilung annehmen.

Die Verteilung wird entweder in Form der *Dichtefunktion* ($f(X) = W\{X = x\}$) oder in Form der *Verteilungsfunktion* ($F(X) = W\{X \leq x\}$) angegeben, wobei hier $W\{\text{Ereignis}\}$ die Wahrscheinlichkeit, daß ‘Ereignis’ eintritt, bedeutet.

WAHRSCHEINLICHKEITSERZEUGENDE FUNKTION (PGF): (englisch: Probability Generating Function) Die PGF $P(z)$ einer Zufallsvariable wird definiert durch

$$\begin{aligned} P(z) &= p_0 + p_1z + p_2z^2 + p_3z^3 + \dots = \\ &= \sum_{k \geq 0} p_k z^k, \end{aligned} \quad (2.1)$$

wobei p_i die Wahrscheinlichkeit ist, daß die Zufallsvariable den Wert i annimmt.

Selbstredend ist die Summe *aller* möglichen Wahrscheinlichkeiten für eine Zufallsvariable gleich mit der Wahrscheinlichkeit “sicher”.

Wenn die Zufallsvariable nur die endlichen Werte $0, 1, 2, 3, \dots$ annehmen kann (oder exakter: Wenn es eine Abbildung der Werte der Zufallsvariablen auf den Zahlenbereich \mathbb{N}_0 gibt), dann gilt somit

$$\begin{aligned} P(1) &= p_0 + p_1 + p_2 + p_3 + \dots = \\ &= \sum_{k \geq 0} p_k = 1. \end{aligned} \quad (2.2)$$

Dies entspricht also der Wahrscheinlichkeit, daß die Zufallsvariable endliche Werte annimmt.

Kann hingegen die Zufallsvariable auch noch andere Werte annehmen, wie z. B. im konkreten Fall des Rush-Hour-Case, bei dem z. B. die BPD auch unendlich lang sein kann, so muß man obige Gleichung 2.2 folgendermaßen exakter formulieren:

$$P(1) = \sum_{k \geq 0} p_k \leq 1,$$

oder auch:

$$P(1) = \sum_{k \geq 0} p_k = 1 - p_\infty, \quad (2.3)$$

wobei p_∞ die Wahrscheinlichkeit ist, daß die Zufallsvariable den Wert “ ∞ ” annimmt (und $0 \leq p_\infty \leq 1$ gilt), d. h. daß sie nicht endlich ist. Achtung: p_∞ ist jedenfalls nicht der $\lim_{k \rightarrow \infty} p_k$, sondern eben ein Symbol mit der soeben genannten Bedeutung. Demzufolge entspricht $P(1)$ der Wahrscheinlichkeit, daß die Zufallsvariable endliche Werte annimmt.

Die PGF wird oft verwendet, um Aussagen über eine Zufallsvariable zu treffen, deren Einzelwahrscheinlichkeiten nicht bekannt oder nicht exakt bestimmbar sind — speziell im hier vorliegenden Dokument wird sie ein wesentliches Element des Problem-Lösungsweges sein.

MITTELWERT: Der Mittelwert μ oder auch Erwartungswert $E[X]$ der Zufallsvariablen X ist definiert durch

$$\begin{aligned} E[X] = \mu = P'(1) &= [P'(z)]_{z=1} = \\ &= \left[\sum_{k \geq 1} k p_k z^{k-1} \right]_{z=1} = \\ &= \sum_{k \geq 0} k p_k. \end{aligned} \quad (2.4)$$

VARIANZ: Die Varianz $Var[X]$ der Zufallsvariablen X ist die mittlere quadratische Abweichung der Zufallsvariablen vom Mittelwert μ , also

$$\begin{aligned} Var[X] &= E[(X - \mu)^2] = E[X^2 - 2X\mu + \mu^2] = \\ &= E[X^2] - 2E[X]\mu + \mu^2 = E[X^2] - 2\mu\mu + \mu^2 = \\ &= E[X^2] - \mu^2 = E[X^2] - (E[X])^2. \end{aligned} \quad (2.5)$$

Andererseits ist

$$\begin{aligned} P''(1) &= [P''(z)]_{z=1} = \\ &= \left[\sum_{k \geq 2} k(k-1)p_k z^{k-2} \right]_{z=1} = \\ &= \sum_{k \geq 0} k(k-1)p_k = \\ &= \sum_{k \geq 0} k^2 p_k - \sum_{k \geq 0} k p_k = \\ &= E[X^2] - \mu = E[X^2] - E[X]. \end{aligned}$$

Daraus ergibt sich folgende Beziehung zwischen der Varianz und den Ableitungen der PGF:

$$Var[X] = E[X^2] - \mu^2 =$$

$$\begin{aligned}
&= (P''(1) + \mu) - \mu^2 = \\
&= P''(1) + P'(1) - (P'(1))^2.
\end{aligned} \tag{2.6}$$

MOMENTE: Das n -te Moment der Zufallsvariablen X ist der Mittelwert von X^n , d. h. also

$$n\text{-tes Moment} = E[X^n] = \sum_{k \geq 0} k^n p_k.$$

z. B. ist der Mittelwert von X das 1. Moment ($= E[X]$), die Varianz ist eine Kombination aus 1. und 2. Moment ($= E[X^2] - (E[X])^2$, siehe auch Gleichung 2.5).

2.2 Definitionen: $A(z)$, $L(z)$, $P(z)$

Wir nehmen an, daß die Anzahl der Task-Ankünfte innerhalb eines Cycle einer beliebigen Wahrscheinlichkeitsverteilung folgt, jedenfalls unabhängig von der Anzahl der Task-Ankünfte im vorherigen oder im folgenden Cycle. Ebenso soll die Anzahl von Cycles pro Task (also die Task-Execution-Time) unabhängig davon einer beliebigen Wahrscheinlichkeitsverteilung folgen. Wollen wir nun die PGF einiger notwendiger Größen näher betrachten:

2.2.1 Anzahl der Task-Ankünfte während eines Cycle: $A(z)$

Sei

$A(z)$... die PGF der Task-Ankünfte während eines Cycle,

a_k ... die Wahrscheinlichkeit, daß während eines Cycle k Tasks eintreffen,

dann ist

$$A(z) = \sum_{k \geq 0} a_k z^k. \tag{2.7}$$

Aussagen über $A(z)$:

Wir treffen folgende Annahme:

$$a_0 = A(0) > 0,$$

d. h. also, daß die Wahrscheinlichkeit, daß kein Task während eines Cycle eintrifft, > 0 ist, oder mit anderen Worten, daß es Cycles geben kann, während derer kein Task eintrifft.

2.2.2 Task-Execution-Time (= Task-Länge): $L(z)$

Sei

$L(z)$... die PGF der Task-Länge eines Tasks,

l_k ... die Wahrscheinlichkeit, daß die Task-Länge (die Task-Execution-Time) k Cycles beträgt,

dann ist

$$L(z) = \sum_{k \geq 0} l_k z^k. \quad (2.8)$$

Aussagen über $L(z)$:

Wir treffen folgende Annahme (o. B. d. A.):

$$l_0 = L(0) = 0,$$

d. h. also, daß die Wahrscheinlichkeit, daß ein Task die Länge 0 habe, gleich 0 sei, oder mit anderen Worten, daß jeder Task mindestens die Länge 1 habe — was ja keine Einschränkung ist, sondern unserem bis dato aufgestellten Modell voll entspricht.

2.2.3 Anzahl der Cycle-Ankünfte während eines Cycle: $P(z)$

Sei

$P(z)$... die PGF der Cycle-Ankünfte während eines Cycle,

p_k ... die Wahrscheinlichkeit, daß während eines Cycle k neue Cycles eintreffen,

dann ist

$$P(z) = \sum_{k \geq 0} p_k z^k. \quad (2.9)$$

2.3 Aussagen über $P(z)$

Nun wollen wir uns genauer der PGF der Cycle-Ankünfte widmen, um dadurch die Grundlagen für die späteren Berechnungen zu legen.

2.3.1 Herleitung von $P(z)$ aus $A(z)$ und $L(z)$

Während eines Cycles treffen k neue Cycles genau dann ein, wenn entweder genau ein Task mit k Cycles eintrifft, oder zwei Tasks, deren Summe der Task-Längen k Cycles beträgt, oder allgemeiner wenn i Tasks eintreffen mit einer Summe der Task-Längen von k . Daher ergibt sich daraus für die entsprechenden Wahrscheinlichkeiten folgender Zusammenhang (für $k \geq 1$):

$$\begin{aligned}
 p_k &= a_1 \cdot l_k + \\
 &\quad + a_2 \cdot \sum_{k_1+k_2=k} l_{k_1} l_{k_2} + \\
 &\quad + \dots + \\
 &\quad + a_i \cdot \sum_{\sum k_j=k} \prod_{j=1}^i l_{k_j} + \\
 &\quad + \dots = \\
 &= \sum_{i \geq 1} \left(a_i \cdot \sum_{\sum k_j=k} \prod_{j=1}^i l_{k_j} \right) = \\
 &= [z^k] A(L(z)).
 \end{aligned}$$

Zusammen mit der Erkenntnis, daß $p_0 = a_0$ (womit der Fall $k = 0$ auch abgedeckt ist), weil ja 0 neue Cycles nur genau dann eintreffen können, wenn auch kein Task eintrifft, ergibt dies zusammenfassend:

$$P(z) = A(L(z)). \quad (2.10)$$

2.3.2 Weitere Aussagen über $P(z)$ im Rush-Hour-Case

Weiters gelten noch folgende offensichtliche Aussagen bzw. werden für den Rush-Hour-Case folgende Annahmen getroffen:

$$P(0) = p_0 = A(L(0)) = A(0) = a_0 > 0 \quad (2.11)$$

$$P(1) = \sum_{k \geq 0} p_k = 1 \quad (2.12)$$

$$P'(1) = \sum_{k \geq 0} p_k k > 1 \quad (2.13)$$

$$P''(z) \neq 0 \quad (2.14)$$

$$R_P > 1 \quad (R_P \dots \text{Konvergenzradius von } P(z)). \quad (2.15)$$

Gleichung 2.11 ist keine weitere Einschränkung sondern eine Folge der Gleichungen 2.9 und 2.10 zusammen mit Gleichung 2.7 und Gleichung 2.8 und deren jeweiliger Annahme bezüglich a_0 bzw. l_0 .

Gleichung 2.12 bedeutet, daß o. B. d. A. angenommen wird, daß nur endlich viele Cycles während eines Cycles eintreffen können, weil ja nur dies sinnvoll ist und zu sinnvollen Ergebnissen führt. Siehe dazu einerseits die Definition von $P(z)$ in Gleichung 2.9 und andererseits die Erklärungen bezüglich der Wahrscheinlichkeit von nicht endlichen Werten von Zufallsvariablen im Kapitel 2.1 bei den Gleichungen 2.1 bis 2.3.

Gleichung 2.13 bedeutet, daß die mittlere Cycle-Ankunftsrate ($= P'(1)$) größer sei als die mittlere Abarbeitungsrate ($= 1$, weil ja genau ein Cycle pro Cycle abgearbeitet wird), d. h. eben exakt, daß es sich um einen Rush-Hour-Case handelt. Bemerkung: Wenn $P'(1) < 1$ ist dies der Regular-Case, wenn $P'(1) = 1$ ist dies der Balanced-Case.

Gleichung 2.14 bedeutet, daß der triviale Fall $P''(z) \equiv 0$, also $P(z) = p_0 + (1 - p_0) \cdot z$ (d. h. $p_1 = 1 - p_0$, $p_2 = p_3 = \dots = 0$), im Rush-Hour-Case (und auch im Balanced-Case) gar nicht möglich ist, weil hierbei ja ansonsten $P'(z) = 1 - p_0$ wäre, was im Zusammenwirken mit Gleichung 2.11 hieße, daß $P'(z)$ und damit auch $P'(1) < 1$ sein müßte, was ja ein Widerspruch zu Gleichung 2.13 wäre.

Gleichung 2.15 fordert, daß der Konvergenzradius von $P(z)$ groß genug sei, um unsere Berechnungen zu unterstützen.

Satz 1 *Aus der Taylorreihen-Entwicklung, geometrischen Überlegungen und dem Rouchè-Theorem folgt: $P(z)$ hat zwei Fixpunkte β (mit $0 < \beta < 1$) und 1, bzw. $P(z) - z$ hat eben diese zwei Nullstellen. ■*

Die Abbildung 2.1 illustriert dies für $x \in \mathbb{R}$.

BEWEIS: (skizziert) Betrachten wir vorerst $P(x)$, $x \in \mathbb{R}$. Es gibt offensichtlich einen Fixpunkt $x = 1$. Wenn wir $P(x)$ mittels Taylorreihenentwicklung an der Anschlußstelle 1 entwickeln (was ja wegen Gleichung 2.15 möglich sein muß), ergibt dies

$$\begin{aligned} f(x) = P(x) - x &= (P(1) - 1) + (P'(1) - 1)(x - 1) + R_2(x) = \\ &= (P'(1) - 1)(x - 1) + R_2(x). \end{aligned}$$

Da $P(0) = p_0 (= a_0) > 0$ (siehe Gleichung 2.11) und $P(1) = 1$, folgt: $f(0) = p_0 (= a_0) > 0$ und $f(1) = 0$. Außerdem folgt aus obiger Taylorreihenentwicklung, daß für ein genügend kleines $\epsilon (> 0)$

$$f(x) < 0 \quad \text{für } x \in (1 - \epsilon; 1),$$

weil ja wegen Gleichung 2.13 $(P'(1) - 1) > 0$ und $(x - 1) < 0$ ist; d. h. also, daß es im Intervall zwischen 0 und 1 mindestens eine Nullstelle von $f(x)$ geben muß, von denen wir die kleinste davon β nennen. Weiters gilt, daß $P(x)$ für $x \geq 0$ streng monoton wachsend ist, weil ja $P'(x) > 0$ sein muß (siehe Gleichung 2.13 im Zusammenwirken mit der Tatsache, daß ja alle $p_k \geq 0$ und $k \geq 0$ sind für alle $k \in \mathbb{N}_0$, und Gleichung 2.14 und die Erläuterungen zu dieser).

Durch das Theorem von Rouchè (siehe z. B. [16]) läßt sich (nun im Bereich der komplexen Zahlen, also $z \in \mathbb{C}$) leicht zeigen, daß es nur genau eine Nullstelle innerhalb des Kreises um den Koordinatenursprung mit Radius 1 gibt und genau eine Nullstelle am Kreis selbst. Aus einfachen geometrischen Betrachtungen erkennt man, daß die eine Nullstelle innerhalb des Kreises (nämlich β) eine einfache Nullstelle ist, weil $P'(\beta) < 1$.

Da wir diese beiden Nullstellen bereits auf der reellen Zahlengerade gefunden haben (nämlich β und 1), ist damit der Satz bewiesen. ■

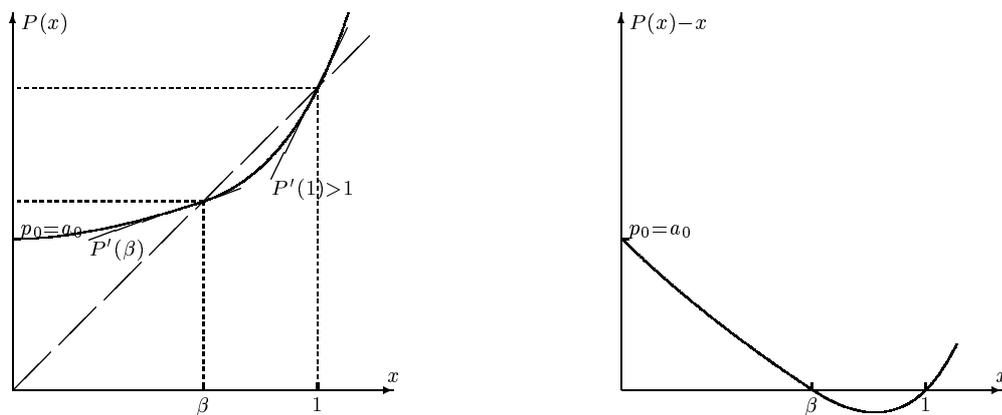


Abbildung 2.1: $P(x)$ und $P(x) - x$

2.3.3 Weitere Aussagen über $P(x)$, x reell, im Rush-Hour-Case

Im vorigen Kapitel 2.3.2 wurde $P(z)$ (mit $z \in \mathbb{C}$) bereits untersucht und einige Erkenntnisse darüber festgeschrieben. Aus diesen und aus der Abbildung 2.1 ergeben sich weitere Eigenschaften von $P(x)$ im reellen Zahlenbereich (also für $x \in \mathbb{R}$), welche wir hier anführen und in der Folge beweisen wollen, weil sie später noch mehrmals bei der Berechnung von Schranken diverser Größen von Bedeutung sein werden.

Satz 2 Für $k \in \mathbb{N}_0$, $x \in \mathbb{R}^+$ (also x reell und > 0) und $x < R_P$, $p_k \geq 0$ (und $p_k \leq 1$, weil p_k ja eine Wahrscheinlichkeit repräsentiert) und unter Berücksichtigung der Gleichung 2.14, welche ja unter anderem besagt, daß zumindest ein p_i , mit $i \geq 2$, > 0 sein muß, gilt:

1. $P(x)$ ist positiv,
2. $P(x)$ ist streng monoton wachsend, und
3. $P(x)$ ist positiv (also nach links) gekrümmt.

■

BEWEIS: Wenn man dies als (Un-)Gleichungen formuliert, erhält man (unter Beibehaltung der Bedeutung und Einschränkungen der beteiligten Ausdrücke wie im Satz 2):

$$P(x) = \sum_{k \geq 0} p_k x^k > 0 \quad (2.16)$$

$$P'(x) = \sum_{k \geq 1} p_k k x^{k-1} > 0 \quad (2.17)$$

$$P''(x) = \sum_{k \geq 2} p_k k(k-1)x^{k-2} > 0 \quad (2.18)$$

Da in allen drei (Un-)Gleichungen alle darin vorkommenden Ausdrücke jedenfalls ≥ 0 sind, muß natürlich auch jedes Produkt davon und daher auch jede Summe dieser Produkte ≥ 0 sein. Somit ist also für alle drei (Un-)Gleichungen bereits “ ≥ 0 ” bewiesen.

Da der Satz 2 seine Aussagen auf positive, reelle x beschränkt ($x \in \mathbb{R}^+$), müssen daher die in den (Un-)Gleichungen 2.16, 2.17 und 2.18 vorkommenden Ausdrücke “ x^i ” jedenfalls > 0 sein. Ebenso sind alle darin als Multiplikand vorkommenden Ausdrücke “ k ” > 0 . Weiters muß gemäß der Einschränkung im Satz 2 mindestens ein p_i , mit $i \geq 2$, > 0 sein, woraus folgt, daß genau dieses Produkt, welches eben dieses p_i enthält, > 0 sein muß. Daher ist die Summe aus zumindest einem Summanden, welcher > 0 ist, und den anderen Summanden, welche alle ≥ 0 sind, jedenfalls auch > 0 , womit der Satz bewiesen ist. ■

2.3.4 Schranken von $P'(\beta)$ im Rush-Hour-Case

Aus Satz 2 im Kapitel 2.3.3 wissen wir bereits, daß $P(x)$ für $x \in \mathbb{R}^+$ (innerhalb des Konvergenzradius R_P von $P(z)$, welcher ja in Gleichung 2.15 im Kapitel 2.3.2

> 1 angenommen wurde) streng monoton wachsend und nach links gekrümmt ist. Daraus und aus geometrischen Überlegungen, welche man leicht an Hand der Abbildung 2.1 nachvollziehen kann, kann man für die Schranken von $P'(\beta)$ folgende Beziehungen erkennen:

Einerseits muß die Steigung der Tangente im Punkt $(\beta; \beta)$ (also $P'(\beta)$) größer sein als die Steigung der Sekante vom Punkt $(0; a_0)$ zum Punkt $(\beta; \beta)$, andererseits muß sie aber kleiner sein als die Steigung der Sekante vom Punkt $(\beta; \beta)$ zum Punkt $(1; 1)$. Daher erhält man unter Berücksichtigung der Gleichungen 2.11 ($P(0) = a_0$) und 2.12 ($P(1) = 1$) und des Satzes 1 ($P(\beta) = \beta$)

$$\begin{aligned} \frac{P(\beta) - P(0)}{\beta - 0} &< P'(\beta) < \frac{P(1) - P(\beta)}{1 - \beta} \\ \frac{\beta - a_0}{\beta} &< P'(\beta) < \frac{1 - \beta}{1 - \beta} \\ 1 - \frac{a_0}{\beta} &< P'(\beta) < 1. \end{aligned}$$

Da ja $a_0 < \beta$ ist, ist $\frac{a_0}{\beta} < 1$ und daher $1 - \frac{a_0}{\beta}$ jedenfalls > 0 (und natürlich auch < 1 , weil ja sowohl a_0 als auch β positiv sind) und damit eine sinnvolle untere Schranke.

Somit haben wir also folgende Schranken für $P'(\beta)$ erhalten:

$$1 - \frac{a_0}{\beta} < P'(\beta) < 1. \quad (2.19)$$

Diese Schranken werden in späteren Kapiteln bei der Berechnung der Schranken anderer Größen noch mehrmals verwendet werden.

Kapitel 3

MODELLIERUNG DER ANKOMMENDEN TASKS MITTELS EINES BAUM-MODELLS

Nachdem im Kapitel 1.2 die Task-Ankünfte graphisch dargestellt wurden, kommen wir nun an eine zentrale Stelle unserer Abhandlung. Hier werden nun die Task-Ankünfte und die Task-Länge eineindeutig auf die Familie der ebenen Wurzelbäume abgebildet. Dadurch kann man einerseits auch die Deadline T der Task-Service-Time auf die Weite oder die Größe von Bäumen abbilden (abhängig von der Scheduling-Strategie, siehe im entsprechenden Kapitel 3.1, 3.2 oder 3.3), andererseits kann man danach die erzeugende Funktion (englisch: *Ordinary Generating Function* OGF) dieser Familie von Bäumen in die PGF von dazupassenden Zufallsvariablen übersetzen (siehe z. B. Kapitel 3.4), sodaß man dann bequem mit diesen weiterrechnen kann.

KONSTRUKTION EINES BAUMES: Ein Task wird symbolisiert durch einen quadratischen Knoten (\square), ein Cycle durch einen runden Knoten (\circ).

Die Task-Länge eines Tasks wird durch die Anzahl der Nachfolgeknoten (lauter Cycles) bestimmt und mit der Wahrscheinlichkeit für diese Nachfolgeknoten (l_i ist die Wahrscheinlichkeit, daß die Task-Länge i beträgt) bewertet.

Treffen während der Bearbeitung eines Cycle i Tasks ein, so erhält der Cycle i Nachfolgeknoten (lauter Tasks) und wird mit der Wahrscheinlichkeit für diese Nachfolgeknoten (a_i ist die Wahrscheinlichkeit, daß i Tasks eintreffen während eines Cycle) bewertet.

Die Wurzel eines solchen Baumes ist daher immer der erste Idle-Cycle, mit welchem ja immer eine Busy-Period (BP) eingeleitet wird. Ein Idle-Cycle, in welchem keine Tasks ankommen, ist Teil einer Idle-Period und wird repräsentiert durch einen einzigen runden Knoten (\circ) (durch einen degenerierten Baum sozusagen).

Ein derart konstruierter Baum stellt hiermit also genau eine BP dar. Der Beginn einer neuen SBP ist bekanntlich genau dann, wenn der letzte Cycle

eines Tasks bearbeitet wird und sich kein sonstiger Task mehr in der Task-List befindet. Es wird sich herausstellen, daß sowohl die SBP als auch die BP unabhängig von der Scheduling-Strategie sind und beide nur von der Anzahl und Rate der ankommenden Cycles abhängig sind.

Aus diesem Konstruktionsprinzip und der Abarbeitungsreihenfolge ergibt sich somit die eindeutige Abbildung von den Task- und Cycle-Ankünften auf die Familie der ebenen Wurzelbäume.

Die Abarbeitungsreihenfolge und die Deadline-Violation selbst sind natürlich Scheduling-Strategie-spezifisch, ebenso ergeben sich bei gleicher Task-Ankunft zwar verschiedene Bäume, was in Summe aber auf die Länge von BP und SBP keinen Einfluß hat. Die Unterschiede in der Abarbeitungsreihenfolge zwischen den verschiedenen Scheduling-Strategien und auch jeweils ein anschauliches Beispiel siehe in den nun folgenden Kapiteln 3.1, 3.2 und 3.3.

3.1 Baum-Modell bei FCFS

Vorweg seien in der Abbildung 3.1 als Beispiel die Bäume dargestellt, welche sich aus dem Beispiel bei Abbildung 1.1 eindeutig ergeben.

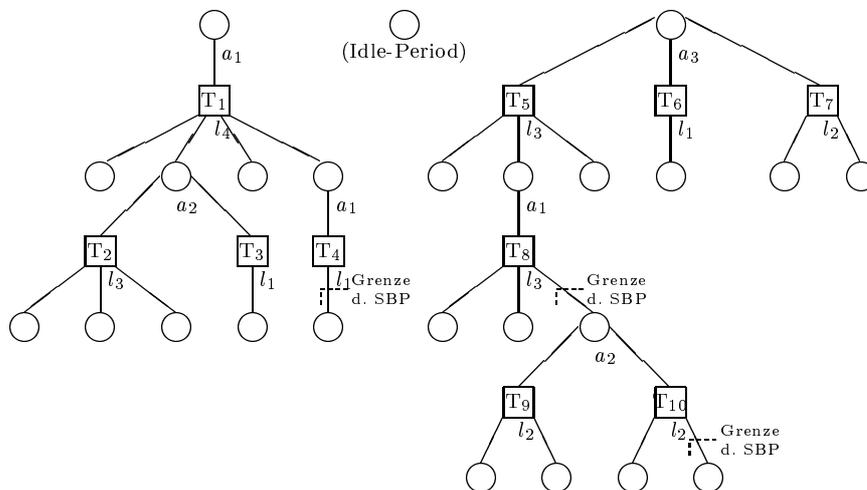


Abbildung 3.1: FCFS: Baum-Modell

ABARBEITUNGSREIHENFOLGE BEI FCFS: Da hier ja ein Task ohne Unterbrechung immer zur Gänze fertig bearbeitet wird, braucht man nur die Ebenen mit den Tasks zu betrachten. Die Bäume werden von links nach rechts,

innerhalb eines Baumes die Ebenen der Tasks von oben nach unten und innerhalb einer Task-Ebene die Tasks von links nach rechts abgearbeitet.

Im Beispiel der Abbildung 3.1 ergibt dies folgende Reihenfolge: $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$.

DEADLINE-VIOLATION BEI FCFS: Die Deadline wird bekanntlich genau dann verletzt, wenn vom Eintreffen des Tasks bis zu seiner Abarbeitung mehr Zeit (in Cycles) verstreicht, als mittels der Deadline vorgegeben ist. Dies kann man hier folgendermaßen prüfen:

Vom zu untersuchenden Task geht man zu seinem Vater-Knoten (nämlich ein runder, also ein Cycle) und zählt von diesem ausgehend alle rechts davon liegenden Cycles (der gleichen Ebene dieses Baumes) und alle Cycles der links vom untersuchten Task liegenden “Geschwister”-Tasks. Übersteigt die Summe aus dieser Anzahl und der Task-Länge des untersuchten Tasks die Deadline T der Task-Service-Time, so liegt eine Deadline-Violation vor.

Dieses Verfahren könnte man etwas umgestalten und damit vereinfachen. Man muß zu diesem Zwecke den Baum ausrichten, sodaß er rechts von oben nach unten gerade verläuft (vom ersten Idle-Cycle ganz oben, in der nächsten Ebene der letzte Task, welcher während dieses ersten Idle-Cycle eingetroffen ist, darunter der letzte Cycle dieses Tasks und so weiter). Von dieser am weitesten rechts liegenden Vertikale werden dann die weiteren Tasks und Cycles nach links hin immer in zueinander parallel verlaufenden Vertikalen passend eingereiht, wobei sie jedoch weiterhin in der ursprünglichen Horizontalen bleiben. Damit hat dieser graphisch umgestaltete, neue Baum genau die gleiche Bedeutung wie der ursprüngliche, jeder Knoten hat weiterhin denselben Vater- und dieselben Sohn-Knoten. Allerdings erkennt man jetzt leichter, wie lange ein Task bis zu seiner Bearbeitung warten muß und wann er fertig bearbeitet ist: Dies hängt dann nämlich nur mehr davon ab, wie weit der Task bzw. all seine Cycles von der rechtesten Vertikale entfernt ist bzw. sind — mit anderen Worten, wie breit der Baum ist.

Mit dieser bloß graphischen (jedenfalls aber nicht funktionellen) Umgestaltung des Baumes ist somit auch das Erkennen einer Deadline-Violation viel leichter: Sobald die Breite des Baumes, der einer SBP entspricht (— welcher also als Wurzel den Cycle hat, der die SBP startet —), die Deadline T übersteigt, ist die Deadline verletzt.

3.2 Baum-Modell bei pLCFS

Vorweg seien in der Abbildung 3.2 als Beispiel die Bäume dargestellt, welche sich aus dem Beispiel bei Abbildung 1.2 eineindeutig ergeben.

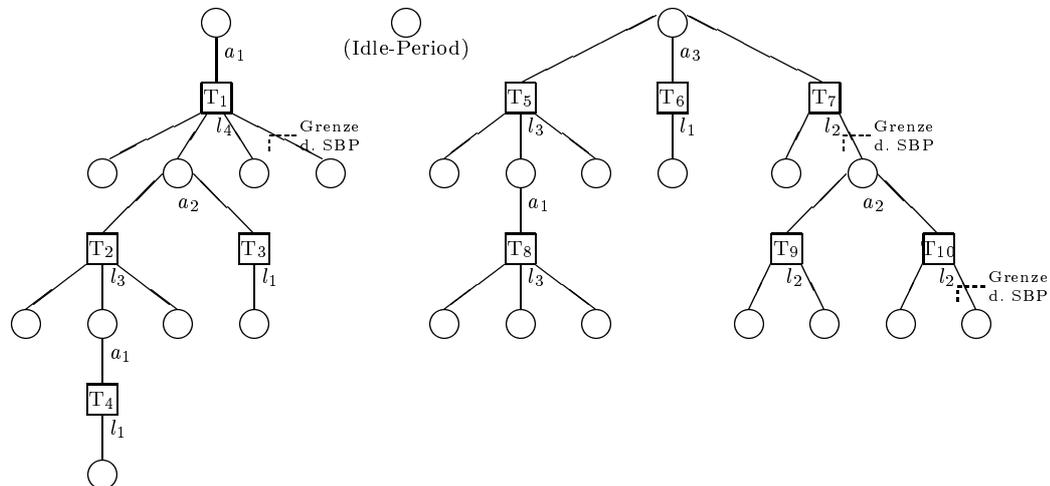


Abbildung 3.2: pLCFS: Baum-Modell

ABARBEITUNGSREIHENFOLGE BEI PLCFS: Die Bäume werden hier via Preorder-Traversing abgearbeitet, d. h., daß man ausgehend von der Wurzel alle Knoten der Reihe nach besucht (immer ganz links beginnend und den Zweigen entlang gehend) und dabei einen Task genau dann startet, wenn man ihn derart zum ersten Mal erreicht. Dadurch ergibt sich einerseits eine Task-Start-Reihenfolge und andererseits auch eine Task-Beendigungs-Reihenfolge, wann also der letzte Cycle eines Tasks bearbeitet wird.

Im Beispiel der Abbildung 3.2 ergibt dies folgende Task-Start-Reihenfolge: $T_1, T_2, T_4, T_3, T_5, T_8, T_6, T_7, T_9, T_{10}$, während die Task-Beendigungs-Reihenfolge natürlich anders aussieht: $T_4, T_2, T_3, T_1, T_8, T_5, T_6, T_7, T_9, T_{10}$.

DEADLINE-VIOLATION BEI PLCFS: Die Deadline wird bekanntlich genau dann verletzt, wenn vom Eintreffen des Tasks bis zu seiner Abarbeitung mehr Zeit (in Cycles) verstreicht, als mittels der Deadline vorgegeben ist. Dies kann man hier folgendermaßen prüfen:

Vom zu untersuchenden Task geht man zu seinem Vater-Knoten (nämlich ein runder, also ein Cycle) und zählt von diesem ausgehend alle Cycles der

links vom untersuchten Task liegenden “Geschwister”-Tasks inklusive deren Sohn- und Enkel- etc. -Tasks. Weiters addiert man die Anzahl der Cycles des untersuchten Tasks und all seiner Sohn- und Enkel- etc. -Tasks (bis auf die Söhne und Enkel etc. des letzten Cycle). Übersteigt die Summe aus diesen beiden Summen die Deadline T der Task-Service-Time, so liegt eine Deadline-Violation vor.

Da weiters bei pLCFS eine Deadline-Violation nur bei dem Task auftreten kann, welcher die BP bzw. die SBP eröffnet (genauer bei einem der Tasks, wenn nämlich gleichzeitig mehrere Tasks während einer Idle-Period eintreffen), kann man obiges Verfahren zur Bestimmung, ob eine Deadline-Violation vorliegt, vereinfachen, indem man nur die ersten Tasks, welche unmittelbar auf den ersten Idle-Cycle folgen, untersucht. Obiges Kriterium lautet dann folgendermaßen: Übersteigt die Größe des Baumes (also die Summe seiner Cycle-Knoten), der einer SBP entspricht (— welcher also als Wurzel den Cycle hat, der die SBP startet —), die Deadline T der Task-Service-Time, dann liegt eine Deadline-Violation vor.

3.3 Baum-Modell bei nLCFS

Vorweg seien in der Abbildung 3.3 als Beispiel die Bäume dargestellt, welche sich aus dem Beispiel bei Abbildung 1.3 eindeutig ergeben.

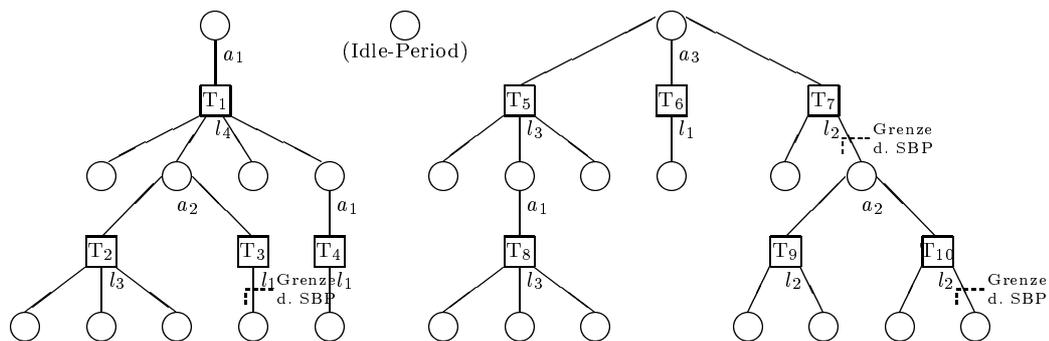


Abbildung 3.3: nLCFS: Baum-Modell

ABARBEITUNGSREIHENFOLGE BEI nLCFS: Die Bäume werden hier via Post-order-Traversing abgearbeitet, d.h., daß man ausgehend von der Wurzel alle Knoten der Reihe nach besucht (immer ganz rechts beginnend und den Zweigen entlang gehend) und dabei einen Task genau dann startet,

wenn man ihn derart zum ersten Mal erreicht. Haben allerdings Tasks den gleichen Vater-Knoten (nämlich einen runden, also einen Cycle), so werden diese “Geschwister”-Tasks der Reihe nach von links nach rechts besucht.

Hier könnte man natürlich auch o. B. d. A. die Abarbeitung von “Geschwister”-Tasks in der anderen Richtung, also von rechts nach links, wählen. Dies hätte den Vorteil, daß man die Abarbeitungsreihenfolge auch bei Tasks mit gleichem Vater-Knoten nicht als Ausnahme zu behandeln hätte, sondern auch hier das Postorder-Traversing strikt anwenden könnte, daß also auch solche Tasks von rechts nach links besucht werden (also eben genau dann gestartet werden, wenn man sie zum ersten Mal erreicht). Es ergäbe sich hiermit natürlich für einen konkreten Fall ein etwas anderer Baum (bzw. andere Bäume), die Bedeutung dieses Baumes, die Wahrscheinlichkeiten für sein Auftreten oder für das Auftreten einer Deadline-Violation blieben davon aber unberührt.

Im Beispiel der Abbildung 3.3 ergibt dies folgende Task-Reihenfolge: $T_1, T_4, T_2, T_3, T_5, T_8, T_6, T_7, T_9, T_{10}$, ohne daß es eine davon abweichende Task-Beendigungs-Reihenfolge gäbe, weil ja hier die Tasks, sobald sie einmal gestartet sind, nicht mehr unterbrochen werden.

Hat man hingegen die andere Variante der Abarbeitung von “Geschwister”-Tasks gewählt (— nämlich die nach der strikten Postorder-Traversing-Methode, also “Geschwister”-Tasks von rechts nach links —), so ergäbe dies in unserem Beispiel folgende Task-Reihenfolge: $T_1, T_4, T_3, T_2, T_7, T_8, T_6, T_5, T_{10}, T_9$. Diese Task-Reihenfolge läßt sich natürlich nicht mehr aus den Bäumen der Abbildung 3.3 herleiten, weil ja die Bäume hierbei ein anderes Aussehen hätten als die in diesem Beispiel.

DEADLINE-VIOLATION BEI NLCFS: Die Deadline wird bekanntlich genau dann verletzt, wenn vom Eintreffen des Tasks bis zu seiner Abarbeitung mehr Zeit (in Cycles) verstreicht, als mittels der Deadline vorgegeben ist. Dies kann man hier folgendermaßen prüfen:

Vom zu untersuchenden Task geht man zu seinem Vater-Knoten (nämlich ein runder, also ein Cycle) und zählt von diesem ausgehend alle Cycles der links vom untersuchten Task liegenden “Geschwister”-Tasks inklusive deren Sohn- und Enkel- etc. -Tasks. Weiters addiert man die Anzahl der Cycles des untersuchten Tasks selbst (nicht jedoch die all seiner Sohn- und Enkel- etc. -Tasks). Sodann summiert man noch die Anzahl aller rechts vom Vater-

Knoten liegenden “Geschwister”-Cycles des Vater-Knotens und die Anzahl der Cycles aller Sohn- und Enkel- etc. -Tasks dieser “Geschwister”-Cycles.

Übersteigt die Summe aus diesen drei Summen die Deadline T der Task-Service-Time, so liegt eine Deadline-Violation vor.

Bei der anderen Variante der Abarbeitung von “Geschwister”-Tasks (— nämlich bei der strikten Postorder-Traversing-Methode, also bei der Abarbeitung von “Geschwister”-Tasks von rechts nach links —) könnte man das Auftreten einer Deadline-Violation etwas einfacher prüfen:

Vom zu untersuchenden Task geht man zu seinem Vater-Knoten (nämlich ein runder, also ein Cycle) und zählt von diesem ausgehend alle Cycles der rechts vom untersuchten Task liegenden “Geschwister”-Tasks inklusive deren Sohn- und Enkel- etc. -Tasks und weiters summiert man noch die Anzahl aller rechts vom Vater-Knoten liegenden “Geschwister”-Cycles des Vater-Knotens und die Anzahl der Cycles aller Sohn- und Enkel- etc. -Tasks dieser “Geschwister”-Cycles. Sodann addiert man die Anzahl der Cycles des untersuchten Tasks selbst (nicht jedoch die all seiner Sohn- und Enkel- etc. -Tasks) hinzu.

Übersteigt nun die Summe aus diesen drei Summen die Deadline T der Task-Service-Time, so liegt eine Deadline-Violation vor.

3.4 Symbolische Notation von Bäumen, die OGF

Die erzeugende Funktion (englisch: *Ordinary Generating Function* OGF) dieser Familie von ebenen Wurzelbäumen (siehe auch Kapitel 3) läßt sich nun folgendermaßen herleiten:

Ein Baum besteht entweder aus einem einzigen Cycle (im Falle eines degenerierten Baumes, d. h. hier im Falle einer Idle-Period), oder aus einem Cycle mit einem Task (d. h. also ein Cycle, während dessen genau ein Task eintrifft), oder aus einem Cycle mit zwei Tasks (also ein Cycle, während dessen zwei Tasks eintreffen) etc.. Ein Task seinerseits enthält mindestens einen, maximal endlich viele Cycles. Da diese Cycles im allgemeinen Fall ihrerseits wieder die Wurzel von Teilbäumen sein können (wenn nämlich während des Cycles wieder Tasks eintreffen), kann man dies so formulieren, daß ein Task einen bis endlich viele Teilbäume enthält. Trifft allerdings während eines Cycles kein Task ein, so besteht eben ein solcher Teilbaum lediglich aus der Wurzel, also nur aus dem Cycle (\bigcirc) selbst.

Dadurch ergibt sich folgende symbolische Notation für die OGF dieser Familie von Bäumen:

$$\mathcal{B} = \sum_{k \geq 0} a_k \begin{array}{c} \circ \\ \diagdown \quad \diagup \\ \dots \\ \underbrace{\mathcal{T} \quad \mathcal{T}}_{k \text{ Zweige}} \end{array}, \quad \mathcal{T} = \sum_{n \geq 1} l_n \begin{array}{c} \square \\ \diagdown \quad \diagup \\ \dots \\ \underbrace{\mathcal{B} \quad \mathcal{B}}_{n \text{ Zweige}} \end{array},$$

wobei

\mathcal{B} ... die OGF dieser Familie von ebenen Wurzelbäumen und

\mathcal{T} ... die OGF der Familie der Tasks ist.

Somit kann man also die betrachteten Bäume mittels dieser geschlossenen Form einfach und eindeutig beschreiben.

Teil III

Busy-Period und Sub-Busy-Period

Kapitel 4

DIE BUSY-PERIOD-DURATION BPD

In diesem Kapitel wird nun die soeben erwähnte OGF der ebenen Wurzelbäume in die PGF von dazupassenden Zufallsvariablen übersetzt, sodaß man dann bequem mit diesen weiterrechnen kann.

4.1 Definition der BPD: $B(z)$

Sei

$B(z)$... die PGF der Länge der Busy-Period (BP), also die PGF der BPD,

b_i ... die Wahrscheinlichkeit, daß die BP die Länge i habe,

dann ist

$$B(z) = \sum_{i \geq 1} b_i z^i. \quad (4.1)$$

4.2 Herleitung der BPD aus $P(z)$, Berechnung von $B(z)$

Aus der OGF der Familie der ebenen Wurzelbäume im Kapitel 3.4 läßt sich nun leicht die PGF der Länge der BP, also die PGF der BPD, herleiten. Man braucht zu diesem Zwecke in der symbolischen Notation der OGF dieser Familie von Bäumen nur die Baumsymbole (\mathcal{B} , \mathcal{T}) durch die entsprechende PGF zu ersetzen und k Zweige eines Symbols als “(Symbol) ^{k} ” zu schreiben. Damit ergeben sich (unter Berücksichtigung der Definitionen von $A(z)$ und $L(z)$ in den Gleichungen 2.7 bzw. 2.8) folgende Zusammenhänge:

$$\begin{aligned} B(z) &= \sum_{k \geq 0} a_k z T^k(z) = z \cdot \sum_{k \geq 0} a_k T^k(z) = \\ &= z \cdot A(T(z)). \\ T(z) &= \sum_{n \geq 1} l_n B^n(z) = \\ &= L(B(z)). \end{aligned}$$

Setzt man nun die zweite Gleichung in die erste ein, so erhält man

$$\begin{aligned} B(z) &= z \cdot A(T(z)) = \\ &= z \cdot A(L(B(z))), \end{aligned}$$

und unter Zuhilfenahme von Gleichung 2.10 ergibt dies also

$$B(z) = z \cdot P(B(z)). \quad (4.2)$$

Konvergenzradius von $B(z)$:

Hiermit wurde also für $B(z)$ ein impliziter Ausdruck gefunden, welchen wir im folgenden Kapitel 4.3 näher untersuchen werden. Um allerdings überhaupt gültige Aussagen über $B(z)$ und seine Ableitungen treffen zu können, benötigt man natürlich auch einen passenden Konvergenzradius von $B(z)$. Wir setzen daher voraus:

$$R_B > 1 \quad (R_B \dots \text{Konvergenzradius von } B(z)). \quad (4.3)$$

Daß dies eine plausible Annahme ist, erkennt man z. B. gleich im Kapitel 4.3.2, weil ja dort $B(z)$ zumindest an der Stelle 1 konvergiert (wegen: $B(1) = \beta$).

4.3 Aussagen über die BPD

4.3.1 Berechnung von $B'(z)$

Durch implizites Ableiten von Gleichung 4.2 nach z und nach Umformen erhält man die 1. Ableitung von $B(z)$:

$$\begin{aligned} B(z) &= z \cdot P(B(z)) \\ B'(z) &= P(B(z)) + z \cdot P'(B(z)) \cdot B'(z) \\ B'(z) \cdot (1 - z \cdot P'(B(z))) &= P(B(z)), \end{aligned}$$

also:

$$B'(z) = \frac{P(B(z))}{1 - z \cdot P'(B(z))}. \quad (4.4)$$

4.3.2 Berechnung von $B(1)$

$B(1)$ ist ja gemäß Gleichung 2.3 im Kapitel 2.1 die Wahrscheinlichkeit, daß die BP (hier: im Rush-Hour-Case) endlich ist.

Aus Gleichung 4.2 erhält man durch Einsetzen des Wertes 1

$$B(1) = P(B(1)).$$

Dies bedeutet, daß $B(1)$ ein Fixpunkt der Funktion $P(z)$ ist. Weiters wissen wir bereits aus Satz 1 im Kapitel 2.3.2, daß $P(z)$ zwei Fixpunkte β und 1 hat, d. h. daß es in der obigen Gleichung für $B(1)$ zwei Lösungen gibt. Welche davon ist hier relevant?

Um dies ermitteln zu können, müssen beide Fälle durchgerechnet werden und auf eventuell auftretende Widersprüche geachtet werden, sodaß man hiermit einen Fall ausscheiden kann:

BEWEIS:

FALL 1) Sei $B(1) = 1$ (also: alle BP sind endlich): Wir untersuchen nun $B'(1)$, indem wir in der obigen Gleichung 4.4 den Wert 1 einsetzen und $P(1)$ wegen Gleichung 2.12 gleich 1 setzen:

$$\begin{aligned} B'(1) &= \frac{P(B(1))}{1 - 1 \cdot P'(B(1))} = \\ &= \frac{P(1)}{1 - P'(1)} = \\ &= \frac{1}{1 - P'(1)}. \end{aligned}$$

Da im Rush-Hour-Case ja per definitionem (siehe Gleichung 2.13) $P'(1) > 1$ ist, folgt daraus, daß $1 - P'(1) < 0$ ist und daß daher $B'(1) < 0$ sein muß.

Andererseits erhält man aus der 1. Ableitung von Gleichung 4.1 im Kapitel 4.1 und Einsetzen des Wertes 1

$$B'(1) = \sum_{i \geq 1} b_i \cdot i, \text{ mit } b_i \geq 0 \quad \forall i \in \mathbb{N},$$

was zur Folge hat, daß $B'(1) \geq 0$ sein muß.

Dazu steht jedoch obige Behauptung im Widerspruch, daß im Fall 1 $B'(1) < 0$ sein müßte, woraus folgt, daß der Fall 1 hier nicht als Lösung in Frage kommt.

Bemerkung: Der Fall 1 wäre hingegen die Lösung beim Regular-Case, weil genau bei diesem $P'(1) < 1$ ist (siehe dazu die Bemerkung zu Gleichung 2.13), womit es zu keinem Widerspruch käme, weil dann ja $B'(1) > 0$ wäre. Damit gilt also für den Regular-Case: $B(1) = 1$ (ist also die Wahrscheinlichkeit, daß eine endliche BP entsteht), $B'(1) = 1/(1 - P'(1))$ (ist also die mittlere Dauer der BP im Regular-Case).

FALL 2) Sei $B(1) = \beta$ ($\beta < 1$; also: nicht alle BP sind endlich): Auch hier untersuchen wir analog zum Fall 1 $B'(1)$, allerdings wird für $P(\beta) = \beta$ gesetzt:

$$\begin{aligned} B'(1) &= \frac{P(B(1))}{1 - 1 \cdot P'(B(1))} = \\ &= \frac{P(\beta)}{1 - P'(\beta)} = \\ &= \frac{\beta}{1 - P'(\beta)}. \end{aligned}$$

Da im Rush-Hour-Case ja (siehe Beweis des Satzes 1) $P'(\beta) < 1$ ist, folgt daraus, daß $1 - P'(\beta) > 0$ ist und daß daher $B'(1) > 0$ sein muß. Dies steht nicht im Widerspruch zu der im Fall 1 ermittelten Bedingung, daß $B'(1) \geq 0$ sein muß, woraus folgt, daß der Fall 2 die Lösung für den Rush-Hour-Case ist. ■

Somit wurde also folgendes gezeigt:

$$B(1) = \beta. \tag{4.5}$$

Dies ist genau die Wahrscheinlichkeit, daß im Rush-Hour-Case die BP endlich ist, d. h., daß eine endliche BP im Rush-Hour-Case entsteht.

Schranken von $B(1)$: Aus Satz 2 im Kapitel 2.3.3 wissen wir bereits (weil $P(x)$ streng monoton wachsend ist), daß

$$a_0 < \beta < 1,$$

womit wir also folgende Schranken für $B(1)$ erhalten:

$$a_0 < B(1) < 1. \tag{4.6}$$

Unendlich lange BP

Die Wahrscheinlichkeit, daß die BP unendlich lang ist, ist offensichtlich $1 - B(1)$, also ist (wegen Gleichung 4.5)

$$1 - B(1) = 1 - \beta. \quad (4.7)$$

Schranken von $1 - B(1)$: Da laut Gleichung 4.6 $B(1)$ durch a_0 und 1 beschränkt ist, folgen unmittelbar die Schranken für $1 - B(1)$:

$$0 < 1 - B(1) < 1 - a_0. \quad (4.8)$$

4.3.3 Berechnung von $B'(1)$

$B'(1)$ ist ja gemäß den Erläuterungen bei Gleichung 2.4 im Kapitel 2.1 der Mittelwert der Zufallsvariablen B , d.h. also hier, daß $B'(1)$ die mittlere Dauer der endlichen BP angibt.

Man braucht hier nur mehr die Erkenntnisse der beiden vorhergehenden Kapitel bzgl. der Berechnung von $B'(z)$ und $B(1)$ zu vereinen und in Gleichung 4.4 für z den Wert 1 einzusetzen und für $B(1)$ gemäß Gleichung 4.5 den Wert β :

$$\begin{aligned} B'(1) &= \frac{P(B(1))}{1 - 1 \cdot P'(B(1))} = \\ &= \frac{P(\beta)}{1 - 1 \cdot P'(\beta)} = \\ &= \frac{\beta}{1 - P'(\beta)}. \end{aligned}$$

Somit ergibt sich also für $B'(1)$, was ja die mittlere Dauer der endlichen BP ist, folgende Gleichung:

$$B'(1) = \frac{\beta}{1 - P'(\beta)}. \quad (4.9)$$

Schranken von $B'(1)$: Aus Gleichung 2.19 im Kapitel 2.3.4 wissen wir bereits die Schranken von $P'(\beta)$, weswegen wir nun leicht die Schranken für $B'(1)$ ermitteln können. Durch Anschreiben der Gleichung 2.19 und elementares Umformen erhält man sukzessive die Schranken für $B'(1)$:

$$\begin{aligned} 1 - \frac{a_0}{\beta} < P'(\beta) < 1 \\ -1 + \frac{a_0}{\beta} > -P'(\beta) > -1 \end{aligned}$$

$$\begin{aligned} \frac{a_0}{\beta} &> 1 - P'(\beta) > 0 \\ \frac{a_0}{\beta^2} &> \frac{1 - P'(\beta)}{\beta} > 0 \\ \frac{\beta^2}{a_0} &< \frac{\beta}{1 - P'(\beta)} < \infty . \end{aligned}$$

Somit haben wir also folgende untere Schranke für $B'(1)$ erhalten, während es keine obere Schranke dafür gibt:

$$\frac{\beta^2}{a_0} < B'(1) . \quad (4.10)$$

4.4 Unabhängigkeit der BPD von der Scheduling-Strategie

Wie bereits bei der Erläuterung zu den Abbildungen 1.1, 1.2 und 1.3 ab Seite 8 erwähnt wurde und aus eben diesen Abbildungen vermutet wird, daß die BPD unabhängig von der Scheduling-Strategie ist, soll dies nun näher erörtert werden.

Satz 3 *Die BPD, d. h. die Länge der BP, ist unabhängig von der gewählten Scheduling-Strategie und von der Deadline T .* ■

BEWEIS: Dies läßt sich sehr leicht an Hand der oben erwähnten Abbildungen erläutern. Eine BP beginnt ja bekanntlich genau dann, wenn während einer Idle-Period ein Task eintrifft (oder gleichzeitig mehrere eintreffen). Ab diesem Zeitpunkt wird vom Scheduler immer seriell genau ein Cycle nach dem anderen abgearbeitet, bis es keine Cycles mehr zu bearbeiten gibt, sodaß der Task genau seine Länge (also die Anzahl seiner Cycles) als Beitrag zur BPD liefert.

Zwischenzeitlich eintreffende weitere Tasks werden zwar je nach Scheduling-Strategie früher oder später bearbeitet, dies hat jedoch überhaupt keinen Einfluß auf die Länge der BP. Hingegen liefert ein jeder dieser Tasks (genauso wie der erste Task), sobald er vom Scheduler bearbeitet wird, für die Bearbeitung jedes seiner Cycles genau den Beitrag 1 zur BPD und somit in Summe genau seine gesamte Länge (also die Anzahl seiner Cycles) als Beitrag zur BPD. Daher ist die Länge der BP eben genau die Summe der Längen der darin enthaltenen Tasks +1 (für den Idle-Cycle, während dessen der erste Task eingetroffen ist und der per definitionem auch noch zur BP hinzu gerechnet wird). Das Ende der BP ist dann erreicht, wenn der letzte

Cycle bearbeitet wurde und in der Task-List keine Tasks mehr enthalten sind, sodaß nun eine Idle-Period folgt.

Daraus ersieht man also, daß die Länge der BP nur von der Länge der in der BP enthaltenen Tasks abhängig ist und die Scheduling-Strategie absolut keine Auswirkung auf die BPD hat.

Weiters wird aus diesen Erläuterungen klar, daß die Deadline T keinen Einfluß auf die BP haben kann. ■

Als Folge dieses Beweises erkennt man vielmehr, daß die BP nur von der Anzahl und Rate der ankommenden Cycles abhängig ist. Genauer: Die Länge der BP, d.h. die BPD, ist genau die Summe der Task-Längen der in der BP enthaltenen Tasks + 1 (für den einen Idle-Cycle am Beginn der BP), also genau die Summe der Anzahl der Cycles jedes in der BP enthaltenen Tasks + 1.

Kapitel 5

DIE SUB-BUSY-PERIOD-DURATION SBPD

Nachdem wir uns im vorigen Kapitel der BP gewidmet haben, gehen wir nun sozusagen eine Stufe tiefer, nämlich zur SBP. Sie ist ja bekanntlich eine ‐Teilmenge‐ der BP und als solche die maximale Begrenzung, binnen welcher ein beliebiger Task beim System ankommt und auch komplett abgearbeitet wird — d. h., daß ein beliebiger Task von der Ankunft beim System bis zur kompletten Abarbeitung (also während der Task-Service-Time) komplett innerhalb einer SBP liegen muß.

Eine Einführung in die Grundlagen der SBP wurde bereits im Kapitel 1.2 gegeben. Wesentlich ist jedenfalls, daß man damit das Modell verfeinern kann und daher exaktere Aussagen über das Verhalten des Schedulers bei der jeweiligen Scheduling-Strategie erhält und damit die $SRD(T)$ genauer berechnen kann.

5.1 Unabhängigkeit der SBPD von der Scheduling-Strategie

Die im Kapitel 4.4 gemachten Aussagen für die BP gelten analog auch für die SBP, wie weiter unten bewiesen werden wird. Auch hierzu möge man sich die Erläuterung der Abbildungen 1.1, 1.2 und 1.3 ab Seite 8 und die Abbildungen selbst anschauen, diesmal mit Hauptaugenmerk auf die in den Abbildungen eingezeichneten SBP. Die Tatsache, daß die SBPD unabhängig von der Scheduling-Strategie ist, soll nun näher erörtert werden.

Satz 4 *Die SBPD, d. h. die Länge der SBP, ist unabhängig von der gewählten Scheduling-Strategie und von der Deadline T .* ■

BEWEIS: Der Beweis wird analog zum Beweis von Satz 3 geführt, nur wird nun eben die SBP betrachtet.

Eine SBP beginnt ja bekanntlich genau beim Beginn einer BP oder am Ende der unmittelbar vorhergehenden SBP. Sie endet genau dann, wenn sich bis auf den soeben bearbeiteten Task kein weiterer mehr in der Task-List befindet und von diesem einen Task gerade der letzte Cycle bearbeitet wird (und zwar endet sie am Beginn dieses letzten Busy-Cycle, er selbst wird nicht mehr zur SBP hinzugezählt).

Ebenso wie beim Beweis von Satz 3 für die BP liefern auch bei der SBP genau die Tasks ihren Beitrag zur SBPD, welche zur betrachteten SBP gehören. Ein jeder dieser Tasks liefert, sobald er vom Scheduler bearbeitet wird, für die Bearbeitung jedes seiner Cycles genau den Beitrag 1 zur SBPD und somit in Summe genau seine gesamte Länge (also die Anzahl seiner Cycles) als Beitrag zur SBPD. Daher ist die Länge der SBP eben genau die Summe der Längen der darin enthaltenen Tasks. In welcher Reihenfolge die Tasks eintreffen und wie sie auf Grund der Scheduling-Strategie abgearbeitet werden hat jedoch überhaupt keinen Einfluß auf die Länge der SBP.

Daraus ersieht man also, daß die Länge der SBP nur von der Länge der in der SBP enthaltenen Tasks abhängig ist und die Scheduling-Strategie absolut keine Auswirkung auf die SBPD hat.

Weiters wird aus diesen Erläuterungen klar, daß die Deadline T keinen Einfluß auf die SBP haben kann. ■

Als Folge dieses Beweises erkennt man vielmehr, daß die SBP nur von der Anzahl und Rate der ankommenden Cycles abhängig ist. Genauer: Die Länge der SBP, d. h. die SBPD, ist genau die Summe der Task-Längen der in der SBP enthaltenen Tasks, also genau die Summe der Anzahl der Cycles jedes in der SBP enthaltenen Tasks.

Folgerung aus der Unabhängigkeit der SBPD von der Scheduling-Strategie

Da — wie soeben erwiesen wurde — die SBPD unabhängig von jedweder Scheduling-Strategie ist, ist es sinnvoll, die zukünftigen Berechnungen bezüglich SBP nur exemplarisch mit einer Scheduling-Strategie durchzuführen, weil die dabei gewonnenen Ergebnisse ohnehin für alle SBP gelten.

Wir beschränken uns daher im Verlauf der folgenden Kapitel auf die Scheduling-Strategie pLCFS, welche stellvertretend für alle anderen möglichen die benötigten Aussagen liefern wird.

5.2 Symbolische Notation der SBP, die OGF der SBP bei pLCFS

Die OGF der SBP erhält man analog zur Bestimmung der OGF der BP im Kapitel 3.4.

Ein Baum besteht entweder aus einem einzigen Cycle (im Falle eines degenerierten Baumes, d. h. hier im Falle einer Idle-Period), oder aus einem Cycle

mit einem Task (d. h. also ein Cycle, während dessen genau ein Task eintrifft), oder aus einem Cycle mit zwei Tasks (also ein Cycle, während dessen zwei Tasks eintreffen) etc., wobei der letzte Task gemäß Abarbeitungsreihenfolge (siehe dazu Kapitel 3) eine Sonderstellung hat: Da ja beim letzten Cycle dieses letzten Tasks bei pLCFS auf jeden Fall bereits alle Tasks der aktuellen SBP abgearbeitet sind, ist dieser letzte Cycle dieses letzten Tasks die Grenze der SBP, d. h., daß er nicht mehr zur aktuellen SBP hinzugerechnet wird aber eventuell zu einer neuen SBP, wenn während seiner Abarbeitung neue Tasks eintröfen.

Ein Task seinerseits enthält mindestens einen, maximal endlich viele Cycles. Da diese Cycles im allgemeinen Fall ihrerseits wieder die Wurzel von Teilbäumen sein können (wenn nämlich während des Cycles wieder Tasks eintreffen), kann man dies so formulieren, daß ein Task einen bis endlich viele Teilbäume enthält. Trifft allerdings während eines Cycles kein Task ein, so besteht eben ein solcher Teilbaum lediglich aus der Wurzel, also nur aus dem Cycle (\circ) selbst.

Für den speziellen letzten Task (welcher also während des die SBP eröffnenden Cycle eingetroffen ist und als letzter (oder einziger) der eventuell gleichzeitig eingetroffenen Tasks bearbeitet wird), gilt genau das gleiche wie für diese “normalen” Tasks mit einer Einschränkung: Sein letzter Cycle wird eben nicht mehr zur aktuellen SBP hinzugerechnet.

In der symbolischen Notation wird dieser spezielle letzte Task mit $\bar{\mathcal{T}}$ bezeichnet, sein letzter Cycle mit einem dreieckigen Knoten (\triangle), um damit anzuzeigen, daß er nicht mehr zur SBP hinzugerechnet wird. Der Baum für die SBP selbst wird analog zum Baum der BP mit $\bar{\mathcal{B}}$ bezeichnet, die bereits aus dem Kapitel 3.4 bekannten Bäume behalten weiterhin ihre Bezeichnungen (\mathcal{B} und \mathcal{T}) und ihre Bedeutung.

Dadurch ergibt sich folgende symbolische Notation für die OGF dieser Familie von Bäumen:

$$\bar{\mathcal{B}} = a_0 \cdot \circ + \sum_{k \geq 1} a_k \underbrace{\begin{array}{c} \circ \\ \swarrow \quad \dots \quad \searrow \\ \mathcal{T} \quad \dots \quad \mathcal{T} \\ \text{\scriptsize } k-1 \text{ Zweige} \end{array}}_{k-1 \text{ Zweige}} \bar{\mathcal{T}},$$

$$\bar{\mathcal{T}} = \sum_{n \geq 1} l_n \underbrace{\begin{array}{c} \square \\ \swarrow \quad \dots \quad \searrow \\ \mathcal{B} \quad \dots \quad \mathcal{B} \\ \text{\scriptsize } n-1 \text{ Zweige} \end{array}}_{n-1 \text{ Zweige}} \triangle,$$

\mathcal{B}, \mathcal{T} ... gleiche Bedeutung wie im Kapitel 3.4,

wobei

$\overline{\mathcal{B}}$... die OGF dieser Familie von eine SBP repräsentierenden ebenen Wurzelbäumen und

$\overline{\mathcal{T}}$... die OGF der Familie der speziellen letzten Tasks ist.

5.3 Definition der SBPD bei pLCFS: $\overline{B}(z)$

Sei

$\overline{B}(z)$... die PGF der Länge der Sub-Busy-Period (SBP), also die PGF der SBPD,

\overline{b}_i ... die Wahrscheinlichkeit, daß die SBP die Länge i habe,

dann ist

$$\overline{B}(z) = \sum_{i \geq 1} \overline{b}_i z^i. \quad (5.1)$$

5.4 Herleitung der SBPD aus BPD bei pLCFS, Berechnung von $\overline{B}(z)$

Aus der OGF dieser Familie von ebenen Wurzelbäumen der SBP im Kapitel 5.2 läßt sich nun leicht die PGF der Länge der SBP, also die PGF der SBPD, herleiten. Man braucht zu diesem Zwecke in der symbolischen Notation der OGF dieser Familie von Bäumen nur die Baumsymbole ($\overline{\mathcal{B}}$, $\overline{\mathcal{T}}$, \mathcal{B} und \mathcal{T}) durch die entsprechende PGF zu ersetzen und k Zweige eines Symbols als “(Symbol) ^{k} ” zu schreiben. Damit ergeben sich (unter Berücksichtigung der Definitionen von $L(z)$ in der Gleichung 2.8 bzw. der Herleitung von $T(z)$ im Kapitel 4.2) folgende Zusammenhänge:

$$\begin{aligned} \overline{B}(z) &= a_0 \cdot z + z \cdot \sum_{k \geq 1} a_k T^{k-1}(z) \cdot \overline{\mathcal{T}}(z), \\ \overline{\mathcal{T}}(z) &= \sum_{n \geq 1} l_n B^{n-1}(z) = \frac{1}{B(z)} \sum_{n \geq 1} l_n B^n(z) = \\ &= \frac{L(B(z))}{B(z)} = \frac{T(z)}{B(z)}. \end{aligned}$$

Setzt man nun die zweite Gleichung in die erste ein, so erhält man

$$\begin{aligned}\overline{B}(z) &= a_0 \cdot z + z \cdot \sum_{k \geq 1} [a_k T^{k-1}(z)] \cdot \frac{T(z)}{B(z)} = \\ &= a_0 \cdot z + \frac{z}{B(z)} \cdot \sum_{k \geq 1} [a_k T^{k-1}(z)] \cdot T(z) = \\ &= a_0 \cdot z + \frac{z}{B(z)} \cdot \sum_{k \geq 1} a_k T^k(z).\end{aligned}$$

Formt man nun den Summen-Ausdruck etwas um, so erkennt man darin die Definition von $A(z)$ (siehe Gleichung 2.7). Unter weiterer Zuhilfenahme der Herleitung von $B(z)$ im Kapitel 4.2 (vor Gleichung 4.2), nämlich daß $B(z) = z \cdot A(T(z))$, erhält man

$$\begin{aligned}\overline{B}(z) &= a_0 \cdot z + \frac{z}{B(z)} \cdot \sum_{k \geq 1} a_k T^k(z) = \\ &= a_0 \cdot z + \frac{z}{B(z)} \cdot \left[\sum_{k \geq 0} a_k T^k(z) - a_0 \cdot T^0(z) \right] = \\ &= a_0 \cdot z + \frac{z}{B(z)} \cdot [A(T(z)) - a_0 \cdot 1] = \\ &= a_0 \cdot z + \frac{z}{B(z)} \cdot \left[\frac{B(z)}{z} - a_0 \right] = \\ &= a_0 \cdot z + 1 - a_0 \cdot \frac{z}{B(z)},\end{aligned}$$

also

$$\overline{B}(z) = 1 + a_0 z - \frac{a_0 z}{B(z)}. \quad (5.2)$$

Konvergenzradius von $\overline{B}(z)$:

Hiermit wurde also für $\overline{B}(z)$ ein Ausdruck gefunden, welchen wir im folgenden Kapitel 5.5 näher untersuchen werden. Analog zum Kapitel 4.2 wollen wir nun auch für $\overline{B}(z)$ einen passenden Konvergenzradius voraussetzen:

$$R_{\overline{B}} > 1 \quad (R_{\overline{B}} \dots \text{Konvergenzradius von } \overline{B}(z)). \quad (5.3)$$

Daß dies eine plausible Annahme ist, erkennt man z. B. gleich im Kapitel 5.5.2, weil ja dort $\overline{B}(z)$ zumindest an der Stelle 1 konvergiert (wegen Gleichung 5.5 und Gleichung 5.6).

5.5 Aussagen über die SBPD bei pLCFS

5.5.1 Berechnung von $\overline{B}'(z)$ bei pLCFS

Durch einfaches Ableiten von Gleichung 5.2 nach z erhält man die 1. Ableitung von $\overline{B}(z)$:

$$\begin{aligned}\overline{B}(z) &= 1 + a_0 z - \frac{a_0 z}{B(z)} \\ \overline{B}'(z) &= a_0 - \frac{a_0 B(z) - a_0 z B'(z)}{B^2(z)},\end{aligned}$$

was also zu folgender Gleichung führt:

$$\begin{aligned}\overline{B}'(z) &= \frac{a_0 \cdot [B^2(z) - B(z) + zB'(z)]}{B^2(z)} = \\ &= a_0 \cdot \left(1 - \frac{1}{B(z)} + \frac{zB'(z)}{B^2(z)}\right).\end{aligned}\tag{5.4}$$

5.5.2 Berechnung von $\overline{B}(1)$ bei pLCFS

$\overline{B}(1)$ ist ja gemäß Gleichung 2.3 im Kapitel 2.1 die Wahrscheinlichkeit, daß die SBP (hier: im Rush-Hour-Case) endlich ist.

Aus Gleichung 5.2 erhält man durch Einsetzen des Wertes 1 für z

$$\begin{aligned}\overline{B}(1) &= 1 + a_0 \cdot 1 - \frac{a_0 \cdot 1}{B(1)} = \\ &= 1 + a_0 - \frac{a_0}{B(1)},\end{aligned}$$

was wegen Gleichung 4.5 im Rush-Hour-Case zu

$$\overline{B}(1) = 1 + a_0 - \frac{a_0}{\beta}\tag{5.5}$$

führt. Dies ist also genau die Wahrscheinlichkeit, daß im Rush-Hour-Case die SBP endlich ist, d. h., daß eine endliche SBP im Rush-Hour-Case entsteht.

Schranken von $\overline{B}(1)$: Da bekanntlich $a_0 < \beta < 1$ ist (siehe Satz 2 im Kapitel 2.3.3), folgt, daß $1 < 1/\beta < 1/a_0$ ist und daher $a_0 < a_0/\beta < 1$. Daraus folgt, daß einerseits (für $a_0/\beta < 1$)

$$\overline{B}(1) = 1 + a_0 - \frac{a_0}{\beta} > a_0,$$

und andererseits (für $a_0/\beta > a_0$)

$$\overline{B}(1) = 1 + a_0 - \frac{a_0}{\beta} < 1,$$

womit wir also folgende Schranken für $\overline{B}(1)$ erhalten:

$$a_0 < \overline{B}(1) < 1. \quad (5.6)$$

Unendlich lange SBP

Die Wahrscheinlichkeit, daß die SBP unendlich lang ist, ist offensichtlich $1 - \overline{B}(1)$, also (wegen Gleichung 5.5) $a_0/\beta - a_0$, und somit ist

$$1 - \overline{B}(1) = a_0 \left(\frac{1}{\beta} - 1 \right). \quad (5.7)$$

Schranken von $1 - \overline{B}(1)$: Da laut Gleichung 5.6 $\overline{B}(1)$ durch a_0 und 1 beschränkt ist, folgen unmittelbar die Schranken für $1 - \overline{B}(1)$:

$$0 < 1 - \overline{B}(1) < 1 - a_0. \quad (5.8)$$

5.5.3 Berechnung von $\overline{B}'(1)$ bei pLCFS

$\overline{B}'(1)$ ist ja gemäß den Erläuterungen bei Gleichung 2.4 im Kapitel 2.1 der Mittelwert der Zufallsvariablen \overline{B} , d. h. also hier, daß $\overline{B}'(1)$ die mittlere Dauer der endlichen SBP angibt.

Man braucht hier nur mehr die Erkenntnisse der vorhergehenden Kapitel bzgl. der Berechnung von $\overline{B}'(z)$, $B(1)$ und $B'(1)$ (siehe Gleichungen 4.5 und 4.9) zu vereinen und in Gleichung 5.4 für z den Wert 1 einzusetzen, für $B(1)$ gemäß Gleichung 4.5 den Wert β und für $B'(1)$ gemäß Gleichung 4.9 den Ausdruck $\beta/(1 - P'(\beta))$:

$$\begin{aligned} \overline{B}'(1) &= \frac{a_0 \cdot [B^2(1) - B(1) + 1 \cdot B'(1)]}{B^2(1)} = \\ &= \frac{a_0 \cdot \left[\beta^2 - \beta + \frac{\beta}{1 - P'(\beta)} \right]}{\beta^2} = \\ &= \frac{a_0}{\beta} \cdot \frac{\beta^2 - \beta + \frac{\beta}{1 - P'(\beta)}}{\beta}. \end{aligned}$$

Somit ergibt sich also für $\overline{B}'(1)$, was ja die mittlere Dauer der endlichen SBP ist, folgende Gleichung:

$$\begin{aligned}\overline{B}'(1) &= \frac{a_0}{\beta} \cdot \left[\beta - 1 + \frac{1}{1 - P'(\beta)} \right] = \\ &= a_0 \cdot \left[1 - \frac{1}{\beta} + \frac{1}{\beta} \cdot \frac{1}{1 - P'(\beta)} \right].\end{aligned}\quad (5.9)$$

Schranken von $\overline{B}'(1)$: Aus der Abschätzung für $B'(1)$ in Gleichung 4.10 im Kapitel 4.3.3 wissen wir bereits, daß

$$\frac{\beta^2}{a_0} < B'(1) = \frac{\beta}{1 - P'(\beta)},$$

also ist

$$\frac{\beta}{a_0} < \frac{1}{1 - P'(\beta)}.$$

Damit können wir nun in Gleichung 5.9 den einen Ausdruck abschätzen und erhalten derart eine Abschätzung für $\overline{B}'(1)$, indem wir ein paar elementare Umformungen durchführen:

$$\begin{aligned}\overline{B}'(1) &= a_0 \cdot \left[1 - \frac{1}{\beta} + \frac{1}{\beta} \cdot \frac{1}{1 - P'(\beta)} \right] > \\ &> a_0 \cdot \left[1 - \frac{1}{\beta} + \frac{1}{\beta} \cdot \frac{\beta}{a_0} \right] = \\ &= a_0 \cdot \left[1 - \frac{1}{\beta} + \frac{1}{a_0} \right] = \\ &= a_0 - \frac{a_0}{\beta} + 1 = \\ &= 1 + a_0 - \frac{a_0}{\beta}.\end{aligned}$$

Da weiters $a_0 < \beta$ ist (weil ja $P(x)$ streng monoton wachsend ist für $x \in \mathbb{R}^+$ und $x < R_P$ — siehe Satz 2 im Kapitel 2.3.3), muß $\frac{a_0}{\beta} < 1$ sein und daher $-\frac{a_0}{\beta} > -1$. Damit können wir nun bei der Abschätzung fortsetzen und erhalten

$$\begin{aligned}\overline{B}'(1) &> 1 + a_0 - \frac{a_0}{\beta} > \\ &> 1 + a_0 - 1 = \\ &= a_0.\end{aligned}$$

Somit haben wir also folgende untere Schranke für $\overline{B}'(1)$ erhalten, während es keine obere Schranke dafür gibt:

$$a_0 < \overline{B}'(1). \quad (5.10)$$

Kapitel 6

BPD UND SBPD BEI BERÜCKSICHTIGUNG DER DEADLINE T

Wie wir bereits in den vorigen Kapiteln gesehen haben, sind sowohl die BP als auch die SBP unabhängig von der Scheduling-Strategie und auch unabhängig von der Task-Service-Time Deadline T , weil sie ja nur die Cycle-Ankunftsrate widerspiegeln. Andererseits werden wir jedoch in diesem Kapitel erstmals auch die Deadline T mit berücksichtigen und sowohl für die BPD als auch für die SBPD, wenn sie die Deadline T nicht verletzen, eine passende PGF definieren und diese näher untersuchen.

6.1 Definition der BPD bei Deadline T : $B_T(z)$

Sei

$B_T(z)$. . . die PGF der Länge der Busy-Period (BP), welche die Deadline T nicht verletzt, also die PGF der BPD, unter der Zusatz-Einschränkung, daß die Deadline T nicht verletzt wird,

$b_{i,T}$. . . die Wahrscheinlichkeit, daß die BP die Länge i habe und die Deadline T nicht verletze,

dann ist

$$B_T(z) = \sum_{i \geq 1} b_{i,T} z^i. \quad (6.1)$$

Bemerkung: Man beachte, daß allgemein $b_{i,T} \neq b_i$ ist, wobei b_i die Wahrscheinlichkeit ist, daß die BP die Länge i hat (siehe Definition der BPD im Kapitel 4.1), und zwar ohne Berücksichtigung der Deadline T . Aus wahrscheinlichkeitstheoretischen Überlegungen ist offenbar, daß

$$b_{i,T} \leq b_i, \quad \forall i \in \mathbb{N}$$

gelten muß.

Weitere Abschätzungen von $b_{i,T}$ mit b_i und in der Folge von $B_T(z)$ mit $B(z)$ und auch die Berechnung von Grenzwerten siehe im Kapitel 6.5.

Annahme über den Grenzwert von $b_{i,T}$:

Da man allerdings a priori keine Aussage über die Monotonie von $b_{i,T}$ bei fixem i und variablem T treffen kann — wir untersuchen ja beliebige Scheduling-Strategien, ohne sie in dieser Hinsicht vorweg einschränken zu wollen —, können wir auch nichts über einen möglichen Grenzwert für $T \rightarrow \infty$ folgern (— die obere Schranke b_i alleine ist zwar notwendig, aber nicht hinreichend dafür).

Daher treffen wir, um einerseits ein sinnvolles Modell und andererseits auch eine gute Grundlage für die späteren Grenzwertberechnungen zu haben, folgende Annahme:

$$\lim_{T \rightarrow \infty} b_{i,T} = b_i \quad (6.2)$$

Diese Annahme erscheint für die meisten der herkömmlichen Scheduling-Strategien sicherlich passend und richtig, lediglich irgendwelche extra konstruierte Exoten dürften diese Annahme nicht erfüllen.

Diese Annahme gelte jedenfalls im ganzen hier vorliegenden Papier.

6.2 Definition der SBPD bei Deadline T : $\bar{B}_T(z)$

Sei

$\bar{B}_T(z)$... die PGF der Länge der Sub-Busy-Period (SBP), welche die Deadline T nicht verletzt, also die PGF der SBPD, unter der Zusatz-Einschränkung, daß die Deadline T nicht verletzt wird,

$\bar{b}_{i,T}$... die Wahrscheinlichkeit, daß die SBP die Länge i habe und die Deadline T nicht verletze,

dann ist

$$\bar{B}_T(z) = \sum_{i \geq 1} \bar{b}_{i,T} z^i. \quad (6.3)$$

Bemerkung: Man beachte, daß allgemein $\bar{b}_{i,T} \neq \bar{b}_i$ ist, wobei \bar{b}_i die Wahrscheinlichkeit ist, daß die SBP die Länge i hat (siehe Definition der SBPD im Kapitel 5.3), und zwar ohne Berücksichtigung der Deadline T . Aus wahrscheinlichkeitstheoretischen Überlegungen ist offenbar, daß

$$\bar{b}_{i,T} \leq \bar{b}_i, \quad \forall i \in \mathbb{N}$$

gelten muß.

Weitere Abschätzungen von $\bar{b}_{i,T}$ mit \bar{b}_i und in der Folge von $\bar{B}_T(z)$ mit $\bar{B}(z)$ und auch die Berechnung von Grenzwerten siehe im Kapitel 6.6.

Annahme über den Grenzwert von $\bar{b}_{i,T}$:

Da man auch hier analog zum Kapitel 6.1 a priori keine Aussage über die Monotonie von $\bar{b}_{i,T}$ bei fixem i und variablem T treffen kann, können wir auch nichts über einen möglichen Grenzwert für $T \rightarrow \infty$ folgern (— die obere Schranke \bar{b}_i alleine ist zwar notwendig, aber nicht hinreichend dafür).

Daher treffen wir genauso wie im Kapitel 6.1 folgende, für die späteren Grenzwertberechnungen nützliche Annahme:

$$\lim_{T \rightarrow \infty} \bar{b}_{i,T} = \bar{b}_i \quad (6.4)$$

Wie bereits im Kapitel 6.1 erwähnt, erscheint auch diese Annahme für die meisten der herkömmlichen Scheduling-Strategien sicherlich passend und richtig.

Diese Annahme gelte jedenfalls im ganzen hier vorliegenden Papier.

6.3 Herleitung der BPD bei Deadline T , Berechnung von $B_T(z)$

Nun gehen wir daran, eine nicht-deadline-verletzende Busy-Period zu konstruieren: Der einfachste Fall ist, daß sie nur aus einem einzigen Cycle (gemäß Definition der BP muß dies ein Idle-Cycle sein) besteht. Wenn nicht, besteht sie aus einer oder mehreren nicht-deadline-verletzenden Sub-Busy-Periods, welche alle unmittelbar aneinander anschließen, plus dem letzten, nicht mehr zur letzten SBP zählenden Busy-Cycle. Als notwendige Einschränkung dürfen diese nicht-deadline-verletzenden SBP jedoch nicht aus einem einzigen Idle-Cycle bestehen sondern müssen mindestens einen Task enthalten (sonst wäre ja in der BP, welche sich aus diesen SBP zusammensetzt, eine ‐Lücke‐, was aber gemäß Definition der BP nicht sein darf).

Setzen wir diese ‐Bauanleitung‐ nun in eine Gleichung für die PGF der Länge der nicht-deadline-verletzenden BP um, dann erkennt man folgende mathematische Ausdrücke: Der einzige Idle-Cycle wird repräsentiert durch den Ausdruck $a_0 z$, welcher zum Rest hinzu addiert wird, weil er ja alternativ auftritt. Eine nicht-deadline-verletzende SBP (allerdings ohne einen Idle-Cycle), wird durch den Ausdruck $\bar{B}_T(z) - a_0 z$ repräsentiert, wo eben $a_0 z$ subtrahiert wird, um damit

den Idle-Cycle auszuschließen. Da diese nicht-deadline-verletzenden SBP alternativ beliebig oft (mindestens einmal) auftreten können, bedeutet dies, daß man dafür $\sum_{k \geq 1} (\text{obiger Ausdruck})^k$ schreibt. Der noch ausständige letzte, nicht mehr zur letzten SBP zählende Busy-Cycle wird nun noch zu diesem Ausdruck mittels “ $\cdot a_0 z$ ” hinzu multipliziert. Somit erhalten wir folgende Gleichung für die PGF der Länge der nicht-deadline-verletzenden BP, also für $B_T(z)$:

$$B_T(z) = a_0 z + \left[\sum_{k \geq 1} (\overline{B}_T(z) - a_0 z)^k \right] \cdot a_0 z .$$

Durch Herausheben von $a_0 z$ und das Hineinnehmen des restlichen Terms in die Summe (wegen: $1 = x^0$) kann man dies unter Zuhilfenahme der Formel für eine geometrische Reihe ($\sum_{i \geq 0} x^i = 1/(1-x)$) wesentlich vereinfachen:

$$\begin{aligned} B_T(z) &= a_0 z + \left[\sum_{k \geq 1} (\overline{B}_T(z) - a_0 z)^k \right] \cdot a_0 z = \\ &= a_0 z \cdot \left[1 + \sum_{k \geq 1} (\overline{B}_T(z) - a_0 z)^k \right] = \\ &= a_0 z \cdot \left[\sum_{k \geq 0} (\overline{B}_T(z) - a_0 z)^k \right] = \\ &= a_0 z \cdot \frac{1}{1 - (\overline{B}_T(z) - a_0 z)} , \end{aligned}$$

sodaß man endgültig folgende Gleichung für $B_T(z)$ erhält:

$$B_T(z) = \frac{a_0 z}{1 - \overline{B}_T(z) + a_0 z} . \quad (6.5)$$

Konvergenzradius von $B_T(z)$ und $\overline{B}_T(z)$:

Nun wollen wir noch für $B_T(z)$ und $\overline{B}_T(z)$ jeweils den Konvergenzradius abschätzen. Da alle $b_{i,T} \leq b_i$ sind, folgt daraus, daß der Konvergenzradius von $B_T(z)$ jedenfalls \geq dem von $B(z)$ sein muß:

$$R_{B_T} \geq R_B \quad (R_{B_T} \dots \text{Konvergenzradius von } B_T(z)). \quad (6.6)$$

Analog gilt, daß alle $\overline{b}_{i,T} \leq \overline{b}_i$ sind, sodaß also der Konvergenzradius von $\overline{B}_T(z)$ jedenfalls \geq dem von $\overline{B}(z)$ sein muß:

$$R_{\overline{B}_T} \geq R_{\overline{B}} \quad (R_{\overline{B}_T} \dots \text{Konvergenzradius von } \overline{B}_T(z)). \quad (6.7)$$

6.4 Aussagen über die BPD und SBPD bei Deadline T

6.4.1 Berechnung von $B_T'(z)$

Durch einfaches Ableiten von Gleichung 6.5 nach z und Umformen erhält man die 1. Ableitung von $B_T(z)$:

$$\begin{aligned}
 B_T(z) &= \frac{a_0 z}{1 - \overline{B}_T(z) + a_0 z} \\
 B_T'(z) &= \frac{a_0(1 - \overline{B}_T(z) + a_0 z) - a_0 z(-\overline{B}_T'(z) + a_0)}{(1 - \overline{B}_T(z) + a_0 z)^2} = \\
 &= \frac{a_0 - a_0 \overline{B}_T(z) + a_0^2 z + a_0 z \overline{B}_T'(z) - a_0^2 z}{(1 - \overline{B}_T(z) + a_0 z)^2} = \\
 &= \frac{a_0 - a_0 \overline{B}_T(z) + a_0 z \overline{B}_T'(z)}{(1 - \overline{B}_T(z) + a_0 z)^2},
 \end{aligned}$$

was also zu folgender Gleichung führt:

$$B_T'(z) = \frac{a_0(1 - \overline{B}_T(z) + z \overline{B}_T'(z))}{(1 - \overline{B}_T(z) + a_0 z)^2}. \quad (6.8)$$

6.4.2 Berechnung von $B_T(1)$

$B_T(1)$ ist ja gemäß Gleichung 2.3 im Kapitel 2.1 und gemäß der Definition der BPD bei Deadline T im Kapitel 6.1 und gemäß der Gleichung 6.1 die Wahrscheinlichkeit, daß die BP (hier: im Rush-Hour-Case) die Deadline T nicht verletzt. Aus Gleichung 6.5 erhält man durch Einsetzen des Wertes 1 für z

$$B_T(1) = \frac{a_0 \cdot 1}{1 - \overline{B}_T(1) + a_0 \cdot 1},$$

und somit

$$B_T(1) = \frac{a_0}{1 - \overline{B}_T(1) + a_0}. \quad (6.9)$$

Dies ist also genau die Wahrscheinlichkeit, daß im Rush-Hour-Case die BP die Deadline T nicht verletzt.

Die Wahrscheinlichkeit, daß im Gegensatz dazu die BP die Deadline T verletzt, ist folglich $1 - B_T(1)$, also $1 - (a_0 / (1 - \overline{B}_T(1) + a_0))$ und somit ist

$$1 - B_T(1) = \frac{1 - \overline{B}_T(1)}{1 - \overline{B}_T(1) + a_0}. \quad (6.10)$$

6.4.3 Berechnung von $B'_T(1)$

$B'_T(1)$ ist ja gemäß den Erläuterungen bei Gleichung 2.4 im Kapitel 2.1 der Mittelwert der Zufallsvariablen B_T , d. h. also hier, daß $B'_T(1)$ die mittlere Dauer der nicht-deadline-verletzenden BP angibt.

Man braucht hier nur mehr in Gleichung 6.8 für z den Wert 1 einzusetzen und erhält hiermit

$$B'_T(1) = \frac{a_0(1 - \overline{B}_T(1) + 1 \cdot \overline{B}'_T(1))}{(1 - \overline{B}_T(1) + a_0 \cdot 1)^2}.$$

Somit ergibt sich also für $B'_T(1)$, was ja die mittlere Dauer der nicht-deadline-verletzenden BP ist, folgende Gleichung:

$$B'_T(1) = \frac{a_0(1 - \overline{B}_T(1) + \overline{B}'_T(1))}{(1 - \overline{B}_T(1) + a_0)^2}. \quad (6.11)$$

6.5 Abschätzung und Grenzwert der BPD bei Deadline T ($B_T(z)$)

In diesem Kapitel wird die im Kapitel 6.1 definierte PGF der Länge der nicht-deadline-verletzenden BP, also die PGF der nicht-deadline-verletzenden BPD, d. h. also $B_T(z)$,

- einerseits mittels $B(z)$, also der PGF der BPD ohne Berücksichtigung der Deadline T , abgeschätzt und
- andererseits deren Grenzverhalten für eine sehr große Deadline T , genauer für $T \rightarrow \infty$, bestimmt.

Wir werden sehen, daß sich im Grenzbereich ($T \rightarrow \infty$) $B_T(z)$ durch $B(z)$ annähern läßt.

6.5.1 Abschätzung von $B_T(z)$

Nun vergleichen wir $B_T(z)$ ($= \sum_{i \geq 1} b_{i,T} z^i$) mit $B(z)$ ($= \sum_{i \geq 1} b_i z^i$): Aus Kapitel 6.1 wissen wir bereits, daß

$$b_{i,T} \leq b_i, \quad \forall i \in \mathbb{N}$$

gelten muß. Dies kann man nun konkretisieren:

Es muß gelten:

$$\begin{aligned} \text{für } i \leq T & : b_{i,T} = b_i, \\ \text{für } i > T & : b_{i,T} \leq b_i, \end{aligned}$$

weil ja einerseits für $i \leq T$ die Deadline T nie verletzt werden kann und andererseits für $i > T$ entweder die Deadline T verletzt wird (und damit $b_{i,T} < b_i$ wäre) oder sie nicht verletzt wird (und damit $b_{i,T} = b_i$ wäre).

Daraus resultierend kann man nun leicht $B_T(z)$ mit $B(z)$ abschätzen, indem man in deren Definitionen (Gleichungen 6.1 bzw. 4.1) die Summanden komponentenweise miteinander vergleicht:

Wegen des soeben erhaltenen $b_{i,T} \leq b_i, \forall i \in \mathbb{N}$ ist natürlich

$$b_{i,T} \cdot z^i \leq b_i \cdot z^i, \forall i \in \mathbb{N}, z \in \mathbb{R}^+,$$

und daher ist auch $\sum_{i \geq 1} b_{i,T} z^i \leq \sum_{i \geq 1} b_i z^i$, für $z \in \mathbb{R}^+$ und $|z| < R_B$, womit man schließlich folgende Abschätzung erhält:

$$B_T(z) \leq B(z). \quad (6.12)$$

6.5.2 Grenzwert von $B_T(z)$

Nun wollen wir den $\lim_{T \rightarrow \infty} B_T(z) = \lim_{T \rightarrow \infty} \sum_{i \geq 1} b_{i,T} z^i$ berechnen. Leider kann man hier den “lim” nicht ungeprüft in die Summe hineinziehen und somit anstatt dessen $= \sum_{i \geq 1} [\lim_{T \rightarrow \infty} b_{i,T} z^i]$ schreiben, weil es sich ja hier um eine unendliche Reihe handelt. Unter gewissen Voraussetzungen (siehe unten) ist dies aber doch möglich.

Daher werden wir nun mit Hilfe der Annahme aus Gleichung 6.2 und einem Hilfssatz aus [5] (“Continuity Lemma”) zeigen, daß der Grenzwert von $B_T(z)$ mit $T \rightarrow \infty$ gleich $B(z)$ ist:

Lemma 1 (“Continuity Lemma” aus [5]) *Unter der Voraussetzung, daß $b_{i,T} \leq b_i$ ist für alle i und T , gilt: $\limsup_{i \rightarrow \infty} \sqrt[i]{|b_i|} = 1/R_B$ für $0 < R_B \leq \infty$ und $\lim_{T \rightarrow \infty} b_{i,T} = b_i$ für alle i genau dann, wenn $B(z)$ den Konvergenzradius R_B hat und $\lim_{T \rightarrow \infty} B_T(z) = B(z)$ für alle $|z| \leq r < R_B$ gleichmäßig konvergiert. ■*

BEWEIS:

RICHTUNG “ \implies ”: Da ja laut Voraussetzung $b_{i,T} \leq b_i$ (und beide ≥ 0) sind, ist $\limsup_{i \rightarrow \infty} \sqrt[i]{|b_{i,T}|} \leq \limsup_{i \rightarrow \infty} \sqrt[i]{|b_i|} = 1/R_B$ für alle T . Daher hat $B(z)$ den Konvergenzradius R_B und der Konvergenzradius R_{B_T} von $B_T(z)$ ist $\geq R_B$ für alle T .

Nun wählen wir für ein beliebiges $\epsilon (> 0)$ eine davon und von r abhängige Marke N ($N = N(\epsilon, r)$) derart, daß $\sum_{i > N} b_i r^i < \epsilon$, dann läßt sich

für $|z| \leq r < R_B$ folgendes herleiten:

$$\begin{aligned}
|B(z) - B_T(z)| &= \left| \sum_{i \geq 1} b_i z^i - \sum_{i \geq 1} b_{i,T} z^i \right| = \\
&= \left| \sum_{i=1}^N (b_i - b_{i,T}) z^i + \sum_{i > N} (b_i - b_{i,T}) z^i \right| \leq \\
&\leq \left| \sum_{i=1}^N (b_i - b_{i,T}) z^i \right| + \left| \sum_{i > N} b_i z^i \right| + \left| \sum_{i > N} b_{i,T} z^i \right| \leq \\
&\leq \sum_{i=1}^N (b_i - b_{i,T}) r^i + \sum_{i > N} b_i r^i + \sum_{i > N} b_{i,T} r^i \leq \\
&\leq \sum_{i=1}^N (b_i - b_{i,T}) r^i + \sum_{i > N} b_i r^i + \sum_{i > N} b_i r^i = \\
&= \sum_{i=1}^N (b_i - b_{i,T}) r^i + 2 \cdot \sum_{i > N} b_i r^i < \\
&< \sum_{i=1}^N (b_i - b_{i,T}) r^i + 2\epsilon ,
\end{aligned}$$

was, wenn man T groß genug wählt, sodaß jedenfalls die obige endliche Summe $\leq \epsilon$ ist, zu

$$|B(z) - B_T(z)| < \epsilon + 2\epsilon = 3\epsilon$$

führt. Damit ist also $|B(z) - B_T(z)|$ unabhängig von z kleiner als 3ϵ , womit die gleichmäßige Konvergenz von $B_T(z) \rightarrow B(z)$ für $T \rightarrow \infty$ und $|z| \leq r < R_B$ bewiesen.

RICHTUNG “ \Leftarrow ”: Wegen der (gleichmäßigen) Konvergenz von $B_T(z) \rightarrow B(z)$ für $T \rightarrow \infty$ innerhalb des Konvergenzradius R_B ist jedenfalls für ein beliebiges x mit $0 < x \leq r$

$$0 \leq \sum_{i \geq 1} (b_i - b_{i,T}) x^i = B(x) - B_T(x) .$$

Da alle Summanden in der Summe nicht-negativ sind, gilt folgende Abschätzung für jedes i und ein beliebiges x in den oben angeführten Grenzen:

$$0 \leq b_i - b_{i,T} x^i \leq B(x) - B_T(x) .$$

Für $T \rightarrow \infty$ geht nun die rechte Seite gegen 0, womit zwangsläufig auch die mittlere gegen 0 gehen muß, womit die Konvergenz $\lim_{T \rightarrow \infty} b_{i,T} = b_i$ bewiesen ist.

Da hier alle Bedingungen dieses Lemmas erfüllt sind (die Beschränktheit von $b_{i,T}$ nach oben durch b_i ; der Konvergenzradius R_B von $B(z)$; unsere Annahme aus Gleichung 6.2 bezüglich $\lim_{T \rightarrow \infty} b_{i,T}$), haben wir hiermit also den Grenzwert von $B_T(z)$ erhalten:

$$\lim_{T \rightarrow \infty} B_T(z) = B(z) . \quad (6.13)$$

Insbesondere heißt dies für $z = 1$ (unter Zuhilfenahme von Gleichung 4.5):

$$\lim_{T \rightarrow \infty} B_T(1) = B(1) = \beta , \quad (6.14)$$

womit also gezeigt wurde, daß die Wahrscheinlichkeit, daß die BP die Deadline T nicht verletzt, für große T gegen die Wahrscheinlichkeit, daß die BP endlich ist (nämlich $= \beta$), strebt.

6.6 Abschätzung und Grenzwert der SBPD bei Deadline T ($\overline{B}_T(z)$)

In diesem Kapitel wird die im Kapitel 6.2 definierte PGF der Länge der nicht-deadline-verletzenden SBP, also die PGF der nicht-deadline-verletzenden SBPD, d. h. also $\overline{B}_T(z)$,

- einerseits mittels $\overline{B}(z)$, also der PGF der SBPD ohne Berücksichtigung der Deadline T , abgeschätzt und
- andererseits deren Grenzverhalten für eine sehr große Deadline T , genauer für $T \rightarrow \infty$, bestimmt.

Wir werden sehen, daß sich im Grenzbereich ($T \rightarrow \infty$) $\overline{B}_T(z)$ durch $\overline{B}(z)$ annähern läßt.

Hier gehen wir ganz analog zum vorigen Kapitel 6.5 bezüglich Abschätzung und Grenzwert von $B_T(z)$ vor. Es wird daher im jetzigen Kapitel nicht mehr in derselben Ausführlichkeit wie im vorigen vorgegangen, sondern nur mehr die wesentlichen Aussagen dargestellt.

6.6.1 Abschätzung von $\overline{B}_T(z)$

Nun wird $\overline{B}_T(z)$ ($= \sum_{i \geq 1} \overline{b}_{i,T} z^i$) mit $\overline{B}(z)$ ($= \sum_{i \geq 1} \overline{b}_i z^i$) verglichen: Aus Kapitel 6.2 wissen wir bereits, daß

$$\overline{b}_{i,T} \leq \overline{b}_i , \forall i \in \mathbb{N}$$

gelten muß, was konkreter folgendermaßen formuliert werden kann:

$$\begin{aligned} \text{für } i \leq T & : \bar{b}_{i,T} = \bar{b}_i , \\ \text{für } i > T & : \bar{b}_{i,T} \leq \bar{b}_i , \end{aligned}$$

mit der analogen Begründung wie eben im vorigen Kapitel 6.5 bei der Abschätzung von $B_T(z)$.

Daraus resultiert wieder wegen der Definitionen von $\bar{B}_T(z)$ und $\bar{B}(z)$ (Gleichungen 6.3 bzw. 5.1) und wegen der soeben erhaltenen Beziehung zwischen $\bar{b}_{i,T}$ und \bar{b}_i

$$\bar{b}_{i,T} \cdot z^i \leq \bar{b}_i \cdot z^i , \forall i \in \mathbb{N} , z \in \mathbb{R}^+ ,$$

und daher ist auch $\sum_{i \geq 1} \bar{b}_{i,T} z^i \leq \sum_{i \geq 1} \bar{b}_i z^i$, für $z \in \mathbb{R}^+$ und $|z| < R_{\bar{B}}$, womit man schließlich folgende Abschätzung erhält:

$$\bar{B}_T(z) \leq \bar{B}(z) . \quad (6.15)$$

6.6.2 Grenzwert von $\bar{B}_T(z)$

Nun wollen wir den $\lim_{T \rightarrow \infty} \bar{B}_T(z) = \lim_{T \rightarrow \infty} \sum_{i \geq 1} \bar{b}_{i,T} z^i$ berechnen. Auch hier kann man wie im Kapitel 6.5.2 leider den “lim” nicht ungeprüft in die Summe hineinziehen und anstatt dessen $= \sum_{i \geq 1} [\lim_{T \rightarrow \infty} \bar{b}_{i,T} z^i]$ schreiben, weil es sich ja hier um eine unendliche Reihe handelt. Unter gewissen Voraussetzungen (siehe unten) ist dies aber doch möglich.

Daher verwenden wir nun die Annahme aus Gleichung 6.4 und wieder das Lemma 1, um zu zeigen, daß der Grenzwert von $\bar{B}_T(z)$ mit $T \rightarrow \infty$ gleich $\bar{B}(z)$ ist:

Da hier alle Bedingungen von Lemma 1 erfüllt sind (die Beschränktheit von $\bar{b}_{i,T}$ nach oben durch \bar{b}_i ; der Konvergenzradius $R_{\bar{B}}$ von $\bar{B}(z)$; unsere Annahme aus Gleichung 6.4 bezüglich $\lim_{T \rightarrow \infty} \bar{b}_{i,T}$), haben wir hiermit also den Grenzwert von $\bar{B}_T(z)$ erhalten:

$$\lim_{T \rightarrow \infty} \bar{B}_T(z) = \bar{B}(z) . \quad (6.16)$$

Insbesondere heißt dies für $z = 1$ (unter Zuhilfenahme von Gleichung 5.5):

$$\lim_{T \rightarrow \infty} \bar{B}_T(1) = \bar{B}(1) = 1 + a_0 - \frac{a_0}{\beta} , \quad (6.17)$$

womit also gezeigt wurde, daß die Wahrscheinlichkeit, daß die SBP die Deadline T nicht verletzt, für große T gegen die Wahrscheinlichkeit, daß die SBP endlich ist, strebt.

6.7 Grenzwerte der (höheren) Ableitungen von $B_T(z)$ und $\overline{B}_T(z)$

Analog zu den in den vorigen Kapiteln 6.5 und 6.6 für $B_T(z)$ respektive $\overline{B}_T(z)$ durchgeführten Abschätzungen und Grenzwertberechnungen läßt sich dies auch für beliebige Ableitungen dieser beiden PGF tun.

In den folgenden Ausdrücken sei $f^{(m)}(z)$ die m -te Ableitung der Funktion $f(z)$, dann kann man derart analog

$$B_T^{(m)}(z) = \sum_{i \geq m} b_{i,T} \cdot i(i-1) \cdots (i-(m-1)) z^{i-m}$$

mit

$$B^{(m)}(z) = \sum_{i \geq m} b_i \cdot i(i-1) \cdots (i-(m-1)) z^{i-m}$$

abschätzen und

$$\overline{B}_T^{(m)}(z) = \sum_{i \geq m} \overline{b}_{i,T} \cdot i(i-1) \cdots (i-(m-1)) z^{i-m}$$

mit

$$\overline{B}^{(m)}(z) = \sum_{i \geq m} \overline{b}_i \cdot i(i-1) \cdots (i-(m-1)) z^{i-m} .$$

Die Abschätzung ergibt sich offensichtlich aus den Abschätzungen bezüglich $b_{i,T}$ bzw. $\overline{b}_{i,T}$ im Kapitel 6.1 bzw. im Kapitel 6.2. Man erhält somit endgültig folgende Abschätzungen:

$$\begin{aligned} B_T^{(m)}(z) &\leq B^{(m)}(z) \\ \overline{B}_T^{(m)}(z) &\leq \overline{B}^{(m)}(z) . \end{aligned}$$

Zur Berechnung der Grenzwerte zieht man auch hier das Lemma 1 heran. Als Folgerung dieses Lemmas im Zusammenwirken mit dem Weierstraßschen Doppelreihen-Theorem (siehe [7], Seiten 133-136) ergeben sich dann folgende Grenzwerte:

$$\lim_{T \rightarrow \infty} B_T^{(m)}(z) = B^{(m)}(z) \quad (6.18)$$

$$\lim_{T \rightarrow \infty} \overline{B}_T^{(m)}(z) = \overline{B}^{(m)}(z) . \quad (6.19)$$

Insbesondere heißt dies für $z = 1$:

$$\lim_{T \rightarrow \infty} B_T^{(m)}(1) = B^{(m)}(1) \quad (6.20)$$

$$\lim_{T \rightarrow \infty} \overline{B}_T^{(m)}(1) = \overline{B}^{(m)}(1) , \quad (6.21)$$

wovon wir später, bei der Berechnung des Grenzwertes des Mittelwertes der SRD(T) im Kapitel 7.4, Gebrauch machen werden.

6.8 Unabhängigkeit der Grenzwerte (der Ableitungen) von $B_T(z)$ bzw. von $\bar{B}_T(z)$ von der Scheduling-Strategie

Wie bereits in Satz 3 und Satz 4 festgestellt, sind sowohl die BPD als auch die SBPD unabhängig von der Scheduling-Strategie und auch unabhängig von der Deadline T . Dies bedeutet daher auch, daß die entsprechenden PGF, nämlich $B(z)$ respektive $\bar{B}(z)$, davon unabhängig sind.

Folglich müssen auch alle Ableitungen dieser beiden PGF davon unabhängig sein, weil ja beim Ableiten nicht plötzlich T hinzukommt und genausowenig die Scheduling-Strategie dabei eine Rolle spielt. Dies ist auch leicht aus den bis jetzt bekannten Ableitungen in Gleichung 4.4 und Gleichung 5.4 ersichtlich, weil ja darin nur die Terme $P(\dots)$, $P'(\dots)$ und $B(\dots)$ respektive die Terme $B(\dots)$ und $B'(\dots)$ auftreten, welche allesamt unabhängig von der Scheduling-Strategie sind. Durch weiteres Ableiten dieser Ableitungen können bekanntlich keine weiteren, davon unabhängige Terme entstehen.

Somit gilt also folgender Satz (als Folge von Satz 3 und Satz 4):

Satz 5 *Die m -te Ableitung der PGF der BPD und der SBPD, also $B^{(m)}(z)$ bzw. $\bar{B}^{(m)}(z)$, für $m \in \mathbb{N}_0$, ist unabhängig von der Scheduling-Strategie und von der Deadline T . ■*

Im Gegensatz dazu sind selbstverständlich $B_T(z)$ und $\bar{B}_T(z)$ einerseits abhängig von der Deadline T (was ja aus ihrer Definition unmittelbar hervorgeht) und andererseits auch abhängig von der jeweils angewendeten Scheduling-Strategie, weil es ja genau bei der Betrachtung, ob nun die BP bzw. die SBP die Deadline T verletzt, entscheidend ist, wann mit der Bearbeitung eines Tasks begonnen und geendet wird, ob er eventuell sogar unterbrochen wird etc. .

Folglich müssen auch alle Ableitungen dieser beiden PGF davon abhängig sein, was man ganz leicht aus deren Definitionen ersieht: Die darin vorkommenden Terme $b_{i,T}$ bzw. $\bar{b}_{i,T}$ bleiben bei der m -ten Ableitung offensichtlich ab $i \geq m$ erhalten. Da aber $b_{i,T}$ bzw. $\bar{b}_{i,T}$ sowohl Information über die Deadline T als auch über die Scheduling-Strategie enthält (siehe auch die Erklärungen im obigen Absatz), gilt folgender Satz:

Satz 6 *Die m -te Ableitung der PGF der BPD und der SBPD unter Berücksichtigung der Deadline T , also $B_T^{(m)}(z)$ bzw. $\bar{B}_T^{(m)}(z)$, für $m \in \mathbb{N}_0$, ist abhängig von der Scheduling-Strategie und von der Deadline T . ■*

Nun braucht man nur mehr aus dem Satz 6 mittels Gleichung 6.18 bzw. Gleichung 6.19 den Grenzübergang für $B_T^{(m)}(z)$ bzw. $\overline{B}_T^{(m)}(z)$ zu machen, um damit $B^{(m)}(z)$ bzw. $\overline{B}^{(m)}(z)$ als Grenze zu erhalten und kann damit unter Zuhilfenahme von Satz 5 unmittelbar die Unabhängigkeit von der Scheduling-Strategie schließen:

Satz 7 *Die Grenzen (wenn $T \rightarrow \infty$) der m -ten Ableitung der PGF der BPD und der SBPD unter Berücksichtigung der Deadline T , also von $B_T^{(m)}(z)$ bzw. von $\overline{B}_T^{(m)}(z)$, für $m \in \mathbb{N}_0$, sind unabhängig von der Scheduling-Strategie und von der Deadline T . ■*

Dies machen wir uns im Kapitel 7.4 zunutze, um dort die Unabhängigkeit des Grenzwertes des Mittelwertes der SRD(T) von der Scheduling-Strategie zu zeigen.

Teil IV
Der Successful-Run

Kapitel 7

DIE SUCCESSFUL-RUN-DURATION SRD(T)

Bis dato haben wir nun alle benötigten “Bausteine” für einen Successful-Run zusammengetragen. Diese sind insbesondere die nicht-deadline-verletzende BP und die nicht-deadline-verletzende SBP mit den jeweiligen PGF der Länge der nicht-deadline-verletzenden BP bzw. SBP, nämlich $B_T(z)$ respektive $\overline{B}_T(z)$.

Nun können wir also daran gehen, aus diesen einen Successful-Run zu konstruieren und aus dem daraus entstehenden Modell Aussagen über die SRD(T) zu treffen.

7.1 Definition der SRD(T): $S_T(z)$

Sei

$S_T(z)$. . . die PGF der Länge eines Successful-Run, also die PGF der Successful-Run-Duration SRD(T), welche ja per definitionem die Deadline T berücksichtigt,

$s_{i,T}$. . . die Wahrscheinlichkeit, daß die SRD(T) die Länge i habe (daß also genau i Cycles nach dem Start des Computersystems eine deadline-verletzende BP beginnt),

dann ist

$$S_T(z) = \sum_{i \geq 1} s_{i,T} z^i. \quad (7.1)$$

7.2 Herleitung der SRD(T)

Nun konstruieren wir einen Successful-Run (— die Definition dessen siehe im Kapitel 1.2): Am kürzesten ist er, wenn gleich die erste BP die Deadline T verletzt. Wenn dem nicht so ist, so kann die erste BP eine nicht-deadline-verletzende sein aber die zweite die Deadline verletzen. Allgemein kann man dies so erweitern, daß also auf 0 bis n nicht-deadline-verletzende BP eine deadline-verletzende BP folgt. Per definitionem reicht der Successful-Run vom Start des Computersystems

genau bis zum Beginn der deadline-verletzenden BP und die $SRD(T)$ ist genau die Zeitdauer eines solchen Successful-Run.

Die Abbildung 7.1 stellt diesen Aufbau eines Successful-Run anschaulich dar. Die genaue Bedeutung der einzelnen Elemente der Abbildung ist bei der Erläuterung zu den Abbildungen 1.1, 1.2 und 1.3 ab Seite 8 zu finden.

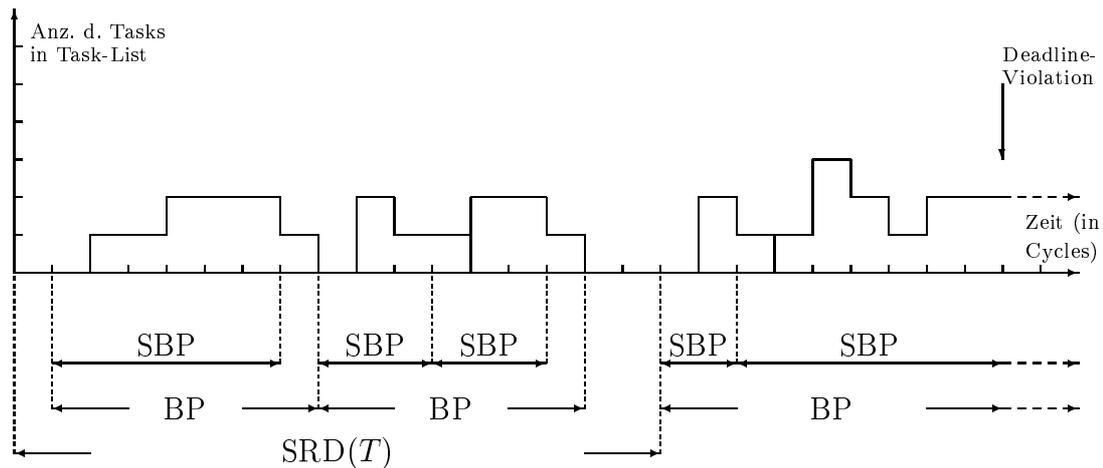


Abbildung 7.1: Die Successful-Run-Duration

Jedenfalls ersieht man aus der Abbildung, daß der Successful-Run beliebig viele nicht-deadline-verletzende BP und eine abschließende deadline-verletzende BP enthält. Nun setzen wir diese “Bauanleitung” in eine Gleichung für die PGF der Länge eines Successful-Run, also in eine Gleichung für die PGF der $SRD(T)$ (also $S_T(z)$) um: Da ja alternativ entweder keine oder eine oder allgemeiner beliebig viele nicht-deadline-verletzende BP den Beginn des Successful-Run bilden, wird dies — weil ja die PGF der nicht-deadline-verletzenden BP gemäß Gleichung 6.1 gleich $B_T(z)$ ist — durch den Ausdruck “ $1 + B_T(z) + B_T^2(z) + B_T^3(z) + \dots$ ” repräsentiert. Der Umstand, daß die terminierende BP die Deadline T jedenfalls verletzt, schlägt sich darin nieder, daß die Wahrscheinlichkeit dafür (nämlich $1 - B_T(1)$ gemäß Gleichung 6.10) zum bereits erhaltenen Ausdruck hinzu multipliziert wird. Somit erhält man folgende Gleichung für die PGF der $SRD(T)$:

$$S_T(z) = (1 + B_T(z) + B_T^2(z) + B_T^3(z) + \dots) \cdot (1 - B_T(1)) .$$

Wegen $1 = B_T^0(z)$ kann man dies unter Zuhilfenahme der Formel für eine geome-

trische Reihe ($\sum_{i \geq 0} x^i = 1/(1-x)$) wesentlich vereinfachen:

$$\begin{aligned} S_T(z) &= \left[\sum_{i \geq 0} B_T^i(z) \right] \cdot (1 - B_T(1)) = \\ &= \frac{1}{1 - B_T(z)} \cdot (1 - B_T(1)) , \end{aligned}$$

womit man schließlich folgende Gleichung für $S_T(z)$ erhält:

$$S_T(z) = \frac{1 - B_T(1)}{1 - B_T(z)} . \quad (7.2)$$

7.3 Berechnung des Mittelwertes der SRD(T)

Der Mittelwert der SRD(T) ist ja gemäß den Erläuterungen bei Gleichung 2.4 im Kapitel 2.1 gleich der 1. Ableitung der PGF der SRD(T) an der Stelle 1. Wir berechnen also vorerst die 1. Ableitung

$$\begin{aligned} S_T(z) &= \frac{1 - B_T(1)}{1 - B_T(z)} \implies \\ S_T'(z) &= \frac{-(1 - B_T(1))(-B_T'(z))}{(1 - B_T(z))^2} = \\ &= \frac{(1 - B_T(1))(B_T'(z))}{(1 - B_T(z))^2} , \end{aligned}$$

woraus wir unmittelbar durch Einsetzen des Wertes 1 für z und Umformen

$$\begin{aligned} [S_T'(z)]_{z=1} = S_T'(1) &= \frac{(1 - B_T(1))(B_T'(1))}{(1 - B_T(1))^2} = \\ &= \frac{B_T'(1)}{1 - B_T(1)} \end{aligned}$$

erhalten. Somit ergibt sich für die mittlere Dauer der SRD(T)

$$E[\text{SRD}(T)] = S_T'(1) = \frac{B_T'(1)}{1 - B_T(1)} \quad (7.3)$$

7.4 Grenzwert des Mittelwertes der SRD(T)

Hier braucht man nur mehr die Erkenntnisse aus den Kapiteln 6.5 und 6.6 über die Berechnung der Grenzwerte von BPD respektive SBPD bei Deadline T und

aus dem Kapitel 6.7 über die Grenzwerte der (höheren) Ableitungen von $B_T(z)$ und $\bar{B}_T(z)$, und zwar die Gleichungen 6.14 und 6.20, zu berücksichtigen, um

$$\begin{aligned} \lim_{T \rightarrow \infty} S'_T(1) &= \lim_{T \rightarrow \infty} \frac{B'_T(1)}{1 - B_T(1)} = \\ &= \frac{\lim_{T \rightarrow \infty} B'_T(1)}{\lim_{T \rightarrow \infty} (1 - B_T(1))} = \\ &= \frac{B'(1)}{1 - B(1)} \end{aligned}$$

zu erhalten. Unter Zuhilfenahme der Gleichungen 4.5 und 4.9 für $B(1)$ respektive $B'(1)$ erhält man

$$\begin{aligned} \lim_{T \rightarrow \infty} S'_T(1) &= \frac{\frac{\beta}{1 - P'(\beta)}}{1 - \beta} = \\ &= \frac{1}{1 - \beta} \cdot \frac{\beta}{1 - P'(\beta)}, \end{aligned}$$

womit man schließlich den Grenzwert für die mittlere Dauer des Successful-Run erhält:

$$E[\text{SRD}(T)] = S'_T(1) \quad , \text{ wobei } \quad \lim_{T \rightarrow \infty} S'_T(1) = \frac{1}{1 - \beta} \cdot \frac{\beta}{1 - P'(\beta)} .$$

Da ja bei Anwendung von Satz 7 bei der Herleitung von $\lim_{T \rightarrow \infty} S'_T(1)$ alle darin vorkommenden $B_T^{(m)}(1)$ in $B^{(m)}(1)$ übergehen, welche unabhängig von der Scheduling-Strategie sind, ist auch der Grenzwert selbst ($\lim(\dots)$) davon unabhängig. Dies ist auch sehr leicht daran zu erkennen, daß in diesem Grenzwert nur β und $P'(\beta)$ vorkommen, welche bekanntermaßen weder von der Scheduling-Strategie noch von der Deadline T abhängen. Also gilt dieser Grenzwert für *alle* Scheduling-Strategien im Rush-Hour-Case, weswegen wir dies zur Deutlichkeit noch extra formulieren:

Satz 8 *Im Rush-Hour-Case gilt für alle Scheduling-Strategien (d. h. also, unabhängig von jedweder Scheduling-Strategie) unter der Voraussetzung von Gleichung 6.2 und Gleichung 6.4 ($\lim_{T \rightarrow \infty} b_{i,T} = b_i$ bzw. $\lim_{T \rightarrow \infty} \bar{b}_{i,T} = \bar{b}_i$):*

Die mittlere SRD(T) strebt mit wachsendem T gegen einen nur von der Wahrscheinlichkeitsverteilung der ankommenden Cycles abhängenden Wert, d. h.,

$$E[\text{SRD}(T)] \longrightarrow \frac{1}{1 - \beta} \cdot \frac{\beta}{1 - P'(\beta)}, \text{ für } T \rightarrow \infty ,$$

wo $P(x)$ die PGF eben dieser Wahrscheinlichkeitsverteilung ist und β der kleinste Fixpunkt davon.

Kapitel 8

QUINTESSENZ UND AUSBLICK

8.1 Quintessenz

Abschließend kann also zusammengefaßt werden, daß wir für alle Scheduling-Strategien zwar einen geschlossenen Ausdruck zur Ermittlung des Mittelwertes der $\text{SRD}(T)$ gefunden haben (siehe Gleichung 7.3), nämlich

$$E[\text{SRD}(T)] = S'_T(1) = \frac{B'_T(1)}{1 - B_T(1)},$$

wobei aber die explizite Berechnung der darin vorkommenden Ausdrücke $B_T(1)$ und $B'_T(1)$ gemäß Gleichungen 6.9 respektive 6.11 zu den Ausdrücken $\overline{B}_T(1)$ und $\overline{B}'_T(1)$ führt, welche allerdings ihrerseits nicht direkt berechnet werden können, sondern wo nur Abschätzungen und Grenzwerte dafür gefunden werden konnten.

Daher wurde schließlich der Grenzwert für den Mittelwert der $\text{SRD}(T)$ ermittelt, welcher sich als *unabhängig von jedweder Scheduling-Strategie* erwies:

$$E[\text{SRD}(T)] = S'_T(1) \longrightarrow \frac{1}{1 - \beta} \cdot \frac{\beta}{1 - P'(\beta)}, \text{ für } T \rightarrow \infty .$$

Dieser gibt ein gutes Bild über das Verhalten des Computersystems wieder, weil er ja bei relativ kleiner Task-Execution-Time (= Task-Länge) und im Verhältnis dazu relativ großer Deadline T sehr nahe an die echte $\text{SRD}(T)$ heranreicht.

Beispiel

Als anschauliches Beispiel möchte ich hier in Tabelle 8.1 und in Tabelle 8.2 für verschiedene Werte von β und $P'(\beta)$ den Grenzwert der mittleren $\text{SRD}(T)$ berechnen. Da ja sowohl β als auch $P'(\beta)$ beschränkt sind (siehe Satz 1, Satz 2 und Gleichung 2.19), werden hierfür natürlich nur Werte innerhalb dieser Schranken angenommen. Interessant sind insbesondere die Werte in der Nähe der jeweiligen Grenzen.

Offensichtlich wächst für großes β und großes $P'(\beta)$ (d. h. für $\beta \rightarrow 1$, $P'(\beta) \rightarrow 1$) der Grenzwert der mittleren $\text{SRD}(T)$ unbeschränkt an (siehe Tabelle 8.2),

β	$P'(\beta)$	$\lim_{T \rightarrow \infty} S'_T(1)$
0,001	0,001	0,001002003
0,01	0,01	0,01020304
0,05	0,01	0,053163
0,05	0,05	0,055401
0,05	0,1	0,058479
0,05	0,5	0,105263
0,05	0,9	0,526315
0,1	0,1	0,123456
0,1	0,5	0,222222
0,1	0,9	1,111111
0,2	0,1	0,277777
0,2	0,5	0,5
0,2	0,9	2,5

Tabelle 8.1: Grenzwert des Mittelwertes der SRD(T) für kleines β und $P'(\beta)$

β	$P'(\beta)$	$\lim_{T \rightarrow \infty} S'_T(1)$
0,5	0,1	1,11
0,5	0,5	2,00
0,5	0,9	10,00
0,9	0,1	10,00
0,9	0,5	18,00
0,9	0,9	90,00
0,95	0,1	21,11
0,95	0,5	38,00
0,95	0,9	190,00
0,95	0,95	380,00
0,95	0,99	1900,00
0,99	0,99	9900,00
0,999	0,999	999000,00

Tabelle 8.2: Grenzwert des Mittelwertes der SRD(T) für großes β und $P'(\beta)$

während er für kleines β und kleines $P'(\beta)$ (d. h. für $\beta \rightarrow 0$, $P'(\beta) \rightarrow 0$) sehr klein wird (konkret: gegen β strebt) (siehe Tabelle 8.1).

Dies ist insofern nachvollziehbar, weil ja bei großem bzw. kleinem β auch $a_0 (=$

p_0) groß werden kann bzw. klein werden muß (siehe dazu Satz 2), wo bekanntlich a_0 bzw. p_0 die Wahrscheinlichkeit für das Eintreffen eines Tasks bzw. eines Cycles während eines Cycles ist. Und wenn eben diese Wahrscheinlichkeit groß ist, so wird verständlicherweise — weil ja nun relativ wenig Cycles eintreffen — die $SRD(T)$ größer werden, während bei einer kleinen solchen Wahrscheinlichkeit natürlich mehr Cycles eintreffen und daher die $SRD(T)$ kleiner werden wird.

8.2 Ausblick

In diesem Papier wurde eine für alle ankommenden Tasks gleiche Deadline T berücksichtigt und deren mittlere $SRD(T)$ für den Rush-Hour-Case berechnet. Für den Regular-Case wurde dies bereits in den Papieren [16], [3] und [17], für den Rush-Hour-Case bei FCFS auch schon in [2] ermittelt.

Darüber hinaus könnte es eventuell für zukünftige Anforderungen interessant und lohnenswert sein, einerseits auch für verschieden große Deadlines T_i , ($i = 1, \dots$) Ergebnisse zu erhalten und andererseits auch z.B. die Varianz oder weitere höhere Momente zu berechnen. Die Vermutung liegt nahe, daß auch die höheren Momente von $SRD(T)$ für wachsendes T unabhängig von der Scheduling-Strategie sind, weil sie wahrscheinlich auch nur Ableitungen von $B_T(z)$ und von $\overline{B}_T(z)$ enthalten werden.

Auch könnte man noch die hier erhaltenen Ergebnisse an Hand konkreter Wahrscheinlichkeitsverteilungen wie z. B. der Poisson-Verteilung, welche sicherlich eine gut passende Modellierung der Cycle-Ankunftsrate darstellt, untermauern.

Weiters harrt auch noch der Balanced-Case — zur Erinnerung: bei diesem ist die mittlere Cycle-Ankunftsrate = 1 (also $P'(1) = 1$) — auf seine genaue Untersuchung.

Abbildungsverzeichnis

1.1	FCFS: Tasks in der Task-List	10
1.2	pLCFS: Tasks in der Task-List	10
1.3	nLCFS: Tasks in der Task-List	11
2.1	$P(x)$ und $P(x) - x$	21
3.1	FCFS: Baum-Modell	25
3.2	pLCFS: Baum-Modell	27
3.3	nLCFS: Baum-Modell	28
7.1	Die Successful-Run-Duration	63

Tabellenverzeichnis

- 8.1 Grenzwert des Mittelwertes der $\text{SRD}(T)$ für kleines β und $P'(\beta)$. 67
- 8.2 Grenzwert des Mittelwertes der $\text{SRD}(T)$ für großes β und $P'(\beta)$. 67

Literatur

- [1] Edward A. Bender, *Asymptotic methods in enumeration*, SIAM Review **16** (1974), no. 4, 485–515.
- [2] Johann Blieberger and Ulrich Schmid, *FCFS scheduling in a hard real-time environment under rush-hour conditions*, BIT **32** (1992), no. 3, 370–383.
- [3] ———, *Preemptive LCFS scheduling in hard real-time applications*, Performance Evaluation **15** (1992), no. 3, 203–215.
- [4] E. Rodney Canfield, *Remarks on an asymptotic method in combinatorics*, J. of Comb. Theory **Series A37** (1984), 348–352.
- [5] Michael Drmota and Ulrich Schmid, *Exponential limiting distributions in queueing systems with deadlines*, SIAM J. on Appl. Math. **53** (1993), 301–318.
- [6] Philippe Flajolet and Andrew M. Odlyzko, *Singularity analysis of generating functions*, SIAM J. Disc. Math. **3** (1990), 216–240.
- [7] Peter Henrici, *Power series — integration — conformal mapping — location of zeros*, Applied and Computational Complex Analysis, vol. 1, John Wiley & Sons Inc., New York, 1974.
- [8] ———, *Special functions — integral transforms — asymptotics — continued fractions*, Applied and Computational Complex Analysis, vol. 2, John Wiley & Sons Inc., New York, 1977.
- [9] Konrad Knopp, *Theorie und Anwendung der unendlichen Reihen*, 5 ed., Springer-Verlag, Berlin, 1964.
- [10] Donald E. Knuth, *Fundamental algorithms*, second ed., The Art of Computer Programming, vol. 1, Addison-Wesley, Reading, Massachusetts, 1973.
- [11] ———, *Sorting and searching*, The Art of Computer Programming, vol. 3, Addison-Wesley, Reading, Massachusetts, 1973.
- [12] A. Meir and J.W. Moon, *On the altitude of nodes in random trees*, Canad. J. Math **30** (1978), 997–1015.

- [13] ———, *On an asymptotic method in enumeration*, J. of Comb. Theory **Series A51** (1989), 77–89.
- [14] W. F. Osgood, *Lehrbuch der Funktionentheorie*, fifth ed., vol. 1, Chelsea Publishing Company, Bronx, N.Y., 1928.
- [15] ———, *Lehrbuch der Funktionentheorie*, second ed., vol. 2, Chelsea Publishing Company, Bronx, N.Y., 1965.
- [16] Ulrich Schmid and Johann Blieberger, *Some investigations on FCFS scheduling in hard real-time applications*, Journal of Computer and System Sciences **45** (1992), no. 3, 493–512.
- [17] ———, *On non-preemptive LCFS scheduling with deadlines*, Journal of Algorithms **18** (1995), 124–158.
- [18] M. Ja. Wygodski, *Höhere Mathematik griffbereit*, Friedrich Vieweg + Sohn, Braunschweig, 1973.