

Farbdrucker Scheduling

Dominik Baumgartner, Thomas Samy Dafir, Laurentiu Vlad

1 Aufgabenstellung

Es soll ein Szenario simuliert werden, bei dem 2 Farbdrucker mehreren Benutzern zur Verfügung stehen und in gewissen Zeitabständen Print-Jobs mit verschiedenen Prioritäten und Abarbeitungszeiten erstellt werden. Insgesamt gibt es 3 Prioritäten: Student (niedrig), Professor (mittel), System (hoch). Jobs werden in Queues gesammelt und von beiden Druckern abgearbeitet. Dabei ist es immer möglich, dass Jobs mit niedrigerer Priorität während ihrer Bearbeitung in einem Drucker von Jobs höherer Priorität unterbrochen werden, sprich Professoren-Jobs unterbrechen Studenten-Jobs, System-Jobs unterbrechen alle anderen. Unterbrochene Jobs werden so bald als möglich wieder fortgesetzt.

2 Modell

Um den Sachverhalt aus der Aufgabenstellung simulieren zu können wurde ein Modell bestehend aus 4 Prozessen und einer Model-Klasse erstellt (+ 1 weiterer Prozess für die Erweiterung). Im Folgenden wird kurz auf die Aufgabe eines jeden Prozesses und seinen Beitrag zum Gesamtmodell eingegangen. Hierbei wird auch direkt die Erweiterung (Wartung) erläutert.

Als Simulationsdauer wurden 10 Stunden gewählt. Dies soll einen normalen Büro-/Uni- Arbeitstag repräsentieren.

2.1 Allgemein

Das Modell wurde so definiert, dass 2 Drucker existieren, von denen jeder seine eigene Drucker-Queue verwaltet. Jeder neue Druckauftrag wird also bei der Erstellung einem Drucker zugeordnet und dann von diesem abgearbeitet. Weiters verwaltet auch jeder Drucker seine eigene *InterruptedQueue* für unterbrochene Aufträge.

2.2 NewJobProcess

Wie in der Aufgabenstellung beschrieben, erhalten unsere beiden Drucker in gewissen Zeitabständen Druckaufträge unterschiedlicher Priorität. Diese werden vermutlich von Usern oder dem System in Auftrag gegeben. Um dies zu simulieren brauchen wir natürlich eine Art “Job-Generator”. Diese Aufgabe wird vom *NewJobProcess* übernommen. *NewJobProcess* läuft während der gesamten Simulationsdauer und erstellt in gewissen Zeitabständen neue Jobs mit einer bestimmten Ausführungsdauer. Danach wird für eine gewisse Zeit gewartet, bevor wieder ein Job erstellt wird. Die Warte- und Ausführungszeiten werden von einem *ContDistUniform*-Objekt in der Modell Klasse bezogen. Ein erstellter Prozess wird sofort aktiviert. Da wir 3 verschiedene Arten von Jobs simulieren, benötigen wir auch 3 Instanzen dieser Klasse.

2.3 JobProcess

Der *JobProcess* wird, wie bereits beschrieben, vom *NewJobProcess* erstellt und sofort aktiviert. Dieser Prozess repräsentiert einen Druckauftrag. Das Verhalten wurde so definiert, dass jedes *JobProcess*-Objekt selbst entscheidet, in welche Queue es sich einordnet. Jeder Job analysiert nachdem er aktiviert wurde beide Drucker-Queues und ordnet sich in jene mit der geringsten Gesamtbearbeitungszeit ein. Hier werden 2 Situationen unterschieden. Ist ein Drucker nicht beschäftigt, reiht sich der Job sofort in dessen Queue ein, setzt diesen auf *besetzt* und aktiviert ihn, sind jedoch beide Drucker besetzt, berechnet der Job beide Gesamtbearbeitungszeiten und reiht sich beim Drucker mit dem kleineren Wert ein. In der Erweiterung wird zusätzlich noch überprüft, ob sich ein Drucker im Wartungsmodus befindet. Ist dies der Fall, reiht sich der Job beim anderen Drucker ein.

2.4 PrinterProcess

Der *PrinterProcess* repräsentiert einen Drucker. Wir benötigen also 2 Instanzen dieser Klasse, um das Modell zu realisieren. Ein *PrinterProcess* läuft während der gesamten Simulationsdauer, wird jedoch zwischendurch immer wieder auf *hold* gesetzt, was der Abarbeitung eines Jobs entspricht. Der Drucker holt sich immer den Job mit der höchsten Priorität aus der Queue mit der höchsten Priorität (die *InterruptedQueue*, falls diese nicht leer ist). Der Drucker wird dann auf *hold* gesetzt (bearbeitet den Job). Jetzt können 2 Situationen eintreten. Im regulären Fall bearbeitet der Drucker den Job komplett, beendet ihn und holt sich den nächsten Job aus der Queue mit

der höchsten Priorität. Im zweiten Fall bearbeitet der Drucker den Job nur teilweise und wird dann unterbrochen, da sich ein Job mit höherer Priorität in der *JobQueue* befindet (*Supervisor* vergleicht Jobs und aktiviert den Drucker). Er berechnet dann die Restzeit des aktuellen Jobs, weist sie diesem zu und verschiebt ihn in die *InterruptedQueue*. Außerdem wird im Job eine *interruptable*-Flag auf *false* gesetzt. Dies wurde so definiert, um zu verhindern, dass der gleiche Job immer wieder unterbrochen wird. Weiters wird der Drucker auf *interrupted* gesetzt, was bewirkt, dass er jetzt nicht den Job aus der Queue mit der höchsten Priorität wählt (das wäre ja die *InterruptedQueue* mit dem soeben unterbrochenen Job), sondern den neuen Job aus der *JobQueue* (der der den aktuellen Job unterbrochen hat). Hier können natürlich während der Abarbeitung wieder die gleichen 2 soeben beschriebenen Situationen eintreten.

2.5 Supervisor

Der *Supervisor* steuert die Unterbrechung von Prozessen, die gerade bearbeitet werden. Eine Instanz dieser Klasse überwacht jeweils einen Drucker und wird immer dann aktiviert, wenn sich ein neuer *JobProcess* in die *JobQueue* eines Druckers einordnet. Der Supervisor vergleicht immer die Priorität des aktuellen Jobs im Drucker mit der des neu eingefügten Jobs. Ist die Priorität des neuen Job höher, aktiviert der *Supervisor* den jeweiligen Drucker (der während der Abarbeitung eines Auftrags auf *hold* gesetzt ist). Der restliche Unterbrechungsvorgang mitsamt der Verschiebung des aktuellen Jobs in die *InterruptedQueue* wird vom (dann wieder aktiven) Drucker übernommen. Der *Supervisor*-Prozess wird dann auf *passivate* gesetzt und wartet darauf, von einem neuen Job reaktiviert zu werden.

2.6 PrinterInkEmptyProcess (Erweiterung)

Dieser Prozess ist Teil unserer Erweiterung. Er wird in unserem Fall 2 mal aktiviert und versetzt dabei einen zufällig gewählten Drucker in den Wartungsmodus, indem er im *PrinterProcess* ein Flag setzt. Der Drucker reagiert darauf, indem er alle Aufträge aus seinen Queues in die des anderen Druckers verschiebt. Auch der aktuelle Job wird dem anderen Drucker zugewiesen und neu abgearbeitet. Dies wurde so gewählt, um sicherzustellen, dass Dokumente an einem einzigen Ort komplett ausgedruckt werden. Sämtliche neu ankommenden Jobs reihen sich nun auch direkt in die Queue des anderen Druckers ein. Ist die Wartung beendet, werden die Aufträge des nicht-gewarteten Druckers nicht wieder aufgeteilt. Die Längen der *JobQueues* beider Drucker gleichen sich sowieso wieder aus, da sich neue Jobs immer in

die kürzere einreihen (bis zum Längenausgleich ist dies die des gewarteten Druckers).

3 Ergebnisse

3.1 Simulationsparameter

Folgende Simulationsparameter wurden sowohl für den Standardfall (beide Drucker sind während der gesamten Simulationszeit aktiv), als auch für die erweiterte Simulation (Drucker können während der Simulationszeit ausfallen) gewählt.

Für die Erweiterung gelten zusätzlich noch folgende Sachverhalte.

Im ersten Fall wird ein Drucker nach 100 Minuten und der andere nach 300 Minuten unterbrochen. Im zweiten Fall wird zwei mal der gleiche Drucker unterbrochen. Das erste Mal nach 100 Minuten und das zweite Mal nach 300 Minuten.

Die Unterbrechungen dauern je 100 Minuten und sollen eine Wartung des Druckers simulieren (z.B. einen erforderlichen Tausch der OPC Trommel eines Tintenstrahldruckers).

- Simulationsdauer: 10h / 600 min
- Durchgeführte Simulationsläufe: 30
- StudJob: Ankunft alle 6-10 min, Bearbeitung: 2-6 min
- ProfJob: Ankunft alle 10-14 min, Bearbeitung: 2-6 min
- SystemJob: Ankunft alle 15-25 min, Bearbeitung: 2-10 min
- Wartung: Beginn: 100 min, 300 min, Dauer: 100 min

3.2 Analyse

	Ohne Erw.	Erw. Standard-Fall	Erw. Extremfall
Mittlere WZ Stud	1,33	38,5	55
Mittlere WZ Prof	0,98	3,9	4
Mittlere WZ Sys	0	0,1	0,1
Jobs Drucker 1 gesamt	86	96	13
Jobs Drucker 2 gesamt	66	67	123
Max WS Drucker 1	1	11	1
Max WS Drucker 2	0	18	18
Mittlere WS Drucker 1	0	1,4	0
Mittlere WS Drucker 2	0	3,2	8,5
Unterbrechungen gesamt	13	36	46
Unterbrechungen/h	1,3	3,6	4,6
Nicht mehr bearbeitet	0	0	36

3.2.1 Ohne Erweiterung

Die Ergebnisse der Standard-Simulation (ohne Erweiterung) entsprechen im Großen und Ganzen den Erwartungen. Die mittlere Wartezeit der System-Jobs ist 0, was bedeutet, dass diese sofort an die Reihe kommen (den aktuellen Job unterbrechen), sobald sie sich in eine Queue eingereiht haben.

Wie bereits erwähnt, erlaubt die Simulation nur eine Unterbrechung pro Job. Wurde also z.B. ein Stud-Job einmal unterbrochen, so kann dieser kein zweites Mal unterbrochen werden. Die Wartezeiten der Stud-Jobs und Prof-Jobs entsprechen ebenfalls den Erwartungen da diese, entsprechend ihrer Priorität, eine höhere oder niedrigere Wartezeit haben.

Auch die Verteilung der Jobs auf die Drucker ist weitestgehend ausgeglichen. (Jobs reihen sich in die Queue mit der niedrigsten Gesamtbearbeitungszeit ein)

Etwas unerwartet sind hingegen die maximalen Längen der Warteschlangen beider Drucker. Die Jobs des zweiten Druckers kamen immer an die Reihe sobald sie sich in die Queue einreichten; nur beim ersten Drucker musste ein Job gelegentlich warten. Das bedeutet, dass zwei Drucker vollkommen ausreichen, um das simulierte Aufkommen an Druckaufträgen zu bewältigen.

3.2.2 Erweiterung: Standardfall

(Ein Drucker nach 100, der andere nach 300 min unterbrochen)

Hier fällt sofort auf, dass sich die mittlere Wartezeit der Stud-Jobs von 1,33 min (Simulation ohne Erweiterung) auf 38,5 min erhöht, was einer Steigerung um ca. 2900% entspricht. Im Gegensatz dazu mussten Prof-Jobs ca. 300% (0,98 min auf 3,9 min) und Sys-Jobs nur 1% (0 min auf 0,1 min) länger warten.

Trotzt der Unterbrechung beider Drucker ist auch hier die Verteilung der Jobs mit 96 von Drucker 1 und 67 von Drucker 2 verarbeiteten Jobs relativ ausgeglichen.

Was sich signifikant erhöht hat ist die maximale Anzahl an Jobs in den Warteschlangen.

Drucker 1: von 1 auf 11

Drucker 2: von 0 auf 18

Dies war zu erwarten, da im Wartungsfall alle Jobs in die Warteschlange des funktionsfähigen Druckers verschoben werden.

Dies erklärt auch die ungefähre Verdreifachung der unterbrochenen Jobs von 13 auf 36.

3.2.3 Erweiterung: Extremfall

(Zweimalige Unterbrechung des gleichen Druckers. Das erste Mal nach 100 min, das zweite Mal nach 300 min.)

Obwohl in diesem Fall zweimal der gleiche Drucker ausgefallen ist, sind die mittleren Wartezeiten von System- und Prof-Jobs nicht nennenswert angestiegen. Wir hätten hier zumindest einen Anstieg der Wartezeit der Prof-Jobs erwartet. Es wurde lediglich eine höhere Wartezeit der Stud-Jobs von ca. 40% (38,5 min auf 55 min) verzeichnet.

Die Verteilung der Jobs auf die Drucker war so zu erwarten, da der erste Drucker ein Drittel der gesamten Simulationszeit inaktiv war (2x 100 min). Dies spiegelt sich auch in den maximalen und mittleren Längen der Warteschlangen beider Drucker wider.