

Informatik I - Tutorium

– Wintersemester 2006/2007 –

Joachim Wilke

<http://joachim-wilke.de/info1tut06.htm>

23. Januar 2007



Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

Quellennachweis & Dank an:
Bernhard Müller

Übersicht



- 1 Organisatorisches und Review
- 2 Und nochmal...
- 3 Prozesse
- 4 Speicher

1 Organisatorisches und Review

2 Und nochmal...

3 Prozesse

4 Speicher

- Wer in der Punktetabelle momentan mehr als 50% der Theorie-Punkte hat, braucht keine Übungsblätter mehr abgeben. Wer mehr als 50% der Praxis-Punkte hat, braucht nicht mehr in die Rechnerübung kommen.
- Die inoffizielle Probeklausur (IPK) findet am 27.01.2007 statt. Informationen und Anmeldung:
<http://anmeldungIPK.webhop.net>
(Link auf meiner Homepage).
- Bitte Samstag pünktlich sein (= bis 10 Uhr da sein).

- 1 Organisatorisches und Review
- 2 Und nochmal...**
- 3 Prozesse
- 4 Speicher

Und nochmal...



Der Reguläre Ausdruck ab^*c^*a ...

- ① ... beschreibt eine CH-3 Sprache.
- ② ... beschreibt die Sprache $L_1 = \{ab^n c^n a \mid n \in \mathbb{N}_0\}$
- ③ ... beschreibt eine endliche Sprache.

Gegeben $G_2 = (\{a, b\}, \{A, B\}, \{A \rightarrow Ab \mid Ba \mid a, B \rightarrow Aa \mid b\}, A)$.
 $L(G_2)$...

- ① ... enthält unendlich viele Elemente.
- ② ... kann nicht durch eine EBNF beschrieben werden.
- ③ ... wird von einem endlichen Akzeptoren A_2 akzeptiert.

Ein Akzeptor, bei dem der Startzustand auch Endzustand ist ...

- ① ... akzeptiert alle Worte $\in \Sigma^*$.
- ② ... akzeptiert alle Worte $\in \Sigma$.
- ③ ... akzeptiert immer auch das leere Worte ϵ .

Und nochmal...



Der Reguläre Ausdruck ab^*c^*a ...

- ① ... beschreibt eine CH-3 Sprache.
- ② ... beschreibt die Sprache $L_1 = \{ab^n c^n a \mid n \in \mathbb{N}_0\}$
- ③ ... beschreibt eine endliche Sprache.

Gegeben $G_2 = (\{a, b\}, \{A, B\}, \{A \rightarrow Ab \mid Ba \mid a, B \rightarrow Aa \mid b\}, A)$.
 $L(G_2)$...

- ① ... enthält unendlich viele Elemente.
- ② ... kann nicht durch eine EBNF beschrieben werden.
- ③ ... wird von einem endlichen Akzeptoren A_2 akzeptiert.

Ein Akzeptor, bei dem der Startzustand auch Endzustand ist ...

- ① ... akzeptiert alle Worte $\in \Sigma^*$.
- ② ... akzeptiert alle Worte $\in \Sigma$.
- ③ ... akzeptiert immer auch das leere Worte ϵ .

Und nochmal...



Der Reguläre Ausdruck ab^*c^*a ...

- ① ... beschreibt eine CH-3 Sprache.
- ② ... beschreibt die Sprache $L_1 = \{ab^n c^n a \mid n \in \mathbb{N}_0\}$
- ③ ... beschreibt eine endliche Sprache.

Gegeben $G_2 = (\{a, b\}, \{A, B\}, \{A \rightarrow Ab \mid Ba \mid a, B \rightarrow Aa \mid b\}, A)$.
 $L(G_2)$...

- ① ... enthält unendlich viele Elemente.
- ② ... kann nicht durch eine EBNF beschrieben werden.
- ③ ... wird von einem endlichen Akzeptoren A_2 akzeptiert.

Ein Akzeptor, bei dem der Startzustand auch Endzustand ist ...

- ① ... akzeptiert alle Worte $\in \Sigma^*$.
- ② ... akzeptiert alle Worte $\in \Sigma$.
- ③ ... akzeptiert immer auch das leere Worte ϵ .

Und nochmal...



Der Reguläre Ausdruck ab^*c^*a ...

- ① ... beschreibt eine CH-3 Sprache.
- ② ... beschreibt die Sprache $L_1 = \{ab^n c^n a | n \in \mathbb{N}_0\}$
- ③ ... beschreibt eine endliche Sprache.

Gegeben $G_2 = (\{a, b\}, \{A, B\}, \{A \rightarrow Ab | Ba | a, B \rightarrow Aa | b\}, A)$.
 $L(G_2)$...

- ① ... enthält unendlich viele Elemente.
- ② ... kann nicht durch eine EBNF beschrieben werden.
- ③ ... wird von einem endlichen Akzeptoren A_2 akzeptiert.

Ein Akzeptor, bei dem der Startzustand auch Endzustand ist ...

- ① ... akzeptiert alle Worte $\in \Sigma^*$.
- ② ... akzeptiert alle Worte $\in \Sigma$.
- ③ ... akzeptiert immer auch das leere Worte ϵ .

- 1 Organisatorisches und Review
- 2 Und nochmal...
- 3 Prozesse**
- 4 Speicher

Grundlagen I



Was ist ein Prozess?

Ein Prozess ist ein Programm und seine Daten, das sich im Hauptspeicher befindet und vom Prozessor ausgeführt werden kann.

Welchen Status kann ein Prozess besitzen?

New Der Prozess wurde erstellt und benötigt noch Ressourcen (Daten, freien Speicher, etc.)

Ready Der Prozess ist zur Ausführung bereit.

Running Der Prozess wird gerade ausgeführt.

Waiting/Blocked Der Prozess wartet auf ein Ereignis (z.B. Daten, Adressen von Zwischenergebnisse, etc.)

Terminated Der Prozess ist beendet, das Programm also ausgeführt. Er befindet sich noch im Hauptspeicher.

Grundlagen II



Da heute oft auch zeitkritische Anwendungen mehrere Prozesse nutzen, musste man sich Gedanken machen wie man die Rechenzeit des Prozessors "gerecht" verteilt, damit zB wichtige Prozesse nicht nach weniger wichtigen Prozessen ausgeführt werden bzw. ein zeitintensiver Prozess einen anderen unnötig lang warten lässt.

Welche Strategien gibt es?

First Come, First Served

Dieses Auswahlverfahren ist mit einer FIFO-Warteschlange zu vergleichen, bei der jeder ankommende Prozess ganz gleich seiner Dauer oder Wichtigkeit an die letzte Position der Warteschlange gestellt wird. Der erste Prozess in der Schlange wird vollständig ausgeführt.

Aufgabe: Nenne einen Vor- und Nachteil dieser Strategie.

Grundlagen II



Da heute oft auch zeitkritische Anwendungen mehrere Prozesse nutzen, musste man sich Gedanken machen wie man die Rechenzeit des Prozessors "gerecht" verteilt, damit zB wichtige Prozesse nicht nach weniger wichtigen Prozessen ausgeführt werden bzw. ein zeitintensiver Prozess einen anderen unnötig lang warten lässt.

Welche Strategien gibt es?

Shortest Job First

Hierbei wird die Warteschlange nach der Dauer der Prozesse, beginnend mit dem Kürzesten, geordnet. Kommt ein neuer Prozess an, wird er hinter den Prozessen mit kürzerer oder gleicher Dauer und vor den Prozessen mit längerer Dauer eingereiht. Der erste Prozess in der Schlange wird vollständig ausgeführt.

Aufgabe: Nenne einen Vor- und Nachteil dieser Strategie.

Grundlagen II



Da heute oft auch zeitkritische Anwendungen mehrere Prozesse nutzen, musste man sich Gedanken machen wie man die Rechenzeit des Prozessors "gerecht" verteilt, damit zB wichtige Prozesse nicht nach weniger wichtigen Prozessen ausgeführt werden bzw. ein zeitintensiver Prozess einen anderen unnötig lang warten lässt.

Welche Strategien gibt es?

Priority

Die Prozesse werden mit Prioritäten versehen (statisch oder dynamisch) und wiederum bei ankommen eines neuen Prozesses beginnend mit der höchsten Priorität sortiert. Ist das Verfahren ein verdrängendes, so wird ein niederpriorer Prozess abgebrochen und neu eingereiht, sobald ein höherpriorer Prozess hinzukommt.

Aufgabe: Nenne einen Vor- und Nachteil dieser Strategie.

Grundlagen II



Da heute oft auch zeitkritische Anwendungen mehrere Prozesse nutzen, musste man sich Gedanken machen wie man die Rechenzeit des Prozessors "gerecht" verteilt, damit zB wichtige Prozesse nicht nach weniger wichtigen Prozessen ausgeführt werden bzw. ein zeitintensiver Prozess einen anderen unnötig lang warten lässt.

Welche Strategien gibt es?

Round Robin

Hier erhalten die Prozesse jeweils ein Zeitfenster, in denen sie ausgeführt werden. Danach wird der Prozess abgespeichert und erneut an das Ende der Schlange eingereiht. Ein vorheriges Sortieren entfällt. Sollte ein Prozess innerhalb oder vor dem Ende seines Fensters beendet sein, schließt der nächste Prozess sofort an.

Aufgabe: Nenne einen Vor- und Nachteil dieser Strategie.

Auswahlstrategien



Gegeben seien die folgenden Prozesse mit ihren Laufzeiten

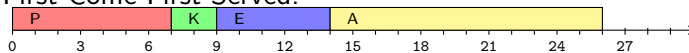
- Berechnung von PI auf 1000 Stellen (P, 7s)
- Berechnung des Klausurdurchschnitts (K, 2s)
- Erstellen von Info1-Übungsscheinen (E, 5s)
- Automatische Korrektur von Übungsblättern (A, 12s)

Die Reihenfolge der Ankunft entspricht der Reihenfolge in dieser Auflistung. In welcher Reihenfolge werden die Prozesse mit den Strategien

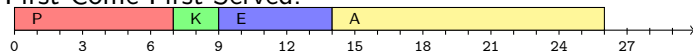
- First Come First Served
- Shortest Job First
- Round Robin (Zeitscheibe 3 sec)

abgearbeitet? Welche Warte- und Verweilzeiten ergeben sich?

First Come First Served:



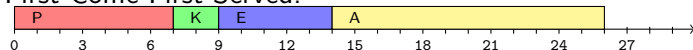
First Come First Served:



Wartezeiten: $P = 0s$, $K = 7s$, $E = 9s$, $A = 14s$, im Mittel $7,5s$.

Verweilzeiten: $P = 7s$, $K = 9s$, $E = 14s$, $A = 26s$, im Mittel $14s$.

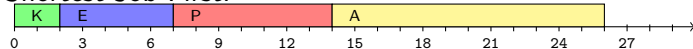
First Come First Served:



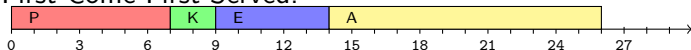
Wartezeiten: $P = 0s$, $K = 7s$, $E = 9s$, $A = 14s$, im Mittel $7,5s$.

Verweilzeiten: $P = 7s$, $K = 9s$, $E = 14s$, $A = 26s$, im Mittel $14s$.

Shortest Job First:



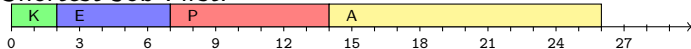
First Come First Served:



Wartezeiten: $P = 0s$, $K = 7s$, $E = 9s$, $A = 14s$, im Mittel $7,5s$.

Verweilzeiten: $P = 7s$, $K = 9s$, $E = 14s$, $A = 26s$, im Mittel $14s$.

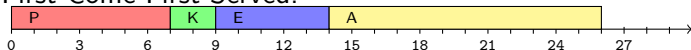
Shortest Job First:



Wartezeiten: $P = 7s$, $K = 0s$, $E = 2s$, $A = 14s$, im Mittel $5,75s$.

Verweilzeiten: $P = 14s$, $K = 2s$, $E = 7s$, $A = 26s$, im Mittel $12,25s$.

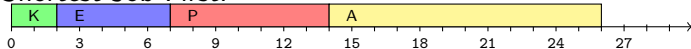
First Come First Served:



Wartezeiten: $P = 0s$, $K = 7s$, $E = 9s$, $A = 14s$, im Mittel $7,5s$.

Verweilzeiten: $P = 7s$, $K = 9s$, $E = 14s$, $A = 26s$, im Mittel $14s$.

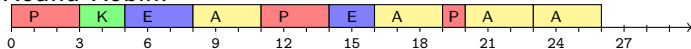
Shortest Job First:



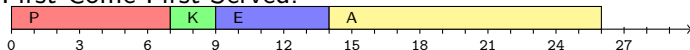
Wartezeiten: $P = 7s$, $K = 0s$, $E = 2s$, $A = 14s$, im Mittel $5,75s$.

Verweilzeiten: $P = 14s$, $K = 2s$, $E = 7s$, $A = 26s$, im Mittel $12,25s$.

Round Robin:



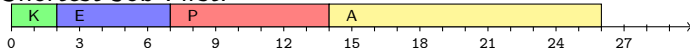
First Come First Served:



Wartezeiten: $P = 0s$, $K = 7s$, $E = 9s$, $A = 14s$, im Mittel $7,5s$.

Verweilzeiten: $P = 7s$, $K = 9s$, $E = 14s$, $A = 26s$, im Mittel $14s$.

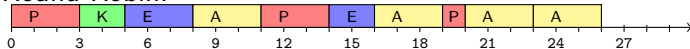
Shortest Job First:



Wartezeiten: $P = 7s$, $K = 0s$, $E = 2s$, $A = 14s$, im Mittel $5,75s$.

Verweilzeiten: $P = 14s$, $K = 2s$, $E = 7s$, $A = 26s$, im Mittel $12,25s$.

Round Robin:



Wartezeiten: $P = 13s$, $K = 3s$, $E = 11s$, $A = 14s$, im Mittel $10,25s$.

Verweilzeiten: $P = 20s$, $K = 5s$, $E = 16s$, $A = 26s$, im Mittel $16,75s$.

- 1 Organisatorisches und Review
- 2 Und nochmal...
- 3 Prozesse
- 4 Speicher**



Wie kann ich Speicherbelegungen darstellen?

Durch einen Graphen, einer Verketteten Liste mit Angabe ob der Speicher frei (F) oder belegt (B) ist oder einem Bitmap - einer Binärzahl in der Länge des Speichers, bei der die 1 eine Belegung an dieser Stelle angibt.

einige Vergabestrategien

First Fit Die Daten werden an die erste mögliche Stelle im Speicher geschrieben.

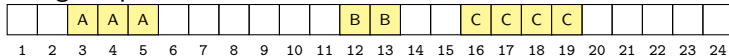
Best Fit Speichert am Anfang der ersten und kleinst-möglichen freien Speicherstelle.

Worst Fit Speichert am Anganf der ersten und größten freien Speicherstelle.

Speicherverwaltung



Gegeben sei ein 24 kB großer Speicherabschnitt der bereits drei belegte Speicherbereiche enthält:

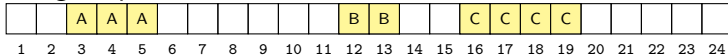


- einen 3 kB großen Bereich A beginnend bei Adresse 3
- einen 2 kB großen Bereich B beginnend bei Adresse 12
- einen 4 kB großen Bereich C beginnend bei Adresse 16

Speicherverwaltung



Gegeben sei ein 24 kB großer Speicherabschnitt der bereits drei belegte Speicherbereiche enthält:



Folgende Speicherbereiche werden jetzt zusätzlich angefordert

- einen 2 kB großen Bereich D
- einen 4 kB großen Bereich E
- einen 3 kB großen Bereich F

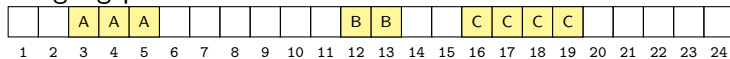
Wie sieht die Speicherbelegung anschliessen aus, wenn folgende Strategien verwendet werden?

- First Fit
- Best Fit
- Worst Fit

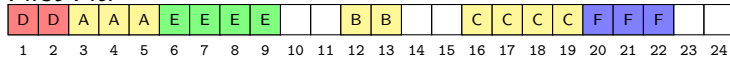
Ausgangspunkt:

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|
| | | A | A | A | | | | | | | | B | B | | | | C | C | C | C | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | | |

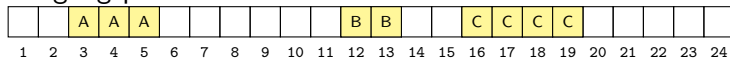
Ausgangspunkt:



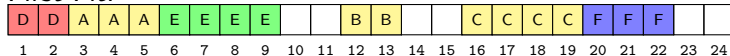
First Fit:



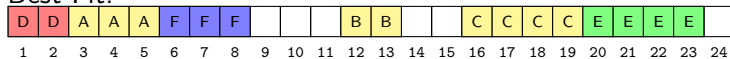
Ausgangspunkt:



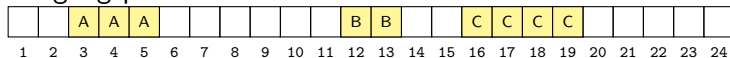
First Fit:



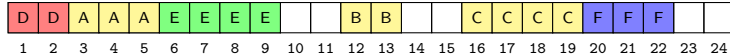
Best Fit:



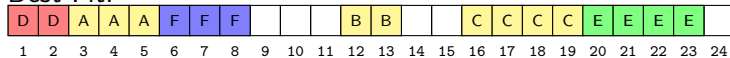
Ausgangspunkt:



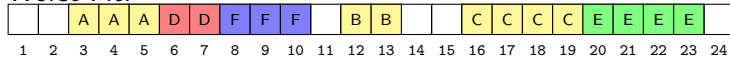
First Fit:



Best Fit:



Worst Fit:





Aenderungen



- $V1 \rightarrow V2$
 - Zahlreiche Theoriefolien erweitert. Thx to Bernhard Müller.