Systemprogrammierung Aufgabenblatt 3

Aufgabe 3.5

1. Unter Verdrängung ist der Entzug der Betriebsmittel von einem laufenden Prozess vor Ablauf seines Quantums zu verstehen. Mit Verdrängung lassen sich Schedulingziele besser erreichen, indem zb. beim Priority based Scheduling einem Prozess mit niedrigerer Priorität der Prozessor zugunsten eines Prozesses mit höherer Priorität entzogen wird.

<http://www.it-academy.cc/article/1284/SchedulingVerfahren.html>

Foilen Scheduling Seite 8

1. Zwischen den Scheduling Zielen den kürzeren Job zu bevorzugen und dem Ziel die längeren Jobs nicht verhungern zu lassen, bildet das HRRN Verfahren einen Kompromiss.

William Stallings: Betriebssysteme: Prinzipien und Umsetzung. 4. Auflage, Prentice-Hall, München 2002

1. Wenn sich ein niedrig priorisierter Prozess und ein hoch priorisierter Prozess dieselbe Ressource teilen (zb. Quellcode) und dem niedrig priorisierten Prozess wird die CPU, zugewiesen, dann wird der hoch priorisierte Prozess geblockt, da er keinen Zugang auf die Ressource hat. Ein mittel priorisierter Prozess könnte nun den niedrig Priorisierten Prozess verdrängen, obwohl der hoch priorisierte Prozess auf die Freigabe seiner Ressource wartet. Das führt dazu, dass der hoch priorisierte Prozess „verhungert“. Mithilfe der Prioritätsvererbung, würde der niedrig priorisierte Prozess die Priorität des hoch priorisierten Prozesses erben, sodass er den mittel priorisierten Prozess verdrängen kann und seine Ressource an den hoch priorisierten Prozess abgibt.
2. Beim off-line Scheduling haben wir schon vor der Laufzeit ein vollständiges Vorliegen von jeglichen Informationen. Es findet eine explizite Planung statt. Es kommen keine weiteren Prozesse hinzu. Voraussetzung hier ist das vollständige Vorliegen von Informationen (Prozessparameter bekannt, geschlossene Prozessmengen etc.)

Beim on-line Scheduling sind lediglich die aktuellen Prozesse bekannt, es können weitere Prozesse während der Laufzeit hinzu kommen. Entscheidungen müssen aufgrund unvollständiger Information gefällt werden. Es handelt sich hier um eine Implizite Planung. Voraussetzung hier ist, das die aktuellen Prozesse bekannt sind.

Folien Scheduling Seite 7

<https://www2.htw-dresden.de/~robge/ezs/vl/ezs-03-sched.pdf>

1. Bei einem Hard real-time System muss jede Deadline eingehalten werden, da Verspätungen zu kritischen Systemausfällen führen und somit nicht tolerierbar sind. Hierzu verwendet man oft off-line Algorithmen. Die Anforderung an so ein System ist nicht nur ein richtiges Ergebnis zu liefern, sondern dieses zur richtigen Zeit zu liefern.

Bei einem Soft real-time System führen Verspätungen lediglich zu einem Qualitätsverlust und sind deshalb nicht kritisch. Das Überschreiten der Deadline wird nicht als Versagen gewertet, wenn die Verspätung sich im Toleranzbereich befindet. Die Deadline kann hier als Richtlinie angesehen werden.

Beispiel für ein Hard real-time System: Airbag => Verspätung hier lebensgefährlich

Beispiel für ein soft real-time System: Videoanruf => Verspätung hier lediglich Qualitätsverlust

Folien Scheduling Seite 32-34

Hermann Kopetz: Real Time Systems. Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, Boston MA u. a. 1997

Aufgabe 3.6



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LCFS-PR | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| CPU | A | A | B | B | C | C | D | D | A | E | E | E | E | E | E | E | B | C | C | C | C | A | A |
| Queue(FIFO) |  |  | A | A | A  B | A  B | A  B  C | A  B  C | B  C | B  C  A | B  C  A | B  C  A | B  C  A | B  C  A | B  C  A | B  C  A | C  A | A | A | A | A |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HRRN | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| CPU | A | A | A | A | A | B | B | B | D | D | C | C | C | C | C | C | E | E | E | E | E | E | E |
| Queue(HRRN) |  |  | B | B | B  C | **B (2)**  C(1,17) | C  D | C  D | C(1,67)  **D(2)** | C  E | **C(2)**  E(1,14) | E | E | E | E | E | **E(2)** |  |  |  |  |  |  |

Der RR steht jeweils in den Klammern. Fett gedruckte Prozesse sind nicht mehr Teil der Warteschlange.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MLF | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| CPU | A | A | A | B | C | B | B | D | C | C | E | D | E | E | A | A | C | C | C | E | E | E | E |
| t=1 | **A** |  | B | **B** | **C** |  | D | **D** |  | E | **E** |  |  |  |  |  |  |  |  |  |  |  |  |
| t=2 |  | **A** | **A** |  | B | **B**  C | **B**  C | C | **C**  D | **C**  D | D | **D**  E | **E** | **E** |  |  |  |  |  |  |  |  |  |
| t=4 |  |  |  | A | A | A | A | A | A | A | A  C | A  C | A  C | A  C | **A**  C  E | **A**  C  E | **C**  E | **C**  E | **C**  E | **E** | **E** | **E** | **E** |



LCFS-PR

A:

Wartezeit=18

Antwortzeit=23

B:

Wartezeit=12

Antwortzeit=15

C:

Wartezeit=11

Antwortzeit=17

D:

Wartezeit=0

Antwortzeit=2

E:

Wartezeit=0

Antwortzeit=7

Durchschnittliche Wartezeit= 8,2

Durchschnittliche Antwortzeit= 12,8

HRRN:

A:

Wartezeit=0

Antwortzeit=5

B:

Wartezeit=3

Antwortzeit=6

C:

Wartezeit=6

Antwortzeit=12

D:

Wartezeit=2

Antwortzeit=4

E:

Wartezeit=7

Antwortzeit=14

Durchschnittliche Wartezeit= 3,6

Durchschnittliche Antwortzeit= 8,2

MLF:

A:

Wartezeit=1

Antwortzeit=16

B:

Wartezeit=2

Antwortzeit=5

C:

Wartezeit=9

Antwortzeit=16

D:

Wartezeit=4

Antwortzeit=6

E:

Wartezeit=7

Antwortzeit=7

Durchschnittliche Wartezeit=4,6

Durchschnittliche Antwortzeit=10