

Übungsblatt 8

Aufgabe 1 (Unterbrechungen)

1. Was sind Interrupts?
2. Was ist der Unterbrechungsvektor (Interrupt Vector)?
3. Was sind Exceptions?
4. Was passiert, wenn es während der Behandlung eines Interrupts zu einem weiteren Interrupt kommt?

Aufgabe 2 (Schedulingverfahren)

1. Warum existiert in einigen Betriebssystemen ein Leerlaufprozess?
2. Erklären Sie den Unterschied zwischen präemptivem und nicht-präemptivem Scheduling.
3. Nennen Sie einen Nachteil von präemptivem Scheduling.
4. Nennen Sie einen Nachteil von nicht-präemptivem Scheduling.
5. Wie funktioniert statisches Multilevel-Scheduling?
6. Wie funktioniert Multilevel-Feedback-Scheduling?
7. Welche Schedulingverfahren sind fair?

Ein Schedulingverfahren ist „fair“, wenn jeder Prozess irgendwann Zugriff auf die CPU erhält.

- | | |
|--|--|
| <input type="checkbox"/> Prioritätengesteuertes Scheduling | <input type="checkbox"/> Shortest Remaining Time First |
| <input type="checkbox"/> First Come First Served | <input type="checkbox"/> Longest Remaining Time First |
| <input type="checkbox"/> Last Come First Served | <input type="checkbox"/> Highest Response Ratio Next |
| <input type="checkbox"/> Round Robin mit Zeitquantum | <input type="checkbox"/> Earliest Deadline First |
| <input type="checkbox"/> Shortest Job First | <input type="checkbox"/> Fair-Share |
| <input type="checkbox"/> Longest Job First | |

8. Welche Schedulingverfahren arbeiten präemptiv (= *unterbrechend*)?

- | | |
|--|---|
| <input type="checkbox"/> First Come First Served | <input type="checkbox"/> Longest Remaining Time First |
| <input type="checkbox"/> Round Robin mit Zeitquantum | <input type="checkbox"/> Fair-Share |
| <input type="checkbox"/> Shortest Job First | <input type="checkbox"/> Statisches Multilevel-Scheduling |
| <input type="checkbox"/> Longest Job First | <input type="checkbox"/> Multilevel-Feedback-Scheduling |
| <input type="checkbox"/> Shortest Remaining Time First | |

9. Bei welchen Schedulingverfahren muss die CPU-Laufzeit (= *Rechenzeit*) bekannt sein?

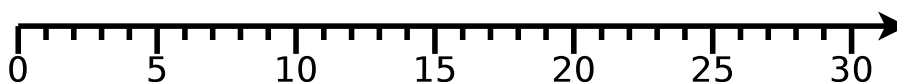
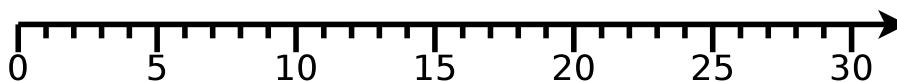
- | | |
|--|--|
| <input type="checkbox"/> Prioritätengesteuertes Scheduling | <input type="checkbox"/> Shortest Remaining Time First |
| <input type="checkbox"/> First Come First Served | <input type="checkbox"/> Longest Remaining Time First |
| <input type="checkbox"/> Last Come First Served | <input type="checkbox"/> Highest Response Ratio Next |
| <input type="checkbox"/> Round Robin mit Zeitquantum | <input type="checkbox"/> Earliest Deadline First |
| <input type="checkbox"/> Shortest Job First | <input type="checkbox"/> Fair-Share |
| <input type="checkbox"/> Longest Job First | |

Aufgabe 3 (Scheduling)

Prozess	CPU-Laufzeit [ms]	Ankunftszeit [ms]
A	5	0
B	7	3
C	2	5
D	6	7
E	1	10
F	5	18
G	4	24

1. Auf einem Einprozessorrechner sollen sieben Prozesse mit unterschiedlichen Ankunftszeiten verarbeitet werden. Skizzieren Sie die Ausführungsreihenfolge der Prozesse mit einem Gantt-Diagramm (Zeitleiste) für...

- Longest Remaining Time First und
- Shortest Remaining Time First.

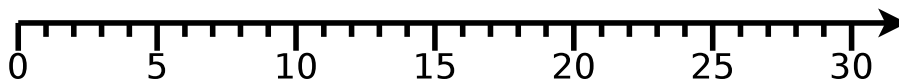
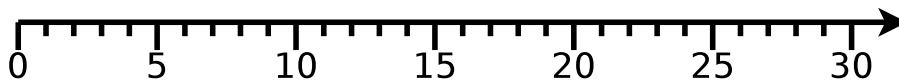
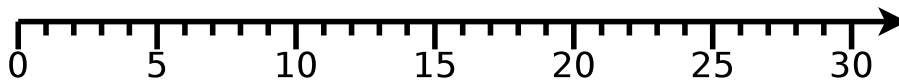
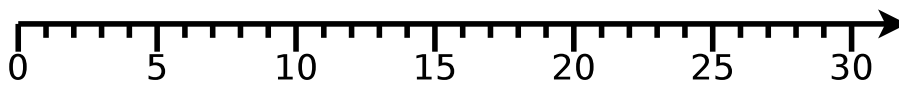
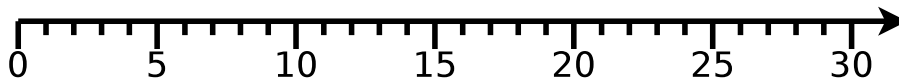


2. Berechnen Sie die mittleren Laufzeiten der Prozesse.
3. Berechnen Sie die mittleren Wartezeiten der Prozesse.

Aufgabe 4 (Scheduling)

Prozess	CPU-Laufzeit [ms]	Ankunftszeit [ms]
A	10	0
B	8	4
C	2	6
D	5	11
E	4	13
F	1	15

1. Auf einem Einprozessorrechner sollen sechs Prozesse mit unterschiedlichen Ankunftszeiten verarbeitet werden. Skizzieren Sie die Ausführungsreihenfolge der Prozesse mit einem Gantt-Diagramm (Zeitleiste) für...
 - First Come First Served,
 - Shortest Job First,
 - Longest Job First,
 - Shortest Remaining Time First und
 - Longest Remaining Time First.



2. Berechnen Sie die mittleren Laufzeiten der Prozesse.
3. Berechnen Sie die mittleren Wartezeiten der Prozesse.

Aufgabe 5 (Shell-Skripte)

1. Schreiben Sie ein Shell-Skript, das den Benutzer bittet, eine der vier Grundrechenarten auszuwählen. Nach der Auswahl einer Grundrechenart wird der Benutzer gebeten, zwei Operanden einzugeben. Die beiden Operanden werden mit der zuvor ausgewählten Grundrechenart verrechnet und das Ergebnis in der folgenden Form ausgegeben:

<Operand1> <Operator> <Operand2> = <Ergebnis>

2. Ändern Sie das Shell-Skript aus Teilaufgabe 1 dahingehend, dass für jede Grundrechenart eine eigene Funktion existiert. Die Funktionen sollen in eine externe Funktionsbibliothek ausgelagert und für die Berechnungen verwendet werden.
3. Schreiben Sie ein Shell-Skript, das eine bestimmte Anzahl an Zufallszahlen bis zu einem bestimmten Maximalwert ausgibt. Nach dem Start des Shell-Skripts, soll dieses vom Benutzer folgende Parameter interaktiv abfragen:

- Maximalwert, der im Zahlenraum zwischen 10 und 32767 liegen muss.
- Gewünschte Anzahl an Zufallszahlen.

4. Schreiben Sie ein Shell-Skript, das die folgenden leeren Dateien erzeugt:

image0000.jpg, image0001.jpg, image0002.jpg, ..., image9999.jpg

5. Schreiben Sie ein Shell-Skript, das die Dateien aus Teilaufgabe 4 nach folgendem Schema umbenennt:

BTS_Übung_<JAHR>_<MONAT>_<TAG>_0000.jpg
BTS_Übung_<JAHR>_<MONAT>_<TAG>_0001.jpg
BTS_Übung_<JAHR>_<MONAT>_<TAG>_0002.jpg
...
BTS_Übung_<JAHR>_<MONAT>_<TAG>_9999.jpg