Übungsblatt 8

Aufgabe 1 (Schedulingverfahren)

1. Beschreiben Sie die Aufgabe eines Leerlaufprozesses.
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. Nennen Sie das Scheduling-Verfahren, das Windows-Betriebssysteme implementieren.
6. Nennen Sie ein Scheduling-Verfahren, das Linux-Betriebssysteme implementieren.
7. Erklären Sie, wie der Completely Fair Scheduler des Linux-Kernels (Kernel 2.6.23 bis Kernel 6.5.13) funktioniert.
   (Hinweis: Ein Schaubild kann hier helfen!)
8. Beschreiben Sie, wie das Multilevel-Feedback-Scheduling funktioniert.
9. Beschreiben Sie was es bedeutet, wenn ein Schedulingverfahren fair ist.
10. Markieren Sie die fairen Schedulingverfahren.

    ☐ Prioritätengesteuertes Scheduling
    ☐ First Come First Served
    ☐ Round Robin mit Zeitquantum
    ☐ Earliest Deadline First
    ☐ Fair-Share

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

    ☐ First Come First Served
    ☐ Round Robin mit Zeitquantum
    ☐ Fair-Share
    ☐ Multilevel-Feedback-Scheduling

Aufgabe 2 (Scheduling)

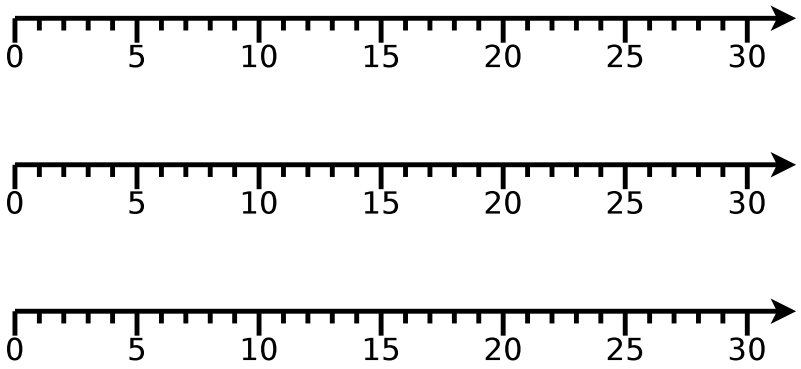
Auf einem Einprozessorrechner (mit nur einem CPU-Kern) sollen fünf Prozesse verarbeitet werden. Alle Prozesse sind zum Zeitpunkt 0 im Zustand `bereit`. Hohe Prioritäten sind durch hohe Zahlen gekennzeichnet.

Prozess	Rechenzeit	Priorität
A	5 ms	15
B	10 ms	5
C	3 ms	4
D	6 ms	12
E	8 ms	7

Skizzieren Sie die Ausführungsreihenfolge der Prozesse mit einem Gantt-Diagramm (Zeitleiste) für **Round Robin** (Zeitquantum $q = 1\,\text{ms}$), **FCFS** und **Prioritätengesteuertes Scheduling**.

Die Spalte Priorität in der Tabelle ist nur für das Prioritätengesteuerte Scheduling relevant und nicht für Round Robin or FCFS.

Berechnen Sie die mittleren Laufzeiten und mittleren Wartezeiten der Prozesse.



Die Rechenzeit ist die Zeit, die der Prozess Zugriff auf die CPU benötigt, um komplett abgearbeitet zu werden.

Laufzeit = „Lebensdauer“ = Zeitspanne zwischen dem Anlegen und Beenden eines Prozesses = (Rechenzeit + Wartezeit).

Laufzeit	A	B	C	D	E
RR					
FCFS					
Prioritätengesteuertes Scheduling					

Die Wartezeit ist die Zeit in der `bereit`-Liste.

Wartezeit = Laufzeit - Rechenzeit.

Wartezeit	**A**	**B**	**C**	**D**	**E**
RR					
FCFS					
Prioritätengesteuertes Scheduling					

Aufgabe 3 (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
   ```