Exercise 3 - Scheduling

Question 3.1: Asynchronous Programming

Figure 3.1.1 (cf. solution sheet) depicts the desired temporal behavior of 4 computation tasks P1 to P4.

a) Draw into figure 3.1.2 the actual sequence of the tasks according to the asynchronous programming method (up to the normalized time t/T = 10).

b) Draw into figure 3.1.3 the temporal sequence of task states of task P4, resulting from the asynchronous programming in figure 3.1.2.

Question 3.2: Scheduling - Methods

Figure 3.2.1 depicts the desired behavior of 3 computation tasks P1 to P3. We want to have a look at the execution of this tasks if different methods for the allocation of the processor (scheduling-methods) are applied.

a) Draw into figures 3.2.2 to 3.2.6 the actual execution sequence of the tasks (up to the normalized time t/T = 6) which results, if one applies the following scheduling methods: fist-in-fist-out (FIFO), fixed priorities, earliest-deadline-first (EDF), and least-laxity (LL). For the fixed-priorities-method the two cases of priority assignment given in table 3.2.1 have to be taken into account.

b) Determine for each method the average time-deviation of every task from the desired behavior, as well as, the average time-deviation of all tasks.

Task	Case 1	Case 2
P1	1	2
P2	2	3
P3	3	1

Table 3.2.1: Priorities of the tasks for the fixed-priorities-method

Solution sheet for question 3.1

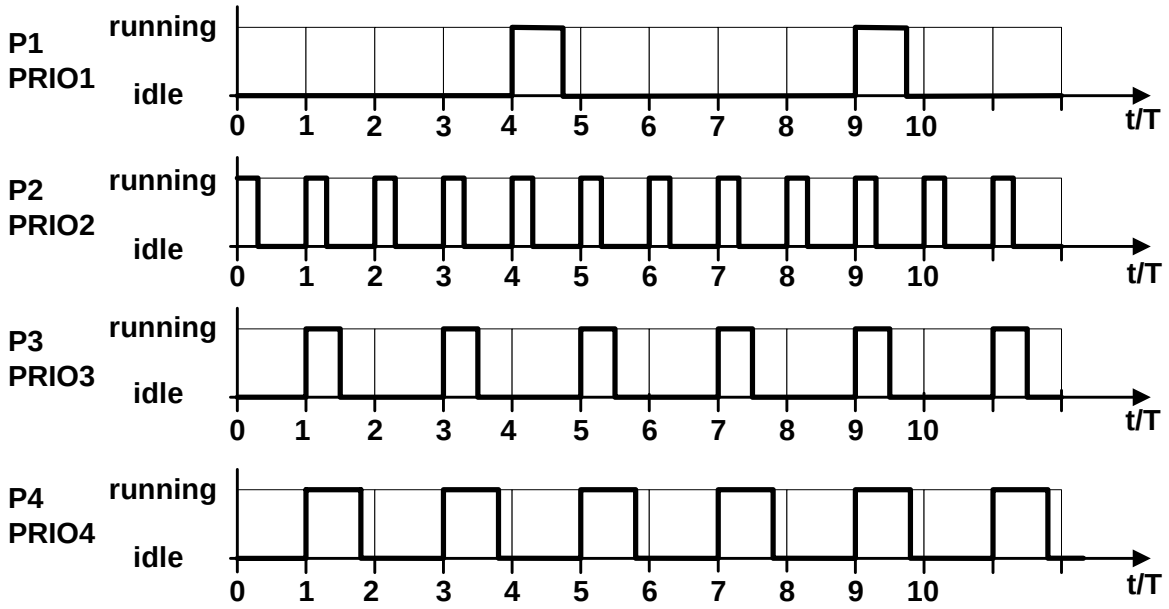


Figure 3.1.1: Desired temporal behavior of the tasks

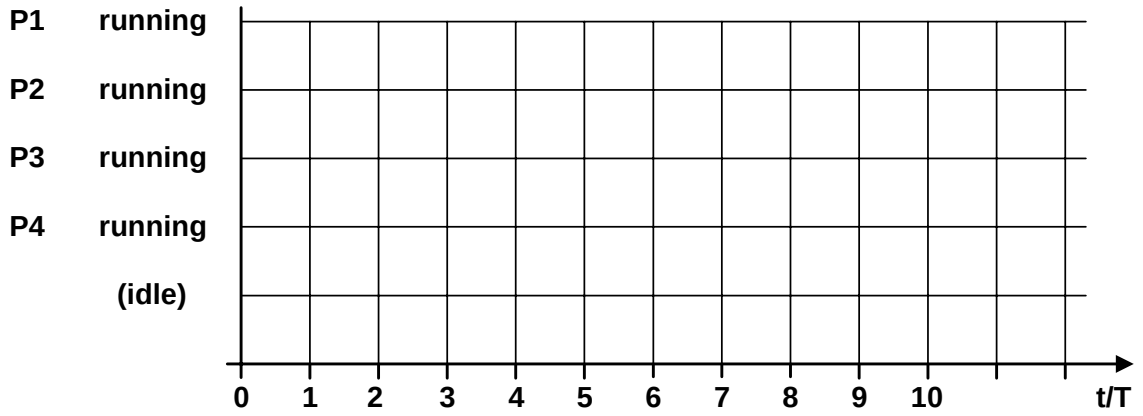


Figure 3.1.2: Actual sequence of the tasks

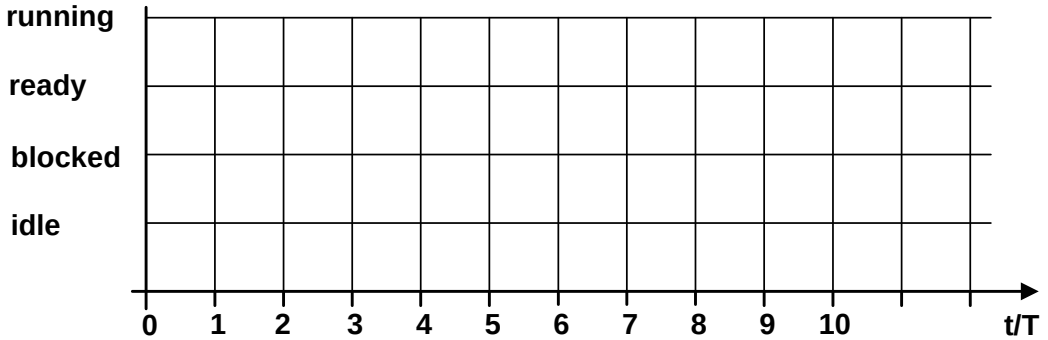


Figure 3.1.3: States of the task P4 (actual behavior)

Solution sheet 1 for question 3.2

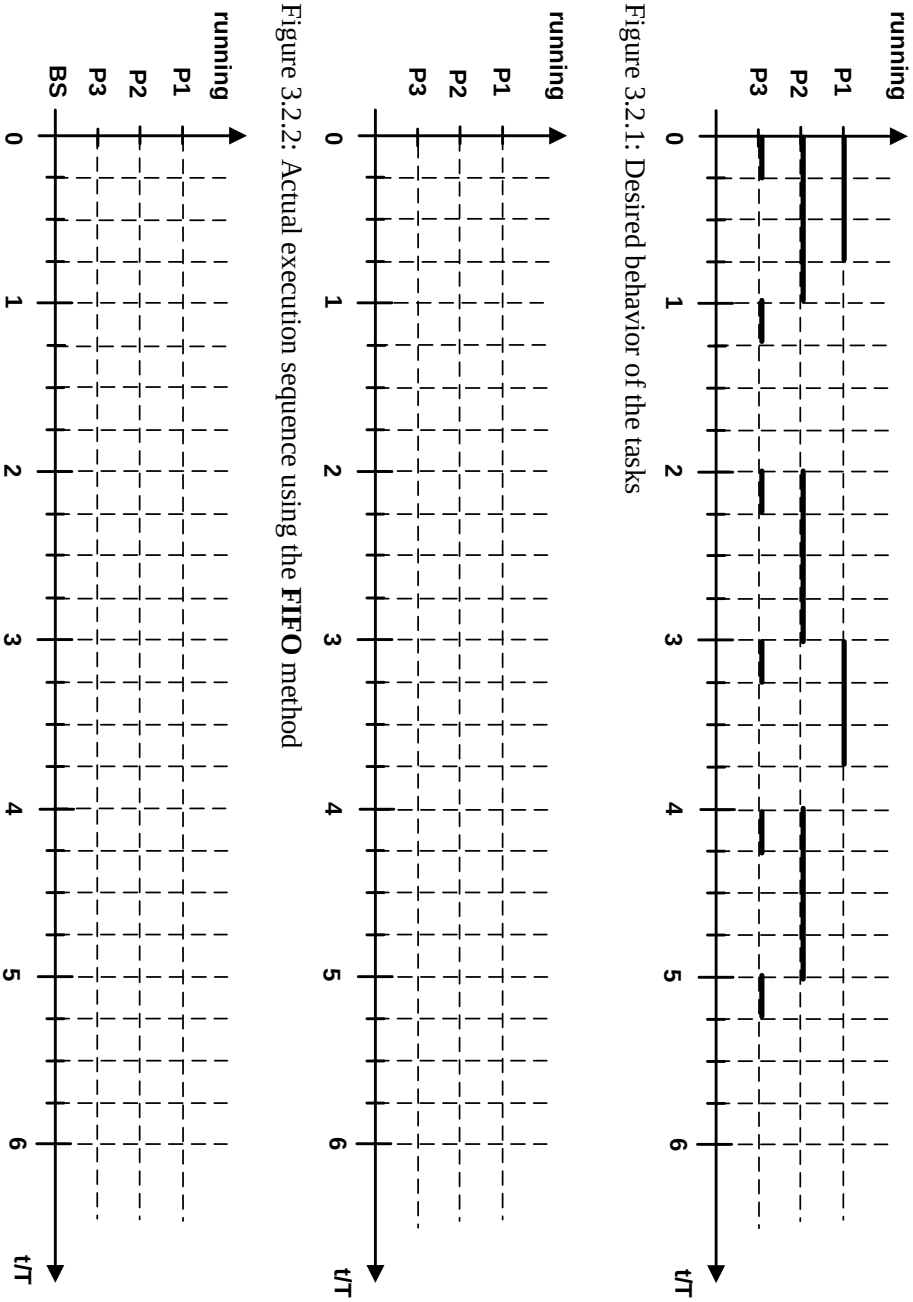
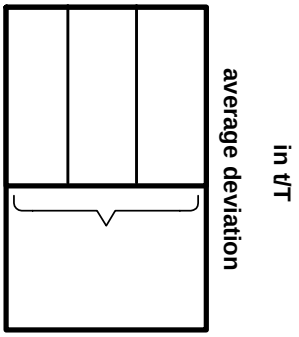
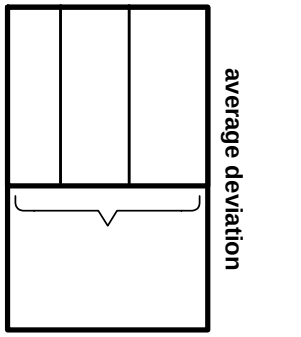


Figure 3.2.1: Desired behavior of the tasks

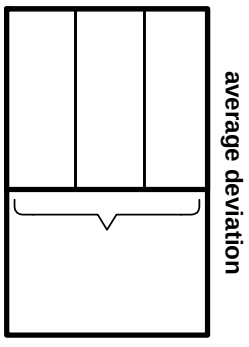
Figure 3.2.2: Actual execution sequence using the **FIFO** method

Figure 3.2.3: Actual execution sequence using the **fixed priorities** method (case 1)





Solution sheet 2 for question 3.2



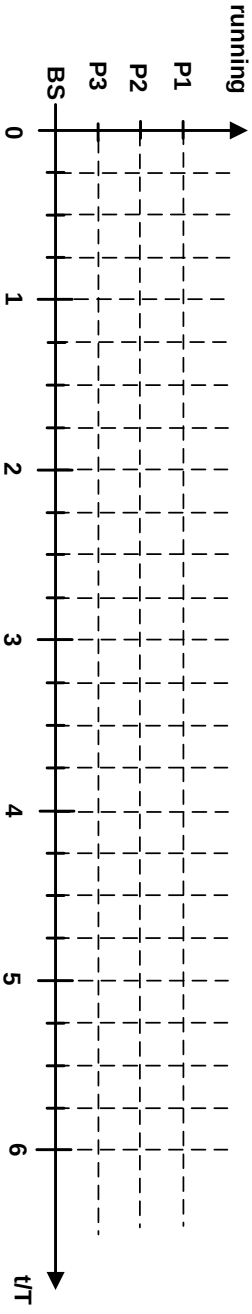
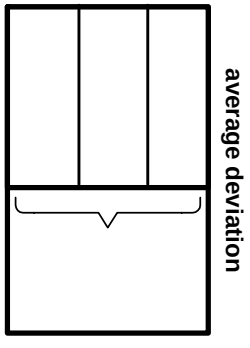


Figure 3.2.4: Actual execution sequence using the **fixed priorities** method (case 2)



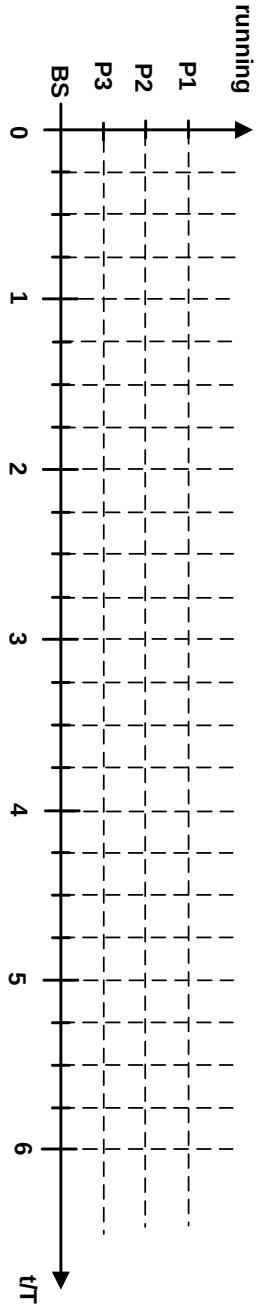
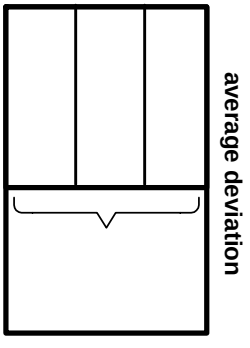


Figure 3.2.5: Actual execution sequence using the **earliest-deadline-first** method



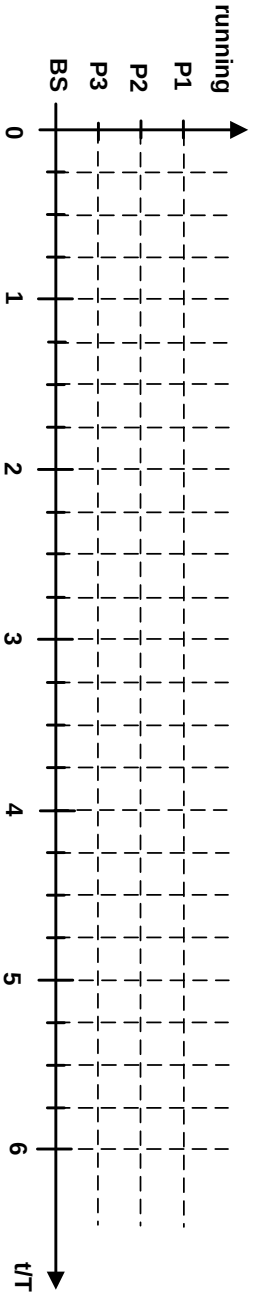


Figure 3.2.6: Actual execution sequence using the **least-laxity** method