Operating Systems 2016/17
Solutions for Assignment 5

Prof. Dr. Frank Bellosa
Dipl.-Inform. Marc Rittinghaus

T-Question 5.1: Scheduling

a. Why are the scheduler and dispatcher good examples for the distinction between policy and mechanism? **1 T-pt**

Solution:
The CPU scheduler selects the next process to run and thus only implements a policy. It does not perform the actual process switch (saving/restoring process context, etc.). That is done by the dispatcher.

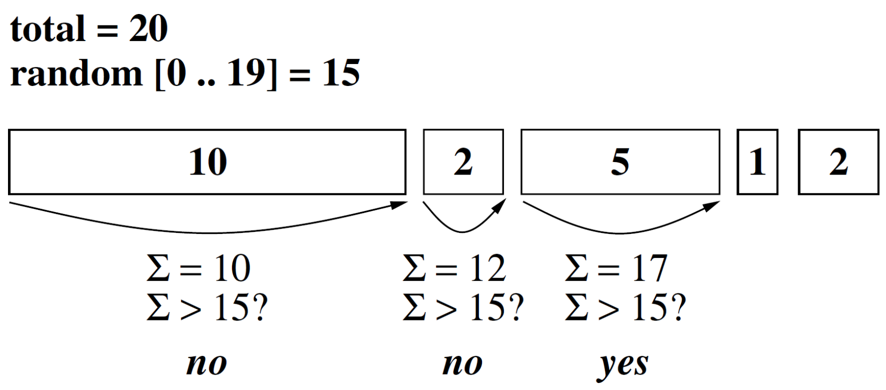
b. Briefly explain the difference between cooperative and preemptive scheduling? What problem does preemptive scheduling solve? **2 T-pt**

Solution:
In cooperative multitasking, the currently running process must explicitly invoke the `yield()` *system call to ask the kernel to switch to another process. With preemptive scheduling, a periodic timer interrupt triggers scheduling decisions (a* `yield()` *system call may also be available).*

Preemptive scheduling thus solves the problem that a single process can (inadvertently) hog the CPU and prevent other processes from running. It also relieves the application programmer from having to deal with process scheduling.

c. With lottery scheduling, every process is assigned a certain number of tickets. To make a scheduling decision the lottery scheduler randomly chooses a ticket and selects the process that owns the ticket. Briefly explain how lottery scheduling can be implemented without allocating any dedicated objects per ticket such as structs, integers, or array elements. **2 T-pt**

Solution:

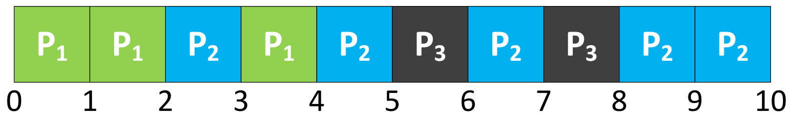


Each process is assigned a number of tickets (single integer in PCB). When the scheduler is triggered, a random ticket number in the range between 0 and the total number of tickets is generated. The scheduler than iterates over the list of ready processes and adds up each process's ticket amount. If the sum becomes greater than the chosen random ticket, the current process must own the ticket and the lottery scheduler consequently selects it for dispatching.

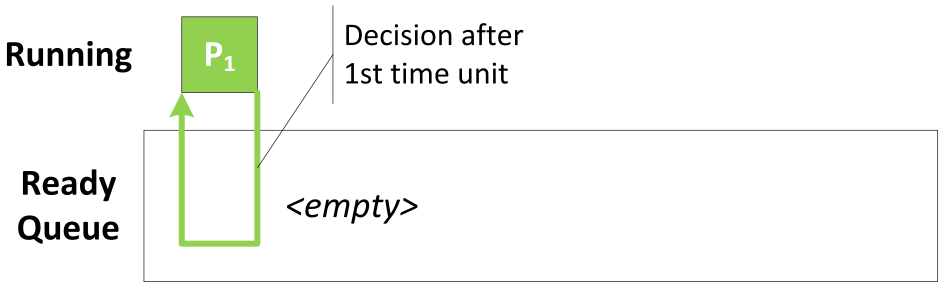
d. Give the scheduling sequence (e.g., P_X, P_Y, P_Z,...) for the following processes with round robin scheduling and a timeslice length of 1 time unit. The scheduler first adds new processes (if any) to the tail of the ready queue and then inserts the previous process to the tail (if it is still runnable). **2 T-pt**

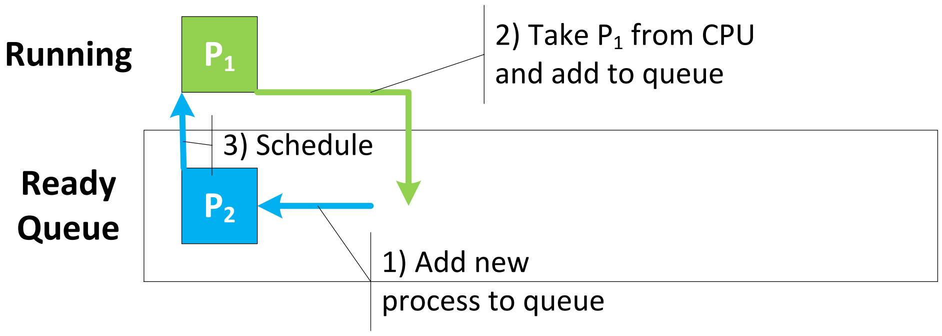
Process	Burst length	Arrival time
P_1	3	0
P_2	5	2
P_3	2	4

Solution:
The resulting round robin schedule is:
P_1, P_1, P_2, P_1, P_2, P_3, P_2, P_3, P_2, P_2



With the round robin policy, the scheduler preempts the current process after each time slice and selects a different thread to run. The table gives the start time and computation time for each process. At time units 0 and 1, only P_1 is present in the ready queue and is thus selected for scheduling. At time unit 2, process P_2 appears. When the scheduler is invoked, P_1 is still running, the ready queue is thus empty and P_2 becomes the only element in the queue. Then P_1 the CPU is taken away from P_1 and P_1 is added to the tail of the ready queue: [P_2, P_1]. Consequently, the scheduler selects P_2 as next process. After a time unit, the queue is [P_1, P_2] and P_1 runs again. This scheme is followed to the end.





e. Calculate the average waiting time for the example in 5.1d. **1 T-pt**

Solution:
The waiting time of a process denotes the number of timeslices a process spent in the ready queue. A timeslice is accounted to the waiting time, if the process is ready to run (i.e., not blocked and waiting for an event or I/O completion), however, the scheduler decided to select a different process for execution or a different process is still running and the scheduler has not been invoked, yet.

To compute the waiting time, we can manually count the timeslices by looking at the scheduling sequence or use the following formula:

$$T^W_{P_i} = ((L_{P_i} + l_{ts}) - A_{P_i}) - B_{P_i}$$

l_{ts} *:= Length of a time slice*
L_{P_i} *:= Last time slice of process* P_i
A_{P_i} *:= Arrival time*
B_{P_i} *:= Burst time*

Accordingly, the solution is:

$$T^W_{P_1} = ((3+1) - 0) - 3 = 1, T^W_{P_2} = ((9+1) - 2) - 5 = 3, T^W_{P_3} = ((7+1) - 4) - 2 = 2$$

$$T_{avg} = \frac{T^W_{P_1} + T^W_{P_2} + T^W_{P_3}}{3} = \frac{1+3+2}{3} = \frac{6}{3} = 2$$

Total:
8 T-pt