Exercise Multimedia Technology
WS 2003/2004

Sheet 12 (January 30th, 2004)

Aufgabe 12.1: Scheduling

(a) What kinds of tasks does a scheduler have to perform in an operating system?

A scheduler decides which inactive process is being chosen as ready next.

(b) Comment on the specific constraints a scheduler has to fulfill in the context of processing continuous data-streams.

Other than traditional processes, processes which handle continous data streams have deadlines, which have to be met. A schaduler should try to meet deadlines in the first place.

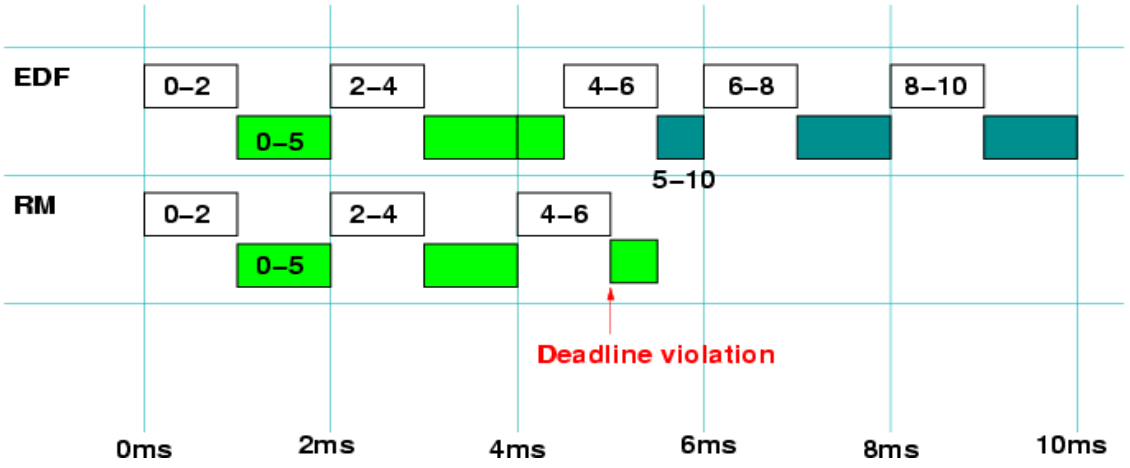
(c) Explain the technique of EDF-scheduling and rate-monotonic scheduling.

EDF-scheduling: The process with the earliest deadline gets to highest priority.

Rate-monotonic scheduling: The process with the highest packet rate gets the highest priority.

(d) In the following table the arrival times and deadlines of two processes are given , each of which processes a data-stream. A packet from process 1 can be processed within 1 ms and a packet from process 2 can be processed within 2.5 ms. Insert the order of processing according to the RM (rate monotonic) scheme, resp. the EDF-scheme into the following figure. The scheduler is able to switch tasks every 0.5 ms.

packet	Arrival process 1	deadline P1	Arrival process 2	Deadline P2
1	0 ms	2 ms	0 ms	5 ms
2	2 ms	4 ms	5 ms	10 ms
3	4 ms	6 ms	-	-
4	6 ms	8 ms	-	-
5	8 ms	10 ms	-	-



In what periods do you encounter violations with regard to the response time using EDF?

no problems

In what periods do you encounter violations with regard to the response time using RM?

Violation of response time in period 3 (4 – 6ms)

Aufgabe 12.2: Videoserver

A video-on-demand server uses non-preemptive scheduling. Its purpose is to send video-data continuously to its receivers. Within 10ms the server is able to deliver a mean amount of data of about 100kByte. A video-stream has an average data-rate of about 2Mbit/s. Switching from one process to another takes 5 ms. What is the maximum number of streams the server can deliver in parallel?

Note: 1kByte = 1000 bytes and 1 Mbit = 1000000 bits. A stream has to be fed with data at least once a second.

Total capacity: 100kByte in 10ms = 10.000kByte/second

Capacity for one stream: 2Mbit/s = 250kByte/s

Switching processes needs 5ms . So the lost capacity of 5ms =10.000*5/1000=50kByte

n = number of streams

n250+n50=10.000 => n=33.33

A total number of 33 streams can be processed at a time.

Aufgabe 12.3: Multimedia data storage

Consider the order of requested blocks of a harddrive in the next figure. Process the request order using the scan-EDF approach with modified deadlines. Write your results into the following diagram. In order to process the modified deadlines choose an appropriate function.

Given:

- Ni as the trach which shall be accessed next time
- A as the current track of the read/write head
- Nmax as the highest track
- Di the deadline (integer) of request i

$$f(N_i) = (1/N_{max}) * \begin{pmatrix} head\ moving\ downwards \wedge N_i < A : A - N_i \\ head\ moving\ downwards \wedge N_i \geq A : N_i \\ head\ moving\ upwards \wedge N_i > A : N_i - A \\ head\ moving\ upwards \wedge N_i < A : N_{max} - N_i \end{pmatrix}$$

The modified deadline Ddi can be defined as follows:

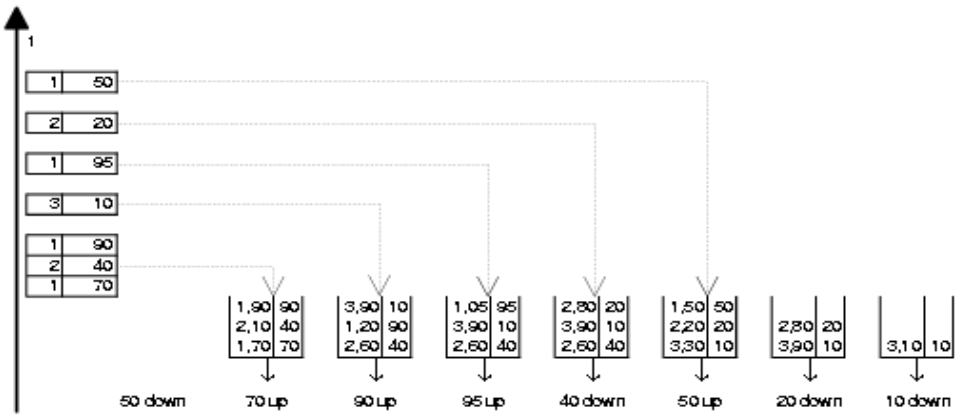
DDi= Di + f(Ni)

Properties the of function DDi:

The function given above exactly defines the requirements for Scan-EDF:

- Requests with a low deadline (number) are processed first as deadlines are integers. The only thing which we actually influence are the decimal places.
- The deadlines are constructed such that requests with equal deadlines are processed in the current scan-order.
- ... that means if the head is moving downwards, the purpose the function f() is to assign low values to requests Ni below the current track position A and higher values to requests Ni above the current position A. Why? Because we don't want to chance dircetion unless

it is necessary. The same idea applies to head movements into the other direction.

- 

Choose the state of the read/write-head after processing the last query.

10 down

Aufgabe 12.4: Threads and Processes

1. Discuss advantages and disadvantages of threads versus processes.

   Advantages of threads:

- They are more lightweight, switching between contexts (the environment in which the program is operating) is faster as we only need a unique stackpointer for each thread.
- They share a common memory environment with other threads. That makes communication easier (messages don't have to be copied).
- The cache of the processor stays valid. No page- or segment-table manipulations are necessary.

  Disadvantages of threads

- No memory protection, so the synchronization of threads is an important issue.

2. Find one application which is better suited to using threads and one to using processes.

   Thread: GUI-Programming in a model-view-controller fashion. A server-demon producing instances of the server for each network-request.

   Processes: Split Netscape, composer and it's mailer into different processes. So one crashing application does not influence the others.

Aufgabe 12.5: Time-Warping

The following signal is given: 4, 8, 1, 5, 2

Match the signal to one of those examples provided below using time warping. Each signal has to be considered from the first to the last sample (don't omit any of them). Which signal fits best?

Signal 1: 4, 4, 8, 1, 1, 5, 5, 2

Signal 2: 9, 3, 7, 5, 2