Advanced Operating Systems
Assignment Week 11

Paul Lödige
Student ID: 37-229753

January 3, 2023

Acronyms **1**

1 Assignment **1**

2 EDCL (Earliest Deadline Critical Laxity) **2**

3 EDF-WM (Earliest Deadline First with Windowconstraint Mitigation) **2**

Acronyms

EDCL Earliest Deadline Critical Laxity 1, 2
EDF Earliest Deadline First 1, 2
EDF-WM Earliest Deadline First with Windowconstraint Mitigation 1, 2
EDZL Earliest Deadline Zero Laxity 2

1 Assignment

Summarize the EDCL algorithm and the EDF-WM algorithm within 400 words, respectively, as compared to the traditional EDF algorithm.

2 EDCL (Earliest Deadline Critical Laxity)

The EDCL algorithm is a derivative of the EDZL algorithm which in turn is a modified version of the EDF algorithm. Because of Dhall's effect the EDF algorithm is non-optimal for scheduling in a multiprocessor environment. EDZL fixes Dhall's effect by promoting any task to the highest priority when its laxity reaches zero. In order for this to work the scheduler has to use a precise timer in order to detect the moment when any laxity reaches zero. EDCL foregoes the need for this precise timer by only checking the laxity at the release time or the completion time of any task. At this moment the scheduler calculates whether any unscheduled task has a laxity that is critical. The EDCL algorithm defines a task's laxity as critical if the minimal remaining execution time of the tasks with the closest deadline would not allow the unscheduled task to hold its deadline. In this case, the task with the critical laxity is promoted to top priority.

Pros

- Dhall's effect does not apply (better than EDF)
- easier to implement than EDZL while being comparable in schedulability

Cons

- non-optimal scheduling (worst-case processor utilization of 50%)

3 EDF-WM (Earliest Deadline First with Window-constraint Mitigation)

The EDF-WM algorithms can be classified as a semi-partitioning scheduling algorithm because it schedules most tasks with a fixed processor assignment and only partitions as few tasks as possible. The algorithm is based on EDF scheduling but avoids Dhall's effect. The algorithm assigns each new task to a specific processor as long as the deadline can be guaranteed by that processor. If the runtime of a new task is longer than the available the task is partitioned. Each partition is then scheduled on a different processor. The deadline of each partition relates to its position in the order of partitions. Additionally the EDF-WM algorithm only allows the scheduler to migrate each task once. This cuts down on the number of context switches while only slightly affecting its worst-case utilization.

Pros

- Dhall's effect does not apply (better than EDF)
- less context switching than other semi-partitioning scheduling algorithms

Cons

- non-optimal scheduling