Université libre de Bruxelles

Project 1

Global vs. Partitioned DM

**Aldar Saranov, Przemyslaw Gasinski**

[Aldar.Saranov@ulb.ac.be](mailto:Aldar.Saranov@ulb.ac.be)

[Przemyslaw.Gasinski@ulb.ac.be](mailto:Przemyslaw.Gasinski@ulb.ac.be)

INFO-F-404 Real-Time Operating Systems II (M-INFOS/F277)

Joël Goossens, Nikita Veshchikov

November 2016

# Software description

## Generator

### Input/output

Command in form

|  |
| --- |
| ./taskGenerator -u 70.0 -n 8 -o tasks.txt [-a 100.0] |

-u – Total utilisation in percent

-n – Number of tasks

-o – Output filepath/filename

-a – Approximate average WCET (optional argument, the higher it is, the more precisely utilisation will be attained. 100.0 by default if not specified)

Output is passed to the specified file. Every line is corresponds to a task. Order: Offset, Period, Deadline, WCET.

|  |
| --- |
| 56 181 134 61  10 181 150 142  38 181 152 132 |

### Generation algorithm

Generator uses following algorithm:

1. Calculate common period by formula
2. For each task generate offset according to uniform distribution from some (hardcoded) and .
3. Distribute the ordered over each of tasks.
   1. Set .
   2. Set .
   3. If
   4. If
   5. Ensure that the rest tasks will get by 100% utilization each in worst case (extra constraint for
   6. Same (c) and (d) for .
   7. Generate utilization for the next task according uniform distribution from to .
   8. Calculate and decrement .
4. For each task calculate .
5. For each task generate according to uniform distribution from and .

## Simulator

### Input/output

### Simulation algorithm

## Study

### Input/output

# Difficulties

## Generation fraction wcets

Initially it was proposed to fix the period (for each task) as a constant during task generation. The problem of such approach may be encountered if user sets too large value of or too small . In such case wcets generated at the next step often were fractions between 0 and 1 and therefore were being truncated due to integer type. The actual utilization deviated from the ordered too much.

## Generated task overload

## Too large hyper period

## Too large simulation timeline

Overflow bugs resolved by migrating from unsigned int to long long.

## Calculating the minimal number of processors required

# Test planning

# Simulation analysis

## Obtained results

## Conclusion

# Appendix