Estimating Operating System Resource Occupation by Simulation

Bernd Däne

Bernd.Daene@TU-Ilmenau.de

Falk Berger, Wolfgang Fengler

Ilmenau Technical University, Germany

Topics

1. Introduction
2. Model Overview
3. Model Details
4. Simulation and Evaluation
5. Conclusion

Parts of this work were and are supported by Deutsche Forschungsgemeinschaft (SFB 622).

1. Introduction

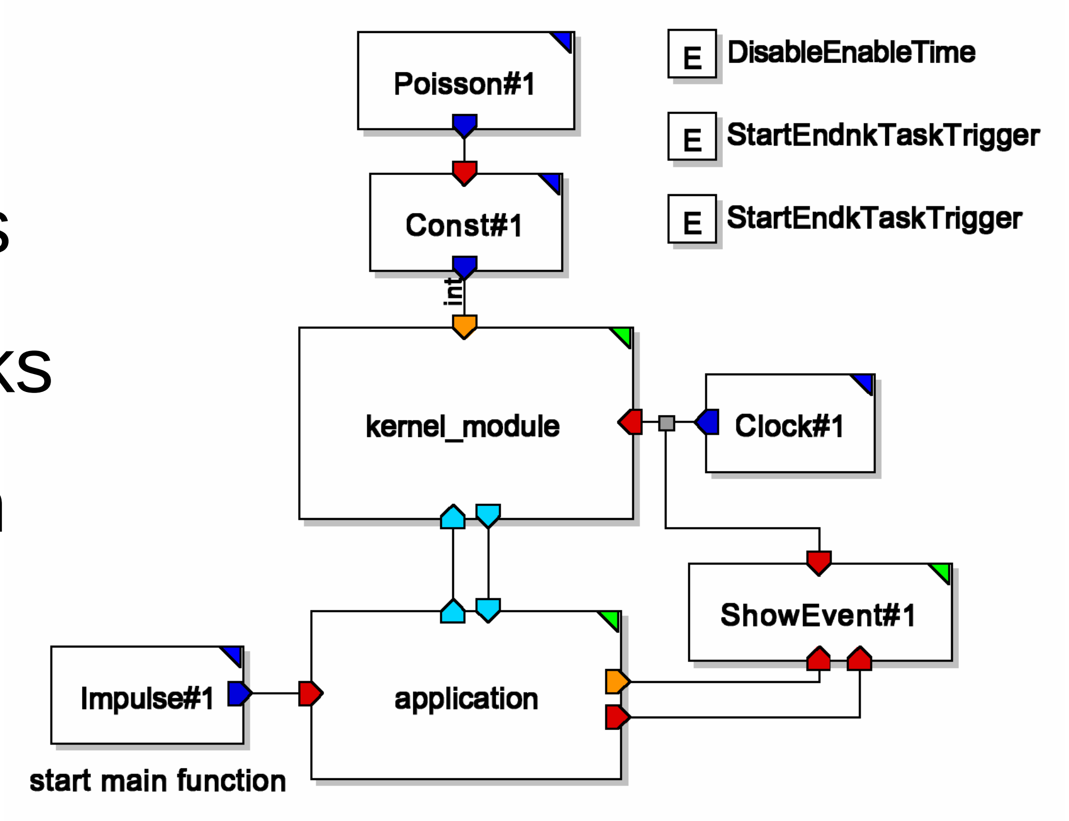
- Modeling Real Time Operating Systems (RTOS)
- Kernel and application levels
- Goals:
  - Functional validation
  - Quantitative estimation of properties
- Hierarchical discrete-event models
- MLDesigner tool used

Operating System *e*RTOS

- Study based on special system ‘eRTOS’
- Developed for high performance DSP systems
- Multiple scheduling strategies:
    - Rate monotonic
    - Preemptive
- Resource management:
    - Device, memory, message
- Circular memory buffer (FIFO)

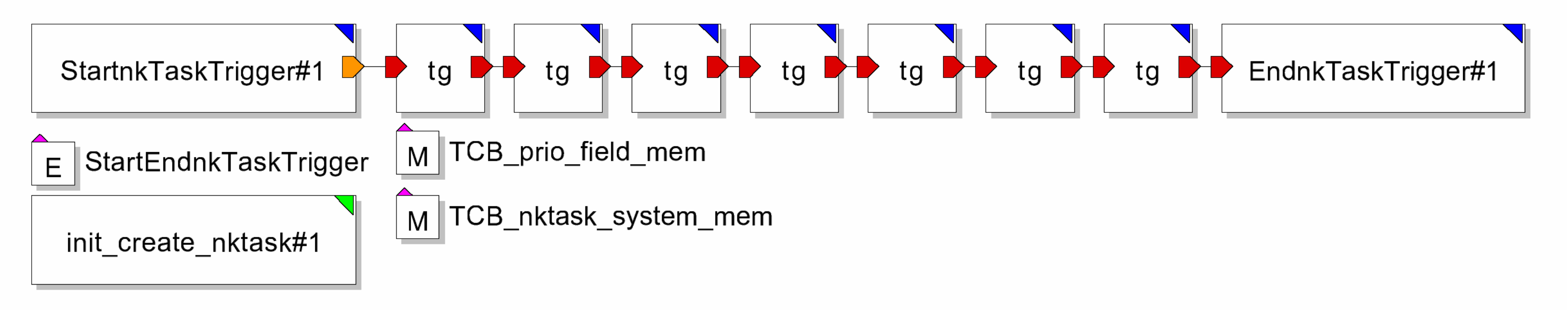
2. Model Overview

- DE domain
- Parts:
  - Kernel modules
  - Application tasks
  - Instrumentation (e.g. triggers, displays)



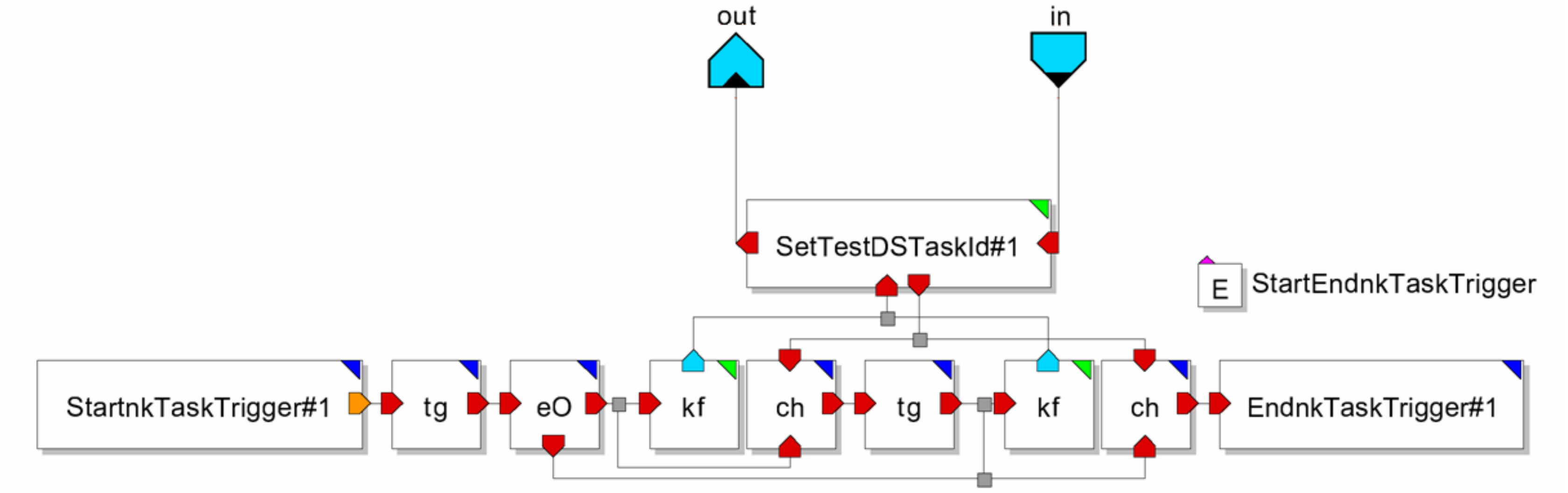
3. Model Details: Basic Task Model

- Atomic blocks with known time consumption
- Task switch at block boundaries only
- Time info collected by instrumentation blocks (not shown)



Complex Task Model

- Branches and forks (here: *eO*)
- Kernel function calls: *kf, ch*
- Call and return ‘busses’ carry named events

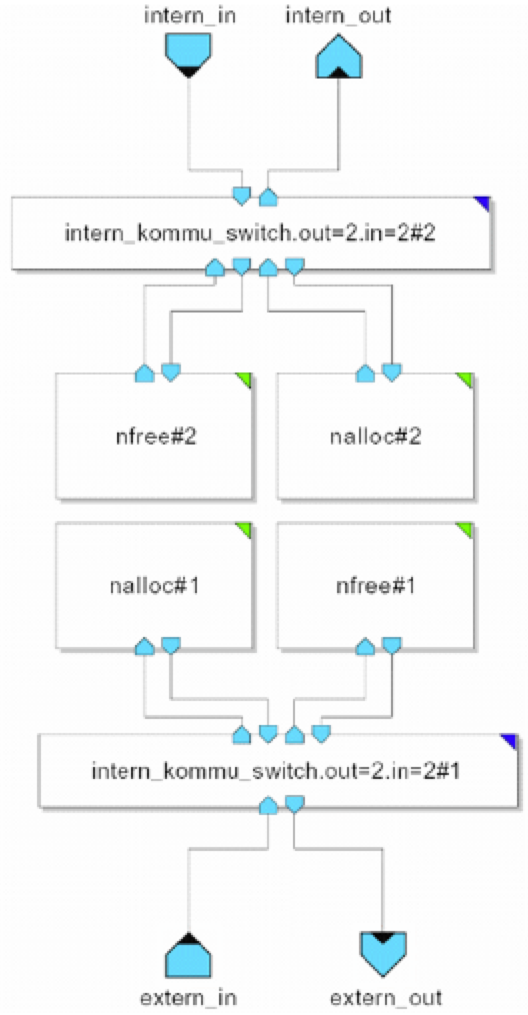


Kernel Model Overview

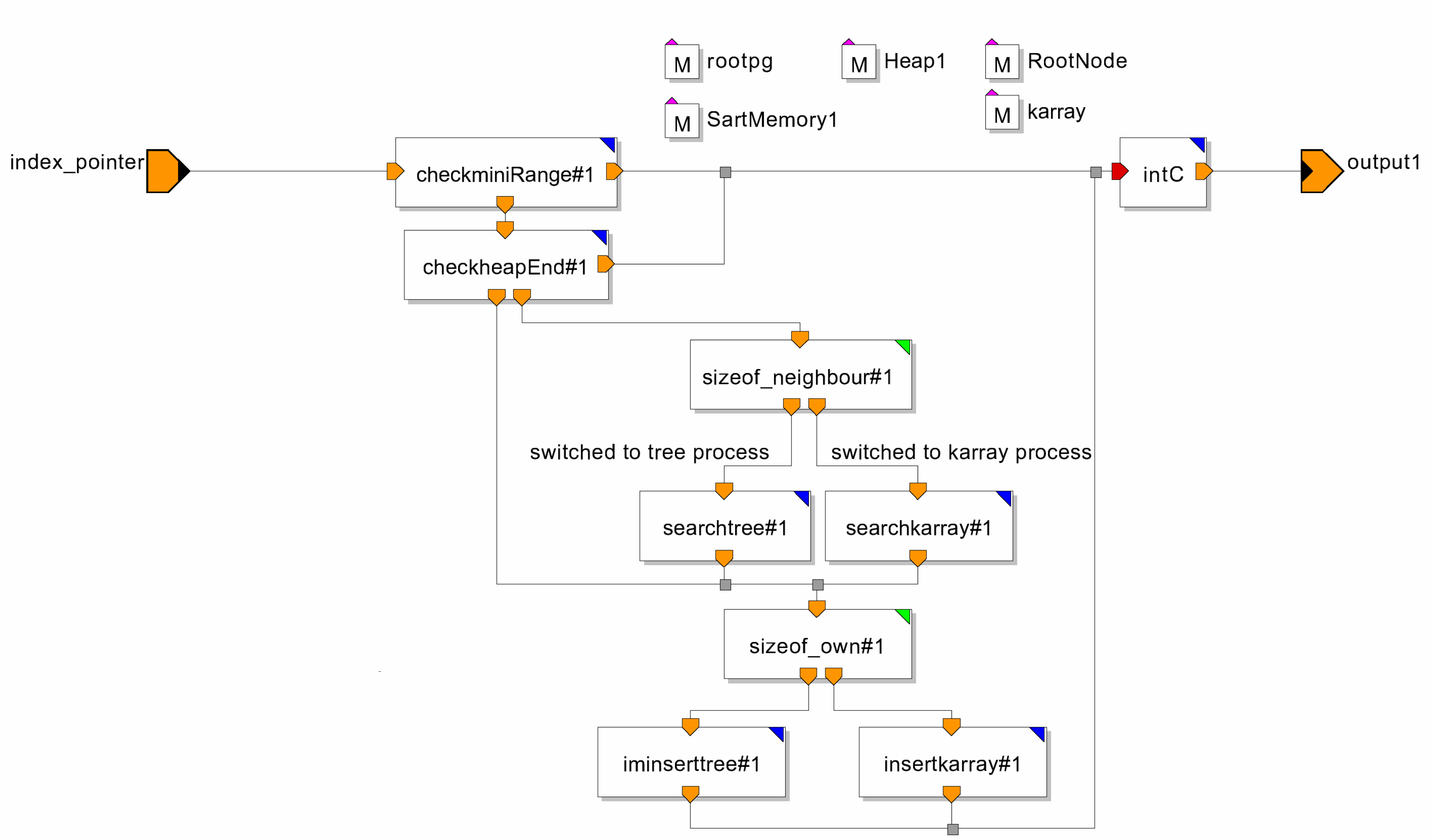
- Scheduler:
  - Combined (rate monotonic + preemptive)
- System services: Pair of blocks for each
  - Interface to application tasks
  - Interface to other kernel modules
- Invoced by events
- States represented by shared objects

Memory Management Module

- Functions shown:
  - *nalloc()* allocating memory
  - *nfree()* releasing memory
- Linked by shared variables
- Info for logging purposes provided



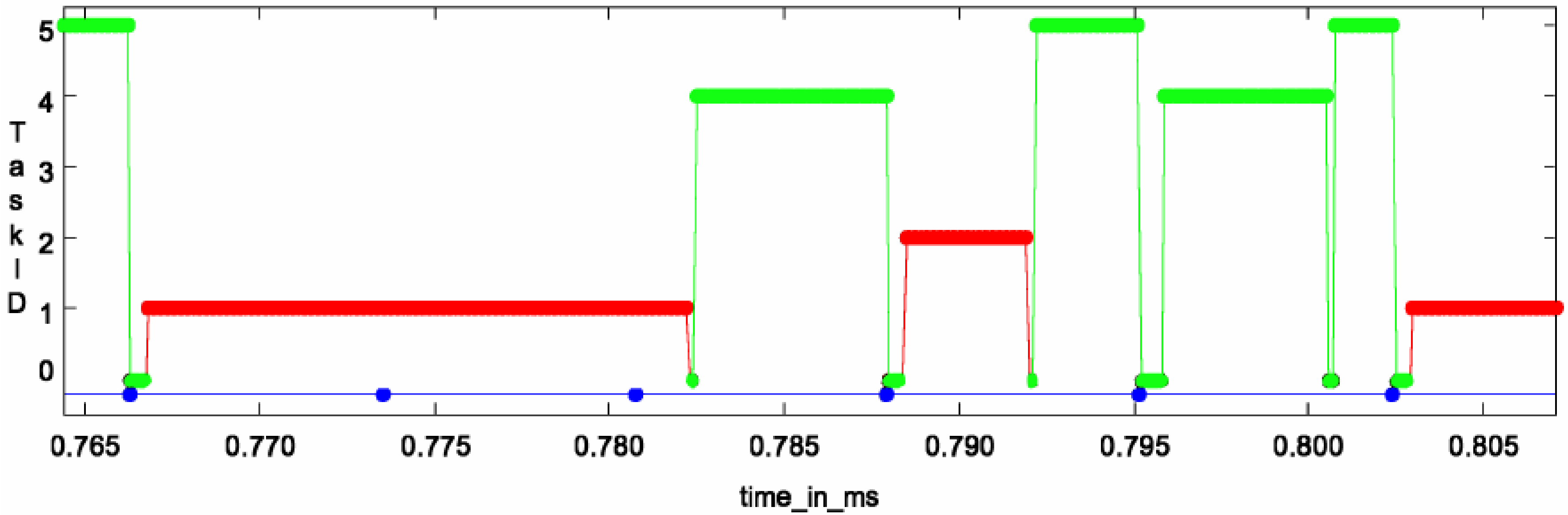
Module ‚nfree‘ in Memory Management



4. Simulation and Evaluation

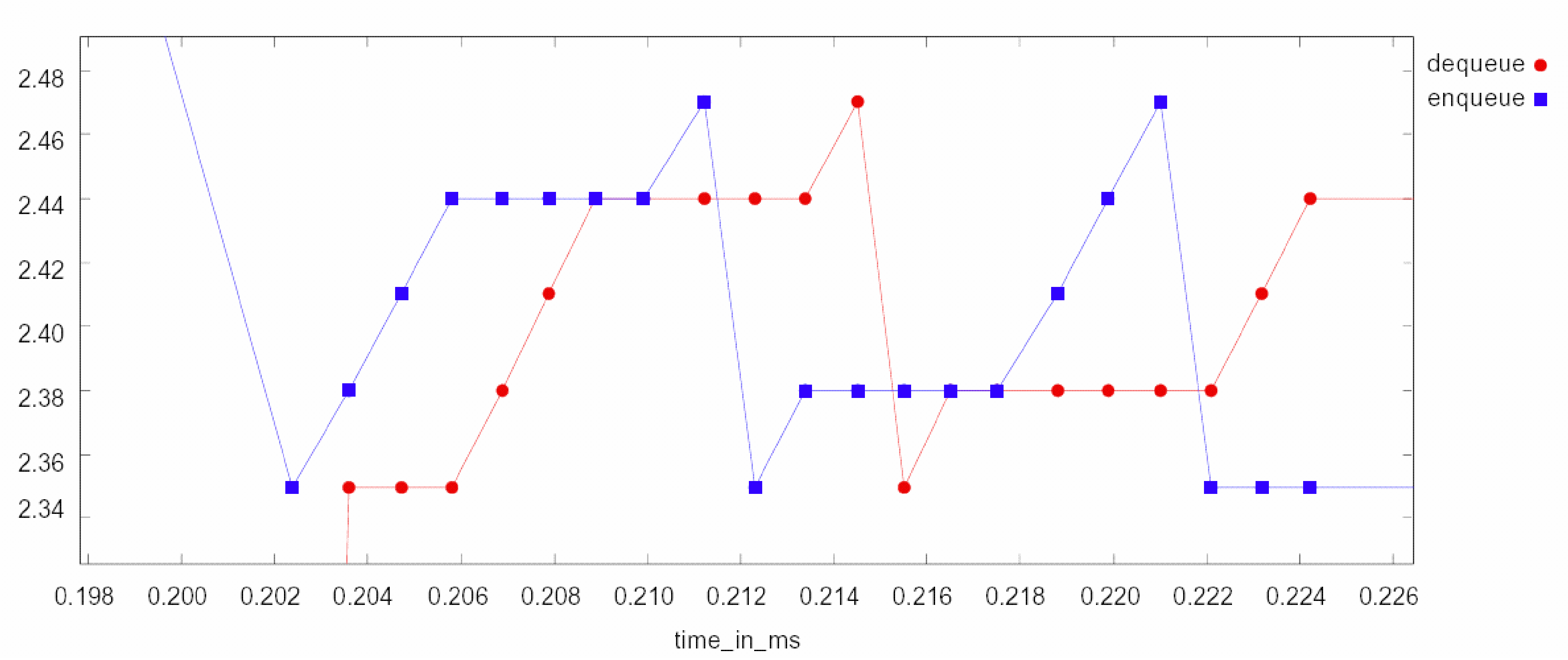
- Runtime scenario for simulation
- Information collected:
  - Time stamps
  - Task status vs. time
  - Resource status vs. time
  - ….
- Visualized by standard MLDesigner components

Example Task Switch Diagram



(enlarged detail)

Example Memory Occupation View

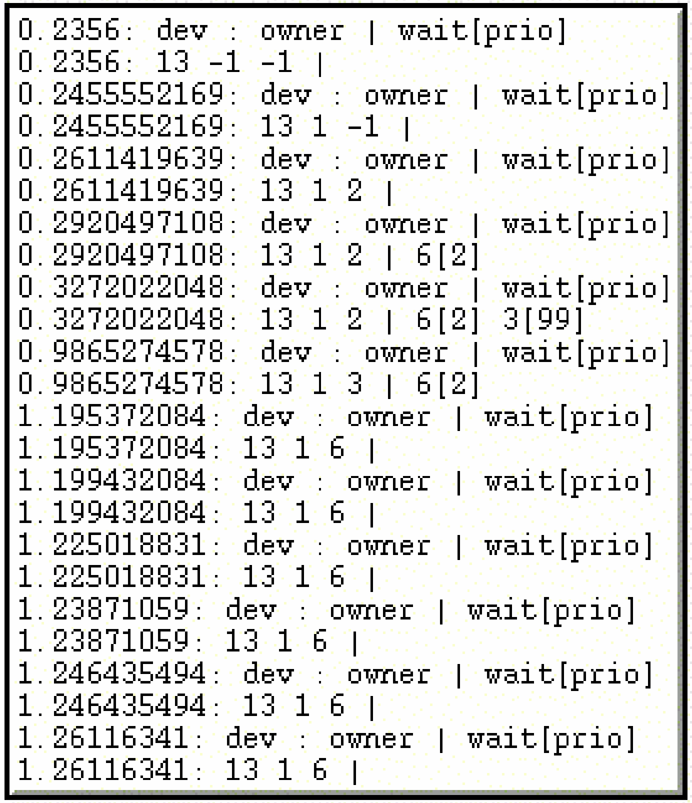


- Shows FIFO memory

(enlarged detail)

Example Device Occupation View

- Colums:
  - Time stamp
  - Device id and state
  - Current owner
  - Pending requests
- Live listing



```
0.2356: dev : owner | wait[prio]
0.2356: 13 -1 -1 |
0.2455552169: dev : owner | wait[prio]
0.2455552169: 13 1 -1 |
0.2611419639: dev : owner | wait[prio]
0.2611419639: 13 1 2 |
0.2920497108: dev : owner | wait[prio]
0.2920497108: 13 1 2 | 6[2]
0.3272022048: dev : owner | wait[prio]
0.3272022048: 13 1 2 | 6[2] 3[99]
0.9865274578: dev : owner | wait[prio]
0.9865274578: 13 1 3 | 6[2]
1.195372084: dev : owner | wait[prio]
1.195372084: 13 1 6 |
1.199432084: dev : owner | wait[prio]
1.199432084: 13 1 6 |
1.225018831: dev : owner | wait[prio]
1.225018831: 13 1 6 |
1.23871059: dev : owner | wait[prio]
1.23871059: 13 1 6 |
1.246435494: dev : owner | wait[prio]
1.246435494: 13 1 6 |
1.26116341: dev : owner | wait[prio]
1.26116341: 13 1 6 |
```

5. Conclusion

- Contributes to validation and testing
- Formal analysis not supported
- **Further work:**
  - More detailed modeling of control flow inside tasks
  - Support for protocol verification
  - Generating software from model

Questions?