Applying Priorities to Memory Allocation

Sven Gestegård Robertz
Department of Computer Science
Lund University
Sweden
Background

Embedded systems

- Small memory
- Real-time applications
- Robustness
Background

Memory is a global resource

- Out-of-memory errors have serious consequences
- Great responsibility on programmers
Not all of the code is critical

- Critical parts must always be executed
- Non-critical parts may be skipped if there is not enough memory to run them safely
- Critical and non-critical “aspects”
The basic idea

Prevent system from running out of memory by limiting the amount of non-critical allocations.

- Traditionally done manually
- Run-time system support

Priorities for memory allocations!
Key points

- Guarantee that all critical high priority allocations succeed w/o delay
- Prevent non-critical memory allocations in high priority processes from starving low priority processes.
- Memory priority and CPU priority are orthogonal
- Worst case analysis only needed for the critical parts
Non-critical limit

Keep the amount of live, non-critically allocated memory below a safe limit

or

Keep the amount of allocatable memory above the safe level
Non-critical limit

- Critical memory allocated by HP process
- High priority allocations
- Memory available for non-critical allocation
- Reserved for high priority allocations
- Non-critical limit
- Only critical allocations allowed
- GC cycle
- Heapsize
- Free memory
- Allocated memory
- Time
Experimental results:

Simple control application with logging

- Control – critical
- Logging – non-critical

```java
void control()
{
    calculateControlSignal();
    updateState();
    try{
        deliverLogData();
    } catch(NoNonCriticalMemoryException e) {
        // not enough memory to safely allocate log data
    }
}
```
all allocations critical
log data is non-critical
close-up
Performance

a) log data objects are always allocated

b) allocation of log data is non-critical
Summary

- Memory requirements can be separated into “critical” and “non-critical”
- Separate memory and CPU time priorities
  Not all of the allocations in a HP process are critical
- Run-time system support
- Improves robustness and performance
- Worst case analysis only needed for critical parts