

# The Simula trace at Lund University

Boris Magnusson  
Computer Science  
Lund University, Sweden  
Boris.Magnusson @ [cs.lth.se](mailto:cs.lth.se)



# How it all began in Lund

- 1973-74 SIMULA for the university UNIVAC 1108 computer
- Course in Data-structures and Simulation by Göran Eriksson
- Projects in Discrete Event Simulation, industrial applications
  
- 1977 Department "mini-computer" Data General Eclipse
- Plans for its use were already out-dated, useless for us.
- Write a Simula system for it - how hard can it be?
  
- 1980 Mostly operational



# Lund Simula for DG Eclipse

- S200 -16-bit computer
- 64kb address-space
- Compiler and RTS written in pure assembler, for size.
- Practical limitation: 2000 source-lines a. A bit restrictive
- Code - objects
  - Compiled Position-Independent code stored on file
  - Loaded dynamically into heap-storage - when called
  - Removed when GC did not free enough memory
- Program size increased significantly
  - Handled programs on top of demos nicely
  - Small program cpu-time increased 6%
  - Larger programs run much faster with fewer GC.s.



# Lund SIMULA for other machines

- DG MV Eclipse - 32bit computer.
- Port the SIMULA system ? (written in assembly) - why not?
- Defined a "universal" assembly - ZMAC
  - typed variables (address, adress-offset, short, long,...)
  - instructions for address manipulation, INCRA - Increment address
- Translation system (in SIMULA)
  - doing the bulk of the jobb
  - interactive, asking questions when it was unsure.
- Compiler for ZMAC to
  - 16bit Eclipse
  - 32bit Eclipse
- Handle SIMULA programs of "any size".



# While at it ...

- Ported to a wide range of computers/operating systems:
  - SUN-3 M68020 / BSD Unix
  - SUN SPARC /Solaris
  - VAX / UNIX & VMS
  - HP M68K/HPUX
  - APOLLO M68K/ UX
  - MAC M68K / Mac OS 6-9
  - MAC PPC / Mac OSX
  - MAC Intel / MAC OSX - Still works.
- Denote-time (cf statictimer by Karel B.)
  - S200 1uSec
  - current Mac 0,6 nanoSec
  - Factor of at least 2000 faster over 35 years



# SIMULA for realtime

- Basic idea:
- Interrupt -mapped to-> detach
- Implemented as a detach injected between lines of code.
  - Used support for single-stepping in the Simula debugger.
- Kernels could now be written in Simula, in the style of Concurrent Pascal and Modula
- One kernel, SimIOProcess distributed with Lund Simula
- Interrupts could be turned on/off
  - When turned off the program was deterministic (handel an input at the time). Simplified debugging.
- Used for responsive GUIs
- Animated Simulations.



# Mjölner SIMULA System

- Part of project 19865-1990
- Integrated development environment for Simula
  - Structured editor driven by a grammar (interpreted)
  - Incremental compilation in real-time
  - Incremental code-generation
  - Integrated execution environment.
  - Change running program.
- LISP like environment for a strongly typed OO language.
- Used internally at Lund, not widely spread



# SIMULA in teaching at LU

- Advanced course in Simulation from 1974
- 1990 replaced Pascal as introductory language
  - We wanted to start OO and choice of language was easy
  - Courses in Data-structures and Real-Time programming also in Simula
- 1997 Java replaced Simula as introductory language.
- (2016 Mixture of Java and Scala)





Thanks for the attention!

