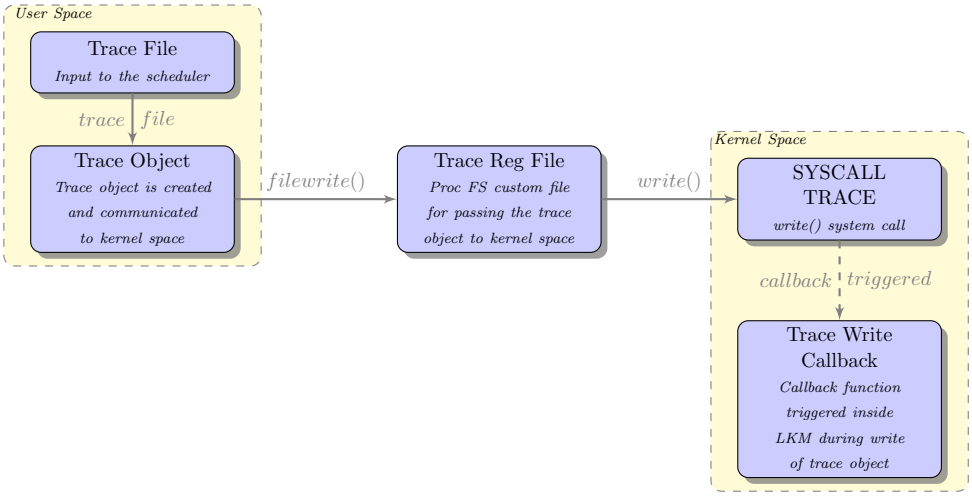
Pseudo implementation of fine-grained scheduler in kernel space

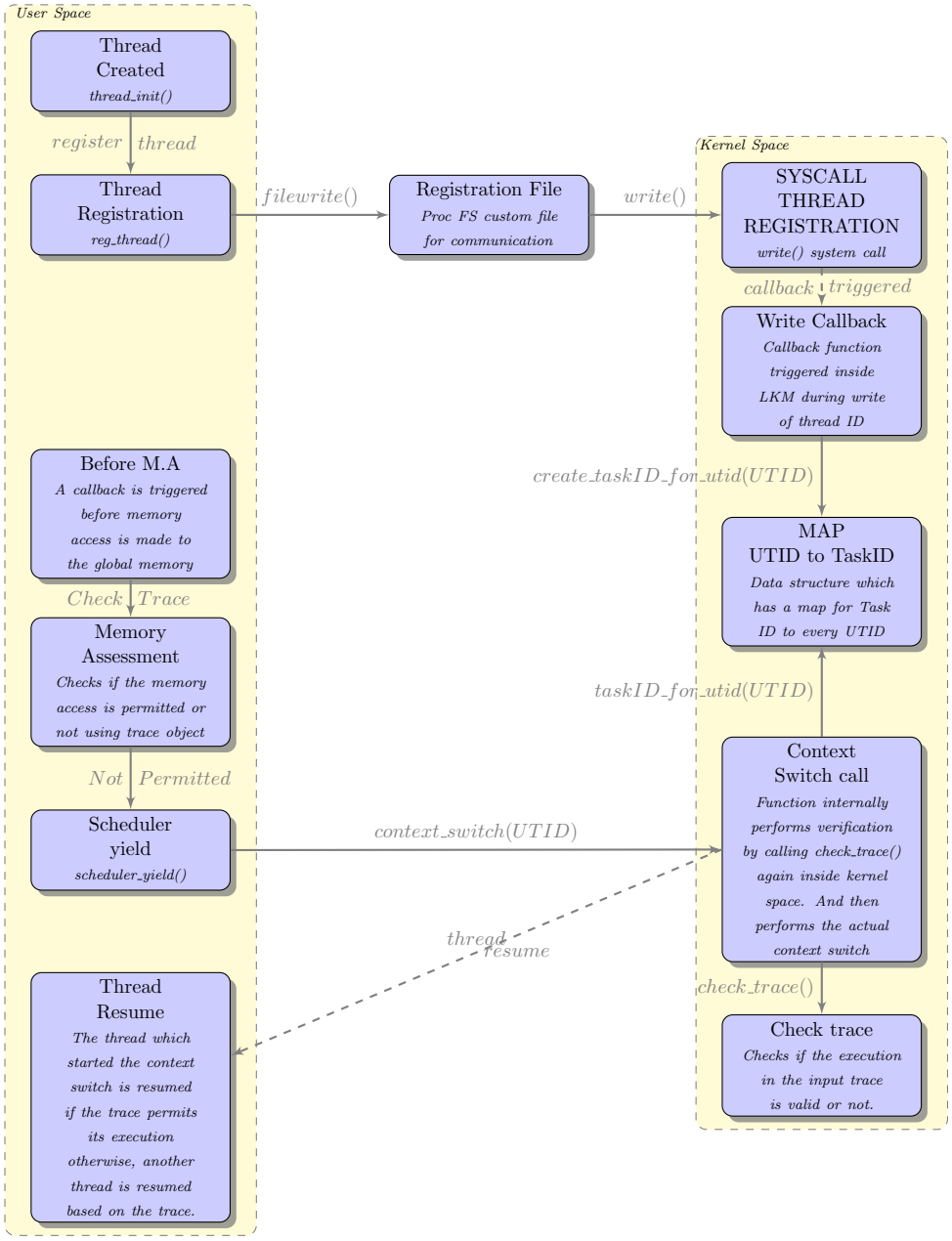
Sreeram Sadasivam
M.Sc Distributed Software Systems
TU Darmstadt, Germany
sreeram.sadasivam@stud.tu-darmstadt.de

Trace file mapping to kernel space



The trace file is passed on as an input for the scheduler. In the above flow diagram, the trace file is read by the main user thread at the start of its execution. It parses the file, creates and passes the trace object to the kernel space as string via a custom file created in the proc file system.

if $i \geq maxval$ **then**
  $i \leftarrow 0$
else
  if $i + k \leq maxval$ **then**
    $i \leftarrow i + k$
  end if
end if



In the above picture, the registration block happens when a user thread is created. The registration happens via a custom proc file system. In the memory assessment block, the user thread initially performs a check of permissions to access memory. The check happens with the help of the trace

object created from the trace input file. The memory assessment block is involved whenever there is a global memory event. Based on the assessment in kernel space, the assessment thread is resumed or another thread which was paused would be resumed.