LUATeX 0.79
an update

BachoTeX 2014
speed

issues

• there is this \TeX\ niche: automated workflows, complex and/or huge documents
• we need to find a compromise between convenience and efficiency
• we deal with font loading and processing, list manipulations
• we need to keep (slower) portable devices in mind

recently done

• integrate \texttt{LUAJIT\TeX} in the source tree
• optimized aspects of the \TeX-LUA interface
• improved some of the logging (also more callbacks)

todo

• locate bottlenecks in the engine
• cleanup some of the IO
• integrate (and improve) the (backend) error handling
nodes

issues

- crossing the \TeX\-LUA boundary comes at a price
- with millions of calls (macro package features) there is a speed penalty
- some complex scripts are demanding (like devanagari)

recently done

- an additional access model for stable and access-heavy node usage

done

- not that much conceptually (the current model is quite okay and probably as efficient as it can be)
- cleanup of merged engine code (redundant tests, nodes, etc)
expansion

issues

• playing with the LUA parbuilder showed inefficiencies
• multiple auto-generated fonts were used for the stretched glyphs
• many calculations happened several times

recently done

• (already done before) make the backend code mode efficient
• introduce a stretch field in glyph nodes
• use that to pass information to the backend (no extra fonts)

todo

• look into protrusion
• cleanup the frontend code a bit (delayed work)
• remove (or make optional) kern stretching
• specify stretch relative to glyph width and not emwidth

(t:/manuals/hybrid/parbuildertest/2013/test.tex)
parbuilder

issues

• the parbuilder code is a merge of engine code
• some code is no longer needed or can be done better

recently done

• fix some inconsistencies in the output oft the parbuilder (math)

todo

• create consistent output from the parbuilder
• make dir whatsits into proper core nodes and improve them a bit
backend

issues

• front and backend are not clearly separated (inheritance of \textsc{PDF\TeX})
• some code has an experimental character

recently done

• make the PDF properties more consistent (accessors)
• improve the \textsc{LUA-\TeX} interface

todo

• promote some generic features to core features (like images and transformations)
• implement some backend register setters as macros on top of \textsc{LUA} (cleaner code)
• separate the backend code even more
• remove some (never used) experimental code (or features that are done in \textsc{LUA} anyway)
• cleanup the code (no longer needed \textsc{pdf} prefixes etc.)
backlinks

issues

- the internal lists were never really meant to be accessed
- some backlinks (prev nodes) were not okay (esp. in math lists)

recently done

- make all lists properly back-linked
- fix the head nodes (that themselves can be next nodes of temp)

todo

- check and double check the fixes
properties

issues

• sometimes you want to carry information with nodes
• we cannot keep extending nodes (never ending story)

recently done

• provide a (global) properties table that can have a table per node
• cleanup and copying is handled automatically

todo

• (in CONTEXT:) move some test code in the core
• finish the experimental interface for associating data
LUA calls

issues

• user nodes have a fixed set of values
• \latexlua nodes are strings but functions can be handy too (they carry states)

recently done

• add a LUA specific field (type)
• check such LUA values for string (compile and run) or functions (run)

todo

• nothing
artefacts

issues

• the code clearly shows that it’s a 30+ year mix
• the (impressive translation to c) code was a first step
• not all code (macros) reflects what it is meant to do (CWEB side effects)
• some documentation is (and was already) not in sync

recently done

• fix bits and pieces

todo

• make all more consistent
• add and fix the documentation
callbacks

issues
- there are still missing callbacks

recently done
- add the missing page flushing callbacks
- add callbacks for opening and closing files reporting
- provide bit more control over error messages

todo
- add proper reporting for library messages
furthermore

recently done

- the usual bug fixes
- the usual library updates
- some pending tracker issues

todo (short term)

- all of this
- look into LUA 5.3 (especially new hybrid number concept)
- finish and integrate swiglib (end of year)