Re: GPL vs LGPL vs CAS

Source: http://sci.tech−archive.net/Archive/sci.math.symbolic/2005−01/0271.html

---

From: Richard Fateman (*fateman_at_cs.berkeley.edu*)
Date: 01/30/05

Date: Sun, 30 Jan 2005 18:20:07 GMT

Bernard Parisse wrote:

>
> *I think it is the right newsgroup to discuss about how*
> *CAS software should be developped, where would you see*
> *this kind of discussion otherwise?*

Since the same issue comes up in other software, including
numerical computation, graphics, and facilities as
diverse as browsers, it seems to me that the particular
perspective of CAS would not add much to the discussion.

> *And there is one deeper question: the development of science*
> *is an open process, therefore the software for science should*
> *also be open*

I do not know why you say this. As far as I can tell
the development of computers and related science has not
been an open process. Konrad Zuse, Bletchley Park, come
to mind. The development of weapons and drugs.

: the source code of the algorithms in a CAS
> *should be published (I don't know for interfaces, but the*
> *distinction could be done, and proprietary interfaces could*
> *certainly work over open−source computing software much like*
> *proprietary software work with the linux kernel).*

I think that you are expressing a preference. I
generally agree with it. It is not something you and
I control.
>
−−− rjf said..
>>
>> *Government research funding in the USA for building*
>> *computer algebra systems is $0.*
>
>
> *There are however most probably academic positions in the corresponding*

> *fields of maths or computer science, and I don't see why contributing*
> *to an open CAS could not be seen as a research publication,*
> *therefore the number is $0 only because people in these academic*
> *positions do not see software development as research.*

Because the US funding agencies do not see system building
as research, professors who do CAS−related work must propose
something else to get funding from NSF, DARPA, DOE, ONR,
NSA, in the United States.

The (European) idea that a professor has automatic
funding and can run a project, with students and staff, without
external funding has good and bad aspects. It might allow
the development of a new CAS, at least for a while. But then
it runs out.

In the US, my experience is the persons who promote faculty members
in mathematics do not see writing programs as the equivalent of publication.
Computer science departments tend to be better in this regard.

.....
.
>>
>
> *mupad is more open than other commercial systems*
> *(except for the kernel).*

My impression is that Maple code can be viewed, except for
comments, and except for the kernel, by anyone with access
to Maple. This is not free, but I have definitely benefited
from looking at how Maple did some things to understand
its behavior or even bugs.

> *At least if it fails commercially*
> *everyone has a copy of the library source code (I would*
> *of course prefer to see the library under a free license*
> *as it was promised by Oliver Kluge in Lyon in 2003).*

> *The problem of funding in EU is probably similar to the US one,*
> *it's that people having an academic position do not consider*
> *that software developement is part of their research work.*
>

This is not true in the US at least in computer science. Software
development of novel software is a big part of some of my
colleague's work.

But completing a system to the point of robust packaging,
delivery, technical support, porting to new hardware etc etc
is not part of academic software. That is when industrial
and commercial partnerships play a role. At that point GPL

interferes with technology transfer. The "BSD" or "MIT" license
is much better for this (that is, you can use it, but don't
sue us if it doesn't work). My university would much
prefer technology transfer to be done by specific licenses
but my colleagues and I try to avoid this.

>> On the other hand, one formerly commercial system, Axiom
>> is now free. Macsyma, in reality the Maxima
>> system based on an older version which was always open−source,
>> has become more popular because the commercial 'fork'
>> is no longer supported.
>
>
> But these systems have suffered from a long hibernation period
> because the copyright owners needed time to acknowledge
> that they should be freed.

The copyright owner for commercial Macsyma has not "freed" it
at all.

> New developments require therefore
> time (they need to catch up the improvements in CAS algorithms
> made during the 10 or 20 last years),

I do not believe that, beyond a certain point of competence that
the algorithm improvements are the key
at all. The set of features does not increase because there
is (say) a faster FFT algorithm for multiplying polynomials.
Most of the algorithms in Maxima date from 1980, but this
includes a sparse modular GCD, an polynomial factoring
algorithms that is probably good enough for most applications,
etc.
In fact I would guess that the theoretical asymptotic O(f(n))
improvements
that have filled the last 20 years of journals are of remarkably
little impact on systems like Mathematica, Maple. They make
their mark by what computations can be done, e.g. look
at Wester's benchmarks. Asymptotic timing is probably not
so important!

> it is not clear that they will do these improvements
> and expand outside of their initial
> users. If for example axiom had been developped openly from
> scratch, it would probably be much more different.
>
I see no reason to believe this. I suspect that for the
longest part of its development as Scratchpad, anyone expressing
an interest in participating could look at it and help. After
some brief discussion (and a long delay because of IBM lawyers),
I actually got a free computer (IBM−RT) and Scratchpad at
Berkeley for experimentation.

>
> *I do believe CAS software development should not necessarily be GPL,*
> *but at least open−source and the right choice of licence should be*
> *made after thinking of the balance of power, e.g. LGPL gives too much*
> *advantages to commercial software, dual−licensed GPL/commercial*
> *is probably a much better option.*

I would prefer to see the largest number people to use my software.
The idea that we could force Maple or Mathematica to become "free"
by making our code GPL seems unlikely. Our code could be
rewritten if it mattered.

The "MIT" license can be seen at
http://www.opensource.org/licenses/mit−license.html

RJF