SO LONG AND THANKS FOR ALL THE SHARE PACKAGES

Steve Linton

Division of Computer Science, St. Andrews

July 7, 1997

Where Now

You can find out more about GAP:

- from the Web site `http://www-gap.dcs.st-and.ac.uk/~gap`
- from the GAP manual
- from papers that report work involving GAP (incomplete bibliography on the WWW site)
- from the `gap-forum` and `gap-trouble` e-mail lists (again see the WWW)
- the `gap-workshop` alias will also be maintained for a while
- from the GAP library (or even kernel) code
- by experiment
- we are contemplating another workshop in '99; we hope that there will be similar workshops elsewhere

When you use GAP

- Please write to the GAP forum and tell us the general direction you are working in. We like to know, it makes us feel appreciated, and people may have helpful comments
- Please remember to cite GAP when you publish
- If appropriate, please consider "going the extra mile" to make your programs generally usable and contributing them.

  We have an experimental refereeing system for share packages that provides "official" recognition for substantial work. Smaller programs can simply be deposited in our `incoming` directory.
- Have fun, and do good work!

GAP 3.4.4

- The version you have been using this week will be released very soon
- After the release there may be bug fixes released and probably will be new share packages (watch the forum)
- Our main efforts go into GAP 4 (more later)
- A few pointers:
  - All library and package functions have names beginning with a capital letter
  - It is safest to always use variable names beginning with a small letter
  - Especially do not use `C`, `E`. `X` or `Z` or things may stop working
  - Variables beginning `My` are also safe
  - A few useful commands – `time`, `Runtime`, `Profile`, `LogTo`, `PrintTo`, `AppendTo`
  - From the break loop you can see what is going on and try to fix it – `Backtrace` command

The Future for GAP

Transfer to St Andrews

- Prof Neubüser retires at the end of July
- Martin Schönert left academia last September
- This year the "centre" of GAP development will move to St Andrews

  > A move from an Aachen-based project with international involvement to an international project coordinated at St Andrews

  In other words, we hope to persuade you, and others world-wide, to do a lot of the work. We are trying to set up ways to help, and to acknowledge the effort involved.
- A GAP Council has been set up to encourage GAP development, and keep us in touch with GAP users world-wide.

GAP 4

- Developed in Aachen (mostly) over the last two-three years
- Re-engineered kernel
- New library organisation
- Many new features – system and mathematical
- Alpha test releases (weekly snapshots of our development version) available now
- Beta test release (with some documentation) in May or June
- First full release later this year

What's Not Changed

- For interactive use, or simple programming, GAP 4 looks a lot like GAP 3
- Some commands have changed name
- Some things will be done much more efficiently

The GAP 4 Kernel

- Martin Schönert, Frank Celler
- Rebuilt from the ground up
- More efficient memory manager
- Easier to extend
- 64 bit clean
- Faster function calling
- Streams
  - More sophisticated file handling
  - More flexible communication with other processes
- Save/load workspace
- GAP compiler

The GAP 4 Library

- Much extended – Aachen team, including Thomas Breuer, Bettina Eick, Alexander Hulpke, Heiko Theißen
- New permutation group code – Akos Seress
- Much better handling of vector spaces and algebras
- Speed problems with Sets of Sets largely removed – immutable lists and records
- New polycyclic gropup code, handles infinite pc groups, based on Deep Thought – Werner Nickel, Wolfgang Merkwitz
- Iterators and enumerators
- Many other goodies
- New structuring

GAP 4 Library Structure

- A very brief introduction
- Most functions a user calls become Operations
- An Operation is a place-holder for a whole lot of different functions – it's Methods
- All Methods for an Operation compute the same thing, but may work for different sets of inputs
- The system will call the "best" Method applicable to the inputs (method selection)
- Method selection can depend on "types" of inputs, on their relationships and on facts learned about them since they were created
- Easy to add new methods that work in specific circumstances
- Fairly easy to add new types of objects, new Operations, etc.

Thanks

- Three main speakers
- Other Speakers
- Mike and Werner
- Computer Science technical staff
- GAP developers
- all of you