Compiling PARI from the GIT repository

B. Allombert and K. Belabas

IMB
CNRS/Université de Bordeaux

12/01/2015

Introduction

This talk focuses on the current development version of the PARI library (2.8.*), available from our GIT repository, see

http://pari.math.u-bordeaux.fr/anongit.html

The text of this talk is available in the files `sources.*` in

http://pari.math.u-bordeaux.fr/Events/PARI2015/talks/

No GIT ?

Linux: install the git package.
Windows users: download a precompiled binary `gp-gmp-git*.exe` from

http://pari.math.u-bordeaux.fr/pub/pari/windows/snapshots/

Other users without GIT: download `pari-2.8-*.tar.gz` from

http://pari.math.u-bordeaux.fr/pub/pari/snapshots/

PARI compilation

```
git clone http://pari.math.u-bordeaux.fr/git/pari.g
cd pari
./Configure --prefix=GP --mt=pthread && make -j4 gp
make doc
make statest-all
make test-parallel
make install
./Configure --prefix=GP.dbg -g && make -j4 gp
make install -C Olinux-x86_64.dbg
GP/bin/gp
```

GP configuration

Create and customize `~/.gprc`. Add

```
histfile = "~/.gp\_history"
colors = "lightbg" \\ or "darkbg"
lines = 40
parisizemax = 4G \\ or the maximum amount of memory
                 \\ GP can use
read "~/.gprc.gp"
```

Create an empty file " /.gprc.gp"

GP2C compilation

From GIT (need automake, autoconf):

```
git clone http://pari.math.u-bordeaux.fr/git/gp2c.g
cd gp2c
./autogen.sh
./configure --prefix=$PWD/../GP \
            --with-paricfg=../GP/lib/pari/pari.cfg
            --with-paricfg.dbg=../GP.dbg/lib/pari/p
make check
make install
cd ..
GP/bin/gp2c -v
```

From source: get `http://pari.math.u-bordeaux.fr/download.html#GP2C` extract and skip `./autogen.sh` above.

Updating GIT

To update GIT to the most recent revision:

```
git fetch
git rebase origin/master
```

Git branches

Local branches : go to 'pari' (or any subdirectory, including the ones `Configure`).

```
> git branch
  kb-modsym
  * master    * = current branch, master is the
DEFAULT branch
  mpbern
```

Go to kb-modsym *local* branch

```
> git checkout kb-modsym
> git branch
* kb-modsym
  master
  mpbern
```

(Configure, make, test things. . .) `git checkout master` to get back to main development branch.

Remotes branches

Remote branches : by default, we only know the official repository (the one we cloned from, `origin`), but you can add other ones.

```
> git branch -a     # -a = add remote branches
...   many branches omitted !
remotes/origin/loic-bnf
...
> git checkout origin/loic-bnf -b bnf
```

Here we created a local branch `bnf` out of the `loic-bnf`, and switched to it.

(Configure, make, test things. . .) `git checkout master` to get back to main development branch.

Git branches

You can go back to the master branch by doing

```
git checkout master
```

You can see the latest commits in the current branch with

```
git log
```