BuildCarat.sh

```
#!/usr/bin/env bash

set -e

# This script attempts to build Carat.  Normally, you should run
# this script from the 'pkg' subdirectory of your GAP
installation.

# You can also run it from other locations, but then you need to
tell the
# script where your GAP root directory is, by passing it as first
argument
# to the script. By default, the script assumes that the parent of
the
# current working directory is the GAP root directory.

# You need at least 'gzip', GNU 'tar', a C compiler, sed, perl to
run this.

# Contact address: aiichi.yamasaki@gmail.com

# Note, that this isn't and is not intended to be a sophisticated
script.
# Even if it doesn't work completely automatically for you, you
may get
# an idea what to do for a complete installation of GAP.

GMP=gmp-6.0.0a.tar.bz2
GMPmd5sum=b7ff2d88cae7f8085bd5006096eed470

LIMIT=10000

if [ $# -eq 0 ]
  then
    echo "Assuming default GAP location: ../.."
    GAPDIR=../..
  else
    echo "Using GAP location: $1"
    GAPDIR="$1"
fi
```

```
# Is someone trying to run us from inside the 'bin' directory?
if [ -f gapicon.bmp ]
  then
    echo "This script must be run from inside the pkg directory"
    echo "Type: cd ../pkg; "$(cd $(dirname $0) && pwd)"/"`basename
$0`
    exit 1
fi

# We need any carat subdirectory, to test if $GAPDIR is right
SUBDIR=`ls -d carat*/ | head -n 1`
if ! (cd $SUBDIR && [ -f $GAPDIR/sysinfo.gap ])
  then
    echo "$GAPDIR is not the root of a gap installation (no
sysinfo.gap)"
    echo "Please provide the absolute path of your GAP root
directory as"
    echo "first argument to this script."
    exit 1
fi

if (cd $SUBDIR && grep 'ABI_CFLAGS=-m32' $GAPDIR/Makefile > /dev/
null) ; then
  echo "Building with 32-bit ABI"
  ABI32=YES
  CONFIGFLAGS="CFLAGS=-m32 LDFLAGS=-m32 LOPTS=-m32 CXXFLAGS=-m32"
fi;

# Many package require GNU make. So use gmake if available,
# for improved compatibility with *BSD systems where "make"
# is BSD make, not GNU make.
if ! [ x`which gmake` = "x" ]; then
  MAKE=gmake
else
  MAKE=make
fi

echo "Attempting to build GAP package Carat."

download_gmp() {
(
echo -n 'Download '"$GMP"' ? '
read input
if [ $input != 'Y' ] && [ $input != 'Yes' ] && [ $input != 'y' ]
&& [ $input != 'yes' ]; then
```

```
  echo 'Aborted.'
  return 2
fi
cd $GAPDIR
if ! [ -d extern ]; then
  mkdir extern
fi
cd extern
if [ "$(uname)" == 'Darwin' ]; then
  curl -O 'https://ftp.gnu.org/pub/gnu/gmp/'$GMP
else
  wget 'https://ftp.gnu.org/pub/gnu/gmp/'$GMP
fi
if [ -f $GMP ]; then
  if [ "$(uname)" == 'Darwin' ]; then
    MD5=`md5 $GMP`
    MD5=${MD5## *}
  else
    MD5=`md5sum $GMP`
    MD5=${MD5%% *}
  fi
  if [ $MD5 == $GMPmd5sum ]; then
    echo 'Download succeeded.'
    return 0
  fi
fi
echo 'Download failed.'
return 1
)
}

malloc_stdlib() {
(
if ! [ -f "$1" ]; then
  echo "$1"' not found.'
  return 1
fi
if ! [ -f "$1".org ]; then
  cp -p "$1" "$1".org
fi
perl -pe 's/malloc\.h/stdlib\.h/g' -i.bak "$1"
)
}

itoa10() {
(
if ! [ -f "$1" ]; then
  echo "$1"' not found.'
  return 1
fi
if ! [ -f "$1".org ]; then
  cp -p "$1" "$1".org
fi
perl -pe 's/itoa\(/itoa10\(/g' -i.bak "$1"
)
}

set_limit() {
(
if ! [ -f "$1" ]; then
  echo "$1"' not found.'
  return 1
fi
if ! [ -f "$1".org ]; then
  cp -p "$1" "$1".org
fi
perl -pe 's/'"$2"'[0-9]+/'"$2$LIMIT"'/g' -i.bak "$1"
)
}

config_makefile() {
(
if ! [ -f Makefile.org ]; then
  cp -p Makefile Makefile.org
fi
GMPDIR=${GMP%.tar.*}
GMPDIR=${GMPDIR%a}
perl -pe 's/^ *CFLAGS.*$/CFLAGS = -O2/g;
  s/tar [jzxv]+f gmp-[0-9.a]+\.tar\.(gz|bz2)/tar jxvf '"$GMP"'/g;
  s/functions\/gmp-[0-9.]+/functions\/'"$GMPDIR"'/g;
  s/ln.*Gmp$/ln -nfs \$(TOPDIR)\/functions\/'"$GMPDIR"' \
$(TOPDIR)\/functions\/Gmp/g;
  s/make.*install/make CFLAGS="\$(CFLAGS) -static" CC="\$(CC)"
install/g;' -i.bak Makefile
)
}

config_gaparch() {
(
cd bin
if ! [ -f Makefile.org ]; then
  cp -p Makefile Makefile.org
```

```
fi
perl -pe 's/`\.\/config.guess`-`basename \$\(CC\)`/\.
\/'"$GAParch"'/g' -i.bak Makefile
)
}

build_carat() {
(
# This was a part of BuildPackages.sh .
# Modified by Aiichi Yamasaki.

. $GAPDIR/sysinfo.gap
GAPROOT=`cd $GAPDIR; pwd`

if ! [ -f $GAPDIR/extern/$GMP ]; then
  echo "$GAPROOT"'/extern/'"$GMP"' not found.'
  download_gmp
fi

if [ "$(uname)" == 'Darwin' ]; then
  MD5=`md5 $GMP`
  MD5=${MD5## *}
else
  MD5=`md5sum $GMP`
  MD5=${MD5%% *}
fi

if [ $MD5 != $GMPmd5sum ]; then
  echo "$GAPROOT"'/extern/'"$GMP"' is broken.'
  download_gmp
fi

if [ -f carat-2.1b1.tgz ]; then
  tar zxvf carat-2.1b1.tgz
  CARATSUBDIR=carat-2.1b1
else
  CARATSUBDIR=carat
fi
if [ -d bin ]; then
  if [ -h bin ]; then
    (
    cd bin
    if [ -h bin ]; then
      rm -f bin
    fi
    )
    rm -f bin
  else
    mv -f bin bin.bak
  fi
fi
ln -nfs $CARATSUBDIR/bin bin
cd $CARATSUBDIR
chmod -R a+rX *
malloc_stdlib include/typedef.h
itoa10 functions/Datei/get_symbol.c
itoa10 functions/Datei/read_symbol.c
itoa10 functions/Getput/get_bravais.c
itoa10 functions/Idem/bravais_catalog.c
itoa10 functions/Tools/itoa.c
itoa10 include/tools.h
set_limit functions/Graph/lattices.c 'calloc\('
set_limit functions/Graph/super-k-groups-fcts.c 'malloc\('
set_limit functions/ZZ/ZZ.c 'NUMBER = '
cp $GAPROOT/extern/$GMP functions/$GMP
config_makefile
config_gaparch
make clean TOPDIR=`pwd`
if [ -d bin/$GAParch ]; then
  mv -f bin/$GAParch bin/$GAParch.bak
fi
make Links TOPDIR=`pwd`
make Gmp TOPDIR=`pwd`
make Links TOPDIR=`pwd`
make TOPDIR=`pwd`
)
}

build_fail() {
  echo "= Failed to build $dir"
}

for dir in `ls -d carat*/`
do
  dir="${dir%/}"
  if [ -f $dir/PackageInfo.g ]; then
    echo "==== Checking $dir"
    (  # start subshell
    set -e
    cd $dir
    build_carat
    ) || build_fail
    # end subshell
  else
    echo "$dir is not a GAP package -- no PackageInfo.g"
  fi
done
```

Copyright © 2017 Akinari Hoshi, Aiichi Yamasaki