Introduction

Bcbench is very simple benchmark tool(?).

It is a shell script, which runs on general UN*X OSes.

This script calculates pi and the base of natural logarithm using bc(1).

```
echo "scale=2000; e(1)" | /usr/bin/time -p /usr/bin/bc -l 2>&1 >/dev/null

echo "scale=2000; 4*a(1)" | /usr/bin/time -p /usr/bin/bc -l 2>&1 >/dev/null
```

Requiring "time(1)" and GNU "bc(1)". "time(1)" must accept the option "-p" (means POSIX), and "bc(1)" must accept the option "scale=2000". (If you use NetBSD, you can use pkgsrc/math/bc.)

Please rewrite this if you see some errors when running on your environment.

Software

Source

```
#!/bin/sh

# bcbench - very simple benchmark tool(?)
# http://www.tunagu.gr.jp/cgi-bin/fswiki/wiki.cgi/isihara?page=bcbench (Japanese)
# http://www.tunagu.gr.jp/cgi-bin/fswiki/wiki.cgi/isihara?page=bcbench-e (Broken English)
#
# Please use GNU bc(1) when bc(1) cannot treat "scale=2000" option.
# Please use GNU time(1) when time(1) cannot treat "-p" option.
#
# e(1)   - e (base of natural logarithm) is calculated.
# 4*a(1) - Pi is calculated.

# ChangeLog:
# 20091106: Public Release. (Today is K-OF2009 at Osaka!)
# 20091108: When built-in time command was used, the bug to which the tool
#         : stopped was corrected.
# 20120109: It corrected that two or more options were not able to be specified.
#         : "-c" option was added. This is used when only one of the two of "pi"
#         : or "e" wants to calculate.
#         : Output an error message to STDERR.
#         : Output a result to STDOUT.
# 20120112: "-o" option was added. select output format.
# 20180210: GNU bc forced. (If you use NetBSD, plase use pkgsrc/math/bc.)
#         : Fix error messages when bc do not exists.

# Original idea by Oshima-san:
# cf. http://www.yagoto-urayama.jp/~oshimaya/nbug/etc/bench/index.html (Japanese,Presentation)
# cf. http://www.yagoto-urayama.jp/~oshimaya/nbug/etc/bench/bcbench.html (Japanese,Result summary)

VERSION=20180210

TIME=builtin
BC=builtin
SCALE=2000
CALC=both
OFORMAT=posix

while getopts b:c:ho:s:t:v i
do
        case "${i}" in
        b)
                BC="$OPTARG"
                ;;
        c)
                CALC="$OPTARG"
```

```
                ;;
        h)
                echo "Usage: $0 [-c kind_of_calc] [ -o output_format ] [-b /path/to/bc] [-t
/path/to/time] [-s scale]"
                echo "Example: $0 -s 200"
                echo "Example: $0 -c pi -b /usr/local/bin/bc"
                echo "Example: $0 -c e -o csv"
                exit
                ;;
        o)
                OFORMAT="$OPTARG"
                ;;
        s)
                SCALE="$OPTARG"
                ;;
        t)
                TIME="$OPTARG"
                ;;
        v)
    echo "Version: bcbench $VERSION"
                        echo  "http://www.tunagu.gr.jp/cgi-bin/fswiki/wiki.cgi/isihara?page=bcbench
(Japanese)"
                echo "http://www.tunagu.gr.jp/cgi-bin/fswiki/wiki.cgi/isihara?page=bcbench-e (Broken
English)"
                      echo  "cf.  http://www.yagoto-urayama.jp/~oshimaya/nbug/etc/bench/bcbench.html
(Japanese)"
    exit
                ;;
        *)
                #break
    exit
                ;;
        esac
done

#
# check command path
#
if [ "$BC" = "builtin" ]; then
  which bc >/dev/null 2>&1
  [ $? -ne 0 ] && echo "ERR: bcbench requires bc(1)" 1>&2 && exit 1
  BC=$(which bc)
fi
$BC -v 2>&1 | grep '^Copyright.*Free Software Foundation.*' >/dev/null 2>&1
[ $? -ne 0 ] && echo "ERR: bcbench requires GBU bc" 1>&2 && exit 1

cshf=1
zshf=1
if [ "$TIME" = "builtin" ]; then
  echo $SHELL | grep 'csh$' >/dev/null 2>&1
  cshf=$?
  echo $SHELL | grep 'zsh$' >/dev/null 2>&1
  zshf=$?
  if [ $cshf -eq 0 -o $zshf -eq 0 ]; then
    for h in /usr/bin/time /bin/time /usr/local/bin/time \
       /opt/sfw/bin/time /opt/gnu/bin/time
    do
      if [ -x $h -a -f $h ]; then
  TIME=$h
        break
      fi
    done
    if [ "$TIME" = "builtin" ]; then
      "ERR: bcbench requires time(1)" 1>&2 && exit 1
    fi
  else
    which time >/dev/null 2>&1
    [ $? -gt 0 ] && "ERR: bcbench requires time(1)" 1>&2 && exit 1
    TIME=time
  fi
fi

for h in $TIME $BC
do
  [ "$h" = "time" ] && continue
  if [ ! -x $h -o ! -f $h ]; then
    echo "ERR: $h can not execute." 1>&2
    exit 1
  fi
done

#
```

```
# "-o" option: "posix" (= time(1) posix)
#              "normal" (= time(1) default)
#              "csv"
#              "tw" (for twitter(?))
#
# default is posix.
#
case "$OFORMAT" in
[Pp][Oo][Ss][Ii][Xx])
        OFORMAT="posix"
        timeopt="-p"
        ;;
[Cc][Ss][Vv])
        OFORMAT="csv"
        timeopt="-p"
        ;;
[Nn][Oo][Rr][Mm][Aa][Ll])
        OFORMAT="normal"
        timeopt=""
        ;;
[Tt][Ww])
        OFORMAT="tw"
        timeopt="-p"
        ;;
*)
        echo "ERR: $OFORMAT is unrecognized option. \"posix\",\"normal\",\"csv\" or \"tw\"." 2>&1
        exit 1
        ;;
esac

#
# "-c" option: "both" (= calc e(1) and 4*a(1))
#              "e" (= calc e(1) only)
#              "pi" (= calc 4*a(1) only)
#
# default is both.
#
runbench()
{
  _arg=$1
  echo 'echo "scale='$SCALE'; '$_arg'" | '$TIME $timeopt $BC' -l 2>&1 >/dev/null'

  if [ "$OFORMAT" = "csv" ]; then
    # NetBSD, OpenBSD and Linux(GNU time 1.7) works fine. Other OS are untested. (Sorry..)
    echo "scale=$SCALE; $_arg" | $TIME $timeopt $BC -l 2>&1 >/dev/null \
    | sed -e 's/[ ]/:/' -e 's/[ ]*//g' \
    | tr '\n' ':' \
    | sed "s/\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):/$_arg,\1,\3,\5:\2,\4,\6:/"
\
    | tr ':' '\n'
  elif [ "$OFORMAT" = "tw" ]; then
    echo "scale=$SCALE; $_arg" | $TIME $timeopt $BC -l 2>&1 >/dev/null \
    | sed -e 's/[ ]/:/' -e 's/[ ]*//g' \
    | tr '\n' ':' \
    | sed "s/\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):\([^:]*\):/$_arg:  \1 \2  \3 \4  \5
\6;/" \
    | tr ';' '\n'
    # echo "scale=$SCALE; $_arg" | $TIME $timeopt $BC -l 2>&1 >/dev/null | sed "s/^/$_arg: /"
  else
    echo "scale=$SCALE; $_arg" | $TIME $timeopt $BC -l 2>&1 >/dev/null
  fi
}

case "$CALC" in
[Bb][Oo][Tt][Hh])
        #CALC="both"
        runbench 'e(1)'
        runbench '4*a(1)'
        ;;
[Ee])
        #CALC="e"
        runbench 'e(1)'
        ;;
[Pp][Ii])
        #CALC="pi"
        runbench '4*a(1)'
        ;;
*)
        echo "ERR: $h is unrecognized option. \"pi\" or \"e\"." 2>&1
        exit 1
        ;;
esac
```

Download

■ Shell script

or http://www.rururu.org/~isihara/bcbench/bcbench.sh

MD5 (bcbench.sh) = b6b54d05da51f7268988cd0e6c066d8a

■ NetBSD's binary package

or or http://www.rururu.org/~isihara/bcbench/bcbench-20180210.tgz

MD5 (bcbench-20180210.tgz) = 940f8b007f976fe88fda70b53e3ddc3c

If you use it excluding NetBSD/amd64, Please type pkg_add -f /path/to/bcbench-20180210.tgz

Reference

- http://www.yagoto-urayama.jp/~oshimaya/nbug/etc/bench/index.html (Japanese,Presentation)
- http://www.yagoto-urayama.jp/~oshimaya/nbug/etc/bench/bcbench.html (Japanese,Result summary)
- http://wiki.netbsd.se/bcbench (English)
- http://www.basicallytech.com/blog/index.php?/archives/23-command-line-calculations-using-bc.html (English)
- http://www.rururu.org/~isihara/bcbench/ (old version scripts)

Thanks

- Thank you for Mr.Oshima who invents the idea of this shell script.
- Thank you for NBUG that exists under my motivation.
- Thank you for K*BUG that exists under my motivation.