How to Build the GP 2 Compiler

Ziad Ismaili Alaoui

October 2023

1 Introduction

This brief tutorial is an update to the existing one on the official GP 2 website[1]. It is based upon the GP 2 version available on the GitHub repository[2] on the 16th of October 2023. It assumes that the user is on a Linux device running Ubuntu[3] 18.04 (or later) and has superuser privileges.

2 Procedure

Download the `master` branch on the official repository as a ZIP file, and unzip its content in a folder of preference. Enter the folder `GP2-master` and open a terminal from that directory. If you find a file called `LICENSE`, rename it to `COPYING`. Call the following commands to generate configuration files and to execute them.

```
$ autoreconf -i; autoconf -i; automake -a
$ ./configure
```

Generate a distribution tar file, unzip it and access the `gp2-1.0` folder.

```
$ make dist
$ tar -xzvf gp2-1.0.tar.gz;
$ cd gp2-1.0
```

Call the configuration procedure and copy the library files into the newly unzipped folder.

```
$ ./configure --prefix=/opt/local/gp2c
$ cp ../lib/*.{c,h} lib/
```

Execute the Makefile. *Warning: the next steps require superuser privileges.*

```
$ make
$ sudo make install
```

[1] `https://uoycs-plasma.github.io/GP2/installation/buildcompiler`
[2] `https://github.com/UoYCS-plasma/GP2`
[3] The tutorial may be relevant to other Linux distributions, but it is not guaranteed.

Finally, create an executable file in `/usr/bin`. That would allow you to call the GP 2 compiler from any directory on your machine via the terminal.

```
$ sudo cp lib/* /opt/local/gp2c/lib
$ sudo cp bin/gp2c /usr/bin
```

You would need to make amendments to `/usr/bin/gp2c`. Access the file manually and edit its content to the following. (The code is available for copying on GitHub Gist.)

```
#!/bin/bash

install_dir="/opt/local/gp2c"

flags=""
code=false
output=false

while getopts ":co" opt; do
  case $opt in
    c)
      code=true
      ;;
                o)
      output=true
      ;;
  esac
done

for opt in $@
do
  flags="$flags $opt"
done
flags=${flags%"${BASH_ARGV[1]} ${BASH_ARGV[0]}"}

c="-c "
if [ $code ]; then
        flags="${flags//$c/}"
fi

if ! $output ; then
        flags="${flags}-o ./gp2_code_temp"
fi


echo "1. Making Code Directory"
echo "                mkdir -p ./gp2_code_temp"
```

```
mkdir -p ./gp2_code_temp
echo ""

echo "2. Executing GP2 Compiler on ${BASH_ARGV[1]}"
echo "                $install_dir/bin/gp2 $flags ${BASH_ARGV[1]}"
$install_dir/bin/gp2 $flags ${BASH_ARGV[1]}
echo ""

echo "3. Coping GP2 Library Files"
echo "                cp $install_dir/lib/*.{c,h} ./gp2_code_temp/"
cp $install_dir/lib/*.{c,h} ./gp2_code_temp/

echo "4. Building GP2 Executable"
echo "                make -C gp2_code_temp"
make -C gp2_code_temp
echo ""

echo "5. Executing on Host Graph ${BASH_ARGV[0]}"
echo "                gp2_code_temp/gp2run ${BASH_ARGV[0]}"
gp2_code_temp/gp2run ${BASH_ARGV[0]}
echo ""

if ! $code ; then
        echo "6. Removing code & executable & log"
        echo "                rm -r -f gp2_code_temp ; rm -f gp2.log"
        rm -r -f gp2_code_temp ; rm -f gp2.log
        echo ""
fi

echo "Final Result (stored in gp2.output) is:"
echo "                cat gp2.output"
cat gp2.output
```

Alternatively, you can edit the file via the terminal through the `nano` command.

```
$ sudo nano /usr/bin/gp2c
```