GLIDER: Simulation Language
LINUX Users Quick Guide
RedHat LINUX
IEAC/FACES - CESIMO/FAI

Thanks to
Interdisciplinary Research Center, San Diego State University, CA

Project CDCHT I-524-95-02-AA
Universidad de Los Andes. Mérida, Venezuela. June 2000

GLIDER Development Group

Contents

GLIDER LINUX version

This is a quick helping guide for users of the LINUX version of the GLIDER Simulation Language [1]. This version was built using the GNU Pascal to C Translator (p2c) [6] and the GNU C Compiler

The Graph and Crt Units of TURBO Pascal language, in which the original MSDOS version of GLIDER is based, are emulated in this version using the TPEX Library [5].

Commands Quick Guide

Running locally

Log on, preferably as X Window session for running programs with graphics. Make the subdirectory with your GLIDER programs the current working directory.

```
cd <a-directory>
```

If you want to have access to a collection of examples, type the command:

```
cpdemos <a-directory>
```

Or

```
cpdemos .
```

for current directory. The collection includes examples with graphics.

To compile the GLIDER program **EXE1.gld**:

- For complete compilation: gli EXE1
- For precompilation to C only : glic EXE1
- For completing C compilation: glil EXE1

Remember to delete later the intermediate files

To run the program: ./EXE1

Running remotely from UNIX or LINUX

1. Log on, preferably as X Window session.

2. Tell your local machine to be X Server for the remote LINUX machine running GLIDER programs:

   ```
   xhost <remote-linux-machine>
   ```
   Necessary to run programs with graphics.

3. Establish telnet or rlogin connection:

   ```
   telnet <remote-linux-machine>
   ```

4. Tell the remote LINUX machine that the X Window Server is in the machine you are working at:

setenv DISPLAY <*internet-name or IP-number of -Xserver-machine*>:0.0
Necessary to run programs with graphics.

5. Make the subdirectory with your GLIDER programs the current working directory.

   ```
   cd <a-directory>
   ```

6. If you want to have access to a collection of examples :

   ```
   cpdemos <a-directory
   ```

   Or

   ```
   cpdemos .
   ```

   for current directory. The collection includes examples with graphics. It is convenient to do steps 2 and 4 to be able to run them.

7. To compile GLIDER program **EXE1.gld**:

   - For complete compilation: gli EXE1
   - For precompilation to C only : glic EXE1
   - For completing C compilation: glil EXE1

Remember to delete later the intermediate files

To run the program: ./EXE1

Running remotely from Windows via telnet

1. Start a telnet connection to the remote LINUX machine (*Start* button, *Run*):

   telnet <remote-linux-machine>

2. Make the subdirectory with your GLIDER programs the current working directory and copy the collection of examples: same as steps 5-6 of running from UNIX or LINUX..

   NOTE: Graphic examples CAN NOT be run in Windows without an X Window Emulator.

3. Steps to compile and run are the same as step 7 of running from UNIX or LINUX.

Running remotely from Windows via X Window Emulator

1. Start an X session in the remote LINUX machine using the X Window Emulator
2. Proceed as running from Windows via telnet. Examples with graphics are executables within the X Emulator.

Creating and editing a GLIDER program

Users of the LINUX version have to create and edit their programs using any LINUX or X Window text editor program, as **vi**, **emacs** or **pico**. GLIDER syntax is case *insensitive*. Refer to the GLIDER Reference Manual for language syntax and model programming examples. A web version of that Manual can be seen in http://www.faces.ula.ve/~carlosd/gliweb

GLIDER program file names must have extension **gld**, as myexample.gld

---

GLIDER Vocabulary

Pascal Keywords
Pascal Types
GLIDER Keywords
GLIDER Types
GLIDER Instructions
GLIDER Functions
Pascal Functions and Procedures
Colors

Pascal Keywords

and, array, begin, case, const, div, do, downto, else, end, for, function, goto, if, in, label, mod, not, of, or, procedure, record, repeat, set, then, to, type, until, var, while, xor

---

Pascal Types

boolean, byte, char, double, file, integer, longint, pointer, procedure, real, string, text, word

GLIDER Keywords

DBTABLES, DECL, END., FACTORS, GFUNCTIONS, INIT, MESSAGES, MODULE, NETWORK, NODES, PROCEDURES, RESPONSES, STATISTICS,TABLES

GLIDER Types

CONT, FREQ, GPOINTER, RET, STRING, str80, DOEVENT, DONODE

GLIDER Instructions

ACT, AGRE, ASSEMBLE, ASSI, BEGINSCAN, BLOCK, CLRSTAT, COUNT, CREATE, DBUPDATE, DEACT, DEBLOCK, DISASSEMBLE, DISFEL, ENDSIMUL, EXTFEL, EXTR, FIFO, FREE, GRAPH, INTI, LIFO, LOAD, MENU, METHOD, MOVE, NOTFREE, ORDER, OUTG, PAUSE, PREEMPTION, PUTFEL, REL, REPLIC, RETARD, SCAN, SELECT, SENDTO, SORT, STAT, STOPSCAN, SYNCHRONIZE,TAB,TRACE, TRANS,UPDATE, UNLOAD, UNTRACE

GLIDER Functions

ber, beta, bin, DMEDL,DMSTL, ENTR, erlg, expo, gama, gauss, LL, lognorm, max, maxi, MAXL, MEDL, min, mini, MINL, modul, MSTL, norm, PFIRST, PLAST, poisson, RANDOM, rnd, TFREE, tria, truncate, unif, unifi, VOIDFEL, weibull

Pascal Functions and Procedures

abs, append, arctan, assign, blockread, blockwrite, close, CloseGraph, clrscr, copy, cos, delay, dispose, eof, exp, filepos, filesize, gettime, getdate, gotoxy, int, keypressed, length, ln, new, open, ord, OutTextXY, read, readkey, readln, release, reset, rewrite, round, seek, sin, sqrt, str, trunc, upcase, val, write, writeln

Colors

BLACK, BLUE, BROWN, CYAN, DARKGRAY, GREEN, LIGHTBLUE, LIGHTCYAN, LIGHTGRAY, LIGHTGREEN, LIGHTMAGENTA, LIGHTRED, MAGENTA, RED, WHITE, YELLOW

Compiling a GLIDER program

Complete Compilation

Use the command **gli** for performing a complete compilation in several steps, starting with the GLIDER precompilation step. If the precompiler does not detect any error, the p2c Translator and the gcc compiler are called several times in order to compile the generated Pascal modules and the main program. The following command line will compile the program file **myexample.gld**:

```
gli myexample
```

Note that the **gli** command expects a program file name with extension **gld**, which is added internally to the given name. If the file does not exist or it does not have the gld extension, the compilation aborts giving the error message:

```
File doesn't exist
```

In case GLIDER Compiler finds any syntax error, an error message is given, indicating line and column numbers where the error was detected and the compilation process stops. Use your text editor for correcting the errors and compile again until successful. If you can not find the error cause, please, contact us via e-mail to the address shown at the end of this section.

If no errors were found, after a while -depending on the size of the program- the compilation process will return to the LINUX command mode. In the user directory, besides the program source file, also there must be the executable program with the same name but without extension. For example, after running the above command for compiling **myexample**, the UNIX command **ls** will show the names **myexample.gld** and **myexample**.

The GLIDER precompiler must generate correct TurboPascal programs, so the compilation must not give any error message. If any error message happens to come, please, send us an e-mail reporting the case and include if possible the source program to one of the following addresses:

sananes@faces.ula.ve

Partial Compilation

If you have any concern about the TurboPascal and C modules that the GLIDER precompiler and p2c generate, use the command **glic <*name-prog*>**, that performs only this step, leaving in the user directory the following generated files:

<*name-prog*.pas>, <*name-prog*.c>, <pru.h>, <pru.c>, <pru.pas>, <unid1.h>, <unid1.c>, <unid1.pas>

If the GLIDER program defines one or more additional modules (MODULE instructions), files with names < *unid*<*x*.h>, < *unid*<*x*.c> y < *unid*<*x*.pas will also be generated for each additional module, *x*=2,3....

For continuing the compilation process until generation of the executable program, the command **glil <*name-prog*>** completes the process producing the executable file <*name-prog*>.

The user is responsible in this case for deleting the intermediate files.

Running a GLIDER program

After successful compilation you can run the program. Following the same example, enter the program name as a command:

```
myexample
```

For details about menu presentation, operation and run options, consult the GLIDER Language Reference Manual [3] (www.faces.ula.ve/~carlosd/gliweb)

***NOTE 1*:** If the program seams to be frizzed, press <ENTER. If still does not respond, press <CTRLC> to cancel. You may ask for help to one of the following addresses:

sananes@faces.ula.ve or mdomingo@rohan.sdsu.edu

NOTE 2: If you get the error:

```
Command not found
```

You need to add also your working directory, where you will store your executable programs, to the *path* variable or move the executable to a directory included in your path (like ~/bin in RedHat LINUX)

Running programs with graphics

For running programs which have one or more **GRAPH** instructions, you must be working in the LINUX machine console in XWindow mode. Or, if running from a remote terminal, it must be located in either a machine running some version of UNIX operating system -including LINUX- or in a XWindow terminal, either real o emulated, as for example, SolarNet PCX which is an X emulator by SunSoft for MS Windows. If not so, when the program intents to open the graphic window, you will get the message:

Can't open Display

And the execution ends.

If you are working in a window (terminal) remotely in a UNIX machine, you must tell the local XWindow server to add the Internet address of the machine to its list of remote machines to serve. For doing that, in a local UNIX window session enter the following command:

```
xhost <your-linux-machine>
```

Conversely, you must tell *your LINUX machine* where is the XWindow Server. In the window where you run the remote LINUX session, enter the command:

setenv DISPLAY <*internet-name or IP-number of the Xserver-machine*>:0.0

For displaying a graphic, a GLIDER program opens a new window, which comes to be the active window, while the main window remains inactive. Be aware of any message in the graphic window asking for pressing a key for continuing. When the graphic stops, you must activate the main window for continuing program execution.

***NOTE*:** If the program seams to be frizzed, press <ENTER. If still does not respond, press <CTRLC for cancel. You may ask for help to one of the following addresses:

sananes@faces.ula.ve

Examples of GLIDER programs

A collection of examples of GLIDER programs is available in the **demos** subdirectory. To copy these examples use the command:

```
cpdemos <a-directory>
```

Where <a-*directory*> is a *path* to a directory, including the references "." to the current directory or ".." to the father of the current directory.

The file names of the examples have the form **EXE**<*x***.gld**, *x* = 1,2,3,..20

Differences of implementation with MSDOS version

Some characteristics of GLIDER have not been implemented yet in the LINUX version or have been implemented differently from the MSDOS version or have not been fully tested.

Characteristics not yet implemented are:

- The GRAPHIC option in menus for files

Differences on implementation are:

- Modes of operation: TP version offers to users two modes of operation: normal and Trace. In both modes the program is interruptible. In the LINUX version there are the same modes, but the program is not interruptible.
- In the LINUX version all input key actions require an ending <ENTER> key press.
- The name for the GAMMA function is GAMA.

The *database* features have not been tested in the LINUX version.

References

1 Domingo C., Hernández M., Sananes Marta, Tonella G.: *Lenguaje de Simulación GLIDER: Guía de Referencia* CESIMO-IEAC, Universidad de Los Andes, Mérida, Venezuela, 1994.

2 GLIDER Development Group: *GLIDER Reference Manual* IEAC-CESIMO/ULA, Venezuela & Interdisciplinary Research Center, Department of Mathematical Sciences, SDSU, August, 1996.

3 Grupo de Desarrollo GLIDER. *Guía de Usuarios RedULA*. IEAC/FACES - CESIMO/FAI, Universidad de Los Andes, 1999.

4 BORLAND International, Inc.: *Turbo Pascal version 6.0*

5 Sananes de Domingo, Marta *TPEX: TURBO Pascal Emulation for UNIX Library*, IEAC-CESIMO/ULA, Venezuela & Interdisciplinary Research Center, SDSU, August, 1996. (Version 2 (1997) with Rafael Tineo)

6 Gillespie, Dave. *Pascal to C Translator p2c Manual Pages* (www.synaptics.com/people/daveg)

GLIDER

C. Domingo, M. Hernández (Diseño Lenguaje original) C. Domingo, J.G. Silva, G. Tonella (Diseño Lenguage extendido) C. Domingo, M. Sananes, G. Tonella (Desarrollo) C. Acosta, H. Hoeger, L. Fuentes, T. Jiménez, F. Palm, S. Quiroz, O. Terán, K. Tucci (Colaboradores) R. del Canto, C. Zoltan (Consultores) **Versión UNIX:** C. Domingo, M. Sananes (Desarrollo) M. Quero, N. Moreno (Colaboradores). **Librería Emulación TPEX:** M. Sananes, R. Tineo (Desarrollo) S. Maldonado (Colaborador)

MSDOS is registered trademark of Microsoft Corporation. All Borland products are trademarks or registered trademarks of Borland International, Inc.

GLIDER: Simulation Language
LINUX User Quick Guide
IEAC/FACES - CESIMO/FAI
Universidad de Los Andes. Mérida, Venezuela. Thanks to Interdisciplinary Research Center, College of Sciences, San Diego State University
San Diego, June 2000